Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Teoria Básica das Estruturas Conceptuais

Miguel Alexandre Wermelinger

Dissertação apresentada para a obtenção do Grau de Mestre em Engenharia Informática pela Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia.

Lisboa
(1995)

Para os meus pais
e para a Claudia

Sumário

As Estruturas Conceptuais são um formalismo de representação de conhecimentos baseado em grafos, os chamados grafos conceptuais. A teoria foi inicialmente desenvolvida por John Sowa há dez anos. Desde então, uma comunidade científica cada vez mais ampla tem-na utilizado em muitas áreas de aplicação e propôs várias alterações à teoria original. Também está em desenvolvimento uma implementação estado-da-arte gratuita e, além disso, os grafos conceptuais foram adoptados num padrão ANSI em preparação. Apesar desta actividade não existe de facto uma definição formal, completa, consistente e revista da Teoria das Estruturas Conceptuais.

Esta dissertação vem contribuir para essa definição ao estender, refinar e clarificar as noções básicas da teoria. A classificação dos grafos conceptuais em

- grafos sintacticamente correctos,
- grafos bem tipados,
- grafos ontologicamente correctos, chamados grafos canónicos,
- e grafos verdadeiros

é a base da clarificação do significado das várias noções e serve de guia às extensões e aos refinamentos introduzidos. As principais extensões foram feitas no sistema de tipos e no esquema de dependências entre vértices de grafos, e o refinamento de quase todos os aspectos da teoria — em particular das regras de formação de grafos canónicos e das regras de inferência para os grafos verdadeiros — inclui o tratamento formal de algumas propostas informais de outros autores.

Abstract

Conceptual Structures is a knowledge representation formalism based on graphs, the so called conceptual graphs. The theory was initially developed by John Sowa 10 years ago. Since then, it has been used by a growing scientific community in many application areas, and several changes to the original theory have been proposed. Furthermore, a freely available state-of-the-art implementation is under way, and conceptual graphs were adopted by an ANSI standard currently in preparation. However, there still doesn't exist a complete, consistent and revised formal definition of Conceptual Structures Theory.

This thesis extends, refines and clarifies the basic notions of the theory, thus contributing to its redefinition. The classification of conceptual graphs into

- syntactically correct graphs,
- well-typed graphs,
- ontologically correct graphs, called canonical graphs,
- and true graphs

is the basis for the clarification of the notions' meaning, and it is the guideline for introducing extensions and making refinements. The main extensions proposed regard the type system and the dependencies among graph nodes. The refinements cover almost every aspect of the theory — particularly the formation rules for canonical graphs and the inference rules for true graphs — and include the formal treatment of some informal proposals done by other authors.

Glossário

Na tentativa de traduzir toda a nomenclatura e todos os exemplos de Inglês para Português surgiram alguns casos que realço aqui quer pela sua importância quer pelo seu desvio à regra habitual, a da tradução literal (segundo a qual *individual marker* tornou-se marcador individual, *propositional rules of inference* foi traduzido como regras proposicionais de inferência, etc.). Inclui igualmente uma listagem dos tipos de relação mais usados no resto deste documento. Para além do termo (ou do tipo) original em Inglês, dá-se uma breve explicação (no caso dos tipos) e refere-se o local em que o termo ou tipo é definido ou utilizado pela primeira vez.

Termos gerais

base de conhecimentos knowledge base

engenheiro do conhecimento knowledge engineer

enquadramento frame

hipótese assumption

lista de distribuição (de mensagens) mailing list

padrão standard

significação meaningfulness

Termos da Teoria das Estruturas Conceptuais

tipo type label (Hipótese 3.2.1 na página 27)

tipo(s) de conceito concept type(s) (Hipótese 1 na página 30)

tipo(s) de relação relation type(s) (Hipótese 5 na página 34)

relação de conformidade conformity relation (Hipótese 3.3.3 na página 68)

regras canónicas de formação canonical formation rules (Hipótese 3.4.3 na página 76)

contexto exterior outer context (Definição 4.2.4 na página 54)

com encaixe par evenly enclosed (Definição 4.2.4 na página 54)

com encaixe ímpar oddly enclosed (Definição 4.2.4 na página 54)

elo de correferência coreference link (Hipótese 4.2.5 na página 55)

restrições de selecção selectional constraints (na página 19)

Tipos de relação

AGNT `AGNT`; liga uma acção ao seu agente (grafo na página 16)

ATRIB `ATTR`; liga uma entidade a um atributo seu (grafo na página 21)

ENTRE `BETW`; liga três objectos, um dos quais se encontra entre os outros dois (grafo na página 17)

CARAC `CHRC`; liga uma entidade a uma característica sua (Exemplo 5 na página 35)

ELO `LINK`; liga dois conceitos arbitrários (Hipótese 3.6.13 na página 33)

MANR `MANR`; liga um acto à maneira como é executado (grafo na página 20)

MED `MEAS`; liga um objecto a uma medida (grafo na página 16)

NOME `NAME`; liga uma entidade a um dos seus nomes (grafo na página 16)

OBJ `OBJ`; liga uma acção ao seu objecto (grafo na página 16)

RCPT `RCPT`; liga um acto ao recipiente da acção (grafo na página 19)

CLASSE `KIND`; liga uma entidade ao seu tipo (Exemplo 31 na página 59)

Simbologia e Notação

$\mathbb{N}$	números naturais (sem o zero)
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$
i, j, k, x, y, z	variáveis designando naturais
$t, t', t_i, \ldots$	variáveis designando tipos
$m, m', m_i, \ldots$	variáveis designando marcadores
$c, c', c_i, \ldots$	variáveis designando conceitos
$r, r', r_i, \ldots$	variáveis designando relações
g, g', g_i, u, v, w	variáveis designando grafos
$S_{op\ i}$	o mesmo que $\bigcup_{j\ op\ i} S_j$ em que $op \in \{<, >, \leq, \geq\}$
$\wp(S)$	partes de S, i.e., conjunto de todos os subconjuntos do conjunto S
$\square$	fim de demonstração

$\mathcal{T}_\mathcal{C}$	tipos de conceito	Hipótese 1 na página 30
$\mathcal{T}_\mathcal{R}$	tipos de relação	Hipótese 5 na página 34
$\top_c$	tipo de conceito universal	Definição 3 na página 30
$\top_r$	tipo de relação universal	Definição 9 na página 34
$\perp_c$	tipo de conceito absurdo	Definição 3 na página 30
$\perp_r$	tipo de relação absurdo	Definição 9 na página 34
T_i^{nc}	tipos de conceito não-relacionais de ordem i	Definição 2 na página 30
T_i^{rc}	tipos de conceito relacionais de ordem i	Definição 2 na página 30
T_i^r	tipos de relação de ordem i	Definição 6 na página 34
$T_{\langle i \rangle}^r$	tipos de relação de aridade i	Definição 6 na página 34
`nonrel`	género não-relacional	Hipótese 1 na página 30
`rel`	género relacional	Hipótese 1 na página 30
on	ordem máxima dos tipos de conceito não-relacionais	Definição 4 na página 32
or	ordem máxima dos tipos de conceito relacionais	Definição 4 na página 32
a	aridade máxima dos tipos de relação	Definição 8 na página 34
$\mathcal{M}$	marcadores	Hipótese 11 na página 37
$*$	marcador genérico	Definição 12 na página 37
$\bar{*}$	marcador absurdo	Definição 12 na página 37
$\#t$	marcador individual	Secção 2.3 na página 36
$\mathcal{C}$	conceitos	Definição 13 na página 40
$\mathcal{C}_f$	conceitos bem-formados	Definição 19 na página 42
$\mathcal{R}$	relações	Definição 24 na página 44
$\mathcal{R}_f$	relações bem-formadas	Definição 29 na página 46

RC_i	conceitos relacionais de ordem i	Definição 21 na página 44
NRC_i	conceitos não-relacionais de ordem i	Definição 21 na página 44
V_C	vértices conceptuais	Hipótese 33 na página 50
V_R	vértices relacionais	Hipótese 33 na página 50
E	arcos	Hipótese 33 na página 50
$::$	relação de conformidade	Hipótese 53 na página 72
$::^*$	fecho da relação de conformidade	Hipótese 54 na página 72
δ	denotação	Hipótese 68 na página 105
ϕ	fórmula de primeira ordem	Hipótese 66 na página 100
π	(semi-)projecção	Definição 38 na página 51

Índice

Capítulo 1

Introdução

Nesta dissertação proponho uma reformulação e uma extensão das noções básicas da Teoria das Estruturas Conceptuais (TEC) de Sowa [Sowa, 1984]. Neste capítulo apresento os objectivos do trabalho realizado e descrevo resumidamente tanto a teoria original como a nova proposta.

A primeira secção introduz os objectivos e motivações da teoria modificada, ao mesmo tempo que a enquadra na fase actual da história do desenvolvimento do formalismo dos grafos conceptuais. Na segunda secção é feita uma breve introdução informal à teoria original, juntamente com a sua notação gráfica e linear, de modo a facilitar a compreensão dos exemplos e definições apresentados no resto desta dissertação. A terceira secção, ao dar uma visão global da nova proposta, serve de introdução às modificações feitas, permitindo ao leitor não perder o fio condutor no meio dos muitos detalhes técnicos apresentados nos capítulos que se seguem. Finalmente a quarta secção introduz a notação utilizada e explica a estrutura deste documento.

1.1 Motivação e Objectivo

As estruturas conceptuais são uma tentativa ambiciosa para representar conhecimento de uma forma natural e expressiva. John Sowa começou a desenvolver este formalismo nos anos 70, inspirado pelos grafos existenciais de Charles Sanders Peirce, um dos percursores da lógica moderna. Como o próprio Sowa diz[1],

> "[As estruturas conceptuais são] um sistema lógico com um formalismo baseado em grafos que visa um grande poder expressivo. O seu objectivo primário é servir de linguagem intermédia entre língua natural e outros formalismos incluindo linguagens de interrogação de bases de dados, enquadramentos, regras de produção e cálculo de predicados."

A "bíblia" da Teoria das Estruturas Conceptuais continua a ser [Sowa, 1984], um livro para ensino publicado há mais de 10 anos. A idade e o objectivo do livro têm

[1]Todas as citações pessoais sem menção de referência dizem respeito a mensagens enviadas para a lista de distribuição de mensagens pelos membros da comunidade científica aderente que trabalha em Estruturas Conceptuais. Todas as mensagens se encontram arquivadas na directoria `users/cg` do servidor `ftp.cs.umn.edu`.

várias implicações. Por um lado, o estilo tutorial com que o formalismo foi apresentado levou a que várias noções introduzidas no início do livro não tivessem sido actualizadas nos capítulos posteriores, tal como o próprio Sowa reconheceu. Assim, fica-se por vezes sem saber como é que os vários apectos da teoria estão interligados. Além disso, a formulação informal e incompleta de várias definições levantou muitas perguntas sobre a teoria, mesmo em relação a aspectos fundamentais, como se pode observar em mensagens enviadas para a lista de distribuição de correio electrónico sobre a TEC.

Por outro lado, os 10 anos decorridos desde a primeira apresentação global da teoria permitiram o crescimento de uma comunidade científica bastante diversificada que introduziu muitas modificações e extensões à medida das suas próprias necessidades. Passando em revista as actas das reuniões anuais, realizadas de 1986 a 1992 sob a forma de workshops, e sob a forma de conferências desde 1993, verificamos que as estruturas conceptuais já foram usadas em muitos domínios, destacando-se as bases de dados [Boksenbaum *et al.*, 1993; Carboneill e Haemmerle, 1994], o raciocínio indutivo [Mineau, 1992], a aprendizagem [Levinson, 1993; Liquière, 1993], a engenharia de software [Delugach, 1992] e a contabilidade [Polovina, 1993]. No entanto, a área de eleição continua a ser a língua natural nas suas várias vertentes e aplicações, tais como geração [Velardi *et al.*, 1988; Nogier e Zock, 1992], compreensão de metáforas [Way, 1992b], análise temporal de um discurso [Moulin e Dumas, 1994], pesquisa de documentos [Myaeng *et al.*, 1994], análise de relatórios médicos [Rassinoux *et al.*, 1994; Schröder, 1993], etc. Como seria de esperar, uma tal variedade levou à explosão de notações, extensões e modificações, muitas vezes apresentadas de forma informal, o que não ajuda na clarificação das noções envolvidas.

Apesar da componente aplicacional continuar a ser a mais forte na investigação realizada em Estruturas Conceptuais, nota-se ultimamente uma maior sensibilização para os aspectos mais formais da teoria e para a sua eficiente implementação. Sintoma disso são os muitos artigos dedicados a estes dois temas nas últimas conferências (veja-se por exemplo [Wuwongse e Manzano, 1993; Pagnucco e Foo, 1993; Levinson, 1994; Mineau, 1994] e sobretudo o trabalho de Michel Chein, Gerard Ellis e John Esch, que será referido ao longo da dissertação). Duas causas são provavelmente o aparecimento do padrão ANSI sobre *Information Resource Dictionary Systems* (IRDS) e o lançamento do projecto PEIRCE.

A crescente divulgação das Estruturas Conceptuais— devida em parte à publicação recente de várias colectâneas de artigos [Nagle *et al.*, 1992; JETAI, 1992; Pfeiffer e Nagle, 1993; Mineau *et al.*, 1993a; Tepfenhart *et al.*, 1994a] — levou a que a comissão ANSI X3H4 adoptasse este formalismo como linguagem normativa para os esquemas conceptuais do padrão IRDS [JTC1/SC21/WG3, 1992; Perez e Sarris, 1993]. A responsabilidade dessa decisão, juntamente com a escolha do esquema conceptual para troca de dados, passou recentemente para a comissão X3T2 dedicada a "Interpretação e Intercâmbio de Informação" que é igualmente um dos grupos de aconselhamento técnico do grupo de trabalho ISO/IEC/JTC1/SC21/WG3 dedicado a "*Conceptual Schema Modelling Facilities*" (CSMF). Para além das Estruturas Conceptuais, foram adoptados dois outros formalismos de representação de conhecimentos: o *Knowledge Interchange Format* [Genesereth e Fikes, 1992] e o *Semantic Unification Meta-Model* [IGES/PDES Organization, 1992]. Todos estes três formalismos, largamente divulgados e utilizados por comunidades diferentes, serão usados na especificação da semântica das CSMF. Isto obriga a ter uma definição precisa e rigorosa de cada um dos formalismos, de modo a poderem ser integrados e compatibilizados, o que é facilitado pelo facto de todos eles serem baseados em lógica de primeira ordem.

Por outro lado, um dos maiores travões ao uso generalizado das Estruturas Conceptuais é a falta de uma implementação do formalismo de fácil acesso. De facto, a maior parte dos sistemas existentes impõem fortes restrições ao direito de utilização pelos investigadores em geral ou, quando não o fazem, requerem linguagens e ferramentas particulares para poderem ser usados. Para colmatar esta grave lacuna foi lançado em 1992 o projecto PEIRCE [Ellis e Levinson, 1993; Ellis e Levinson, 1992; Levinson e Ellis, 1993; Ellis e Levinson, 1994] cujo objectivo é a produção de um ambiente de trabalho estado-da-arte que possa ser obtido gratuitamente. A linguagem de programação usada é o C++ e o ambiente de execução será o Unix (opcionalmente com X-Windows). Neste momento os dois módulos principais (a interface gráfica e o gestor da base de grafos) estão bastante avançados. É de realçar que o autor do módulo central responsável pela gestão da base de conhecimentos tem vindo a desenvolver algoritmos para armazenar e pesquisar um elevado número de grafos conceptuais de maneira extremamente eficiente [Ellis, 1992; Ellis, 1993; Ellis e Lehmann, 1994]. A ideia base é tirar partido da estrutura dos grafos e da sua organização em hierarquias segundo a relação de projecção. O seu algoritmo para pesquisa em conjuntos parcialmente ordenados arbitrários foi considerado o melhor existente [Baader *et al.*, 1992]. Neste contexto será também de referir o trabalho de Michel Chein e Marie-Laure Mugnier [Chein e Mugnier, 1992; Mugnier e Chein, 1993a; Chein e Mugnier, 1993; Mugnier e Chein, 1993b] centrado sobre os algoritmos para junção e projecção de grafos.

Paralelamente a estes desenvolvimentos, John Sowa tem advogado a utilização de tipos de ordem superior para uma melhor representação de ontologias e para poder usar as estruturas conceptuais no meta-nível. No entanto, tem apresentado e motivado as suas propostas de maneira informal, sobretudo através de exemplos. Por outro lado, o seu novo livro, de momento em fase de preparação, contém várias novidades visando uma maior elegância e expressividade da teoria. Infelizmente, a versão [Sowa, 1995] distribuída na última conferência sobre grafos conceptuais encontra-se ainda bastante incompleta em certos aspectos, pelo que se está para ver como é que as várias alterações introduzidas, algumas das quais um tanto radicais em relação à formulação original de 1984, vão interactuar.

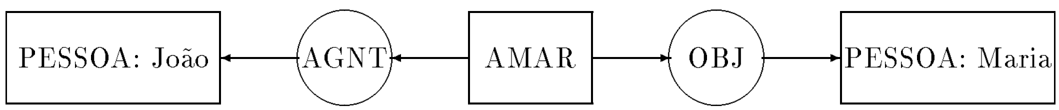
A TEC entra pois na segunda década de existência com uma nova dinâmica e enfrentando novos desafios. É neste enquadramento que surje a presente dissertação, com o intuito de contribuir para a resolução dos problemas teóricos assim como para uma maior divulgação do formalismo. A motivação inicial para este trabalho foi a formalização das ideias de Sowa sobre tipos de ordem superior. No entanto, e como se viu pelo que ficou atrás exposto, rapidamente se chegou à conclusão que o objectivo deste trabalho devia ser uma reformulação de *todo* o núcleo da teoria. A nova proposta deve ser o mais precisa possível, deve incluir várias das ideias surgidas em artigos recentes, e deve clarificar algumas das noções básicas e suas interdependências.

Espero que esta proposta de uma Teoria Básica das Estruturas Conceptuais, uma versão muito melhorada e aumentada da apresentada em [Wermelinger e Lopes, 1994], seja um primeiro passo em direcção a uma definição completa e rigorosa da TEC, tão necessária às actuais tarefas de normalização e de implementação.

1.2 Grafos Conceptuais

Sendo o objectivo deste trabalho a reformulação dos fundamentos da TEC, apresento nesta secção um breve resumo informal da teoria de Sowa[2]. A sua apresentação detalhada e formal far-se-á ao longo da dissertação, a par com a nova proposta.

Basicamente, um *grafo conceptual* é constituído por *conceitos* ligados através de *relações*. Os conceitos são representados graficamente por caixas e as relações por círculos. Por exemplo, a frase "João ama Maria" pode ser expressa pelo grafo



cuja notação linear é

```
[PESSOA: João] <- (AGNT) <- [AMAR] -> (OBJ) -> [PESSOA: Maria].
```

Para poupar espaço usarei a seguinte notação gráfica alternativa, em que as relações são representadas por rectângulos com os cantos arredondados:

Uma relação tem um *tipo* e *arcos* que indicam quais os argumentos da relação. Um conceito é um par da forma $\alpha : \beta$ em que α é o *tipo* do conceito e β é o *campo do referente*. Como veremos adiante, o campo do referente pode conter um ou mais *referentes* e/ou variáveis. Há diversas espécies de referentes, algumas das quais são indicadas no quadro abaixo.

Referente	**Notação**	**Leitura**
genérico	[GATO: *] ou [GATO]	algum gato
individual	[GATO: #]	o gato
nome	[GATO: Morris]	um gato chamado Morris
conjunto	[GATO: {Morris, Felix}]	os gatos Morris e Felix
universal	[GATO: ∀]	todos os gatos
quantidade	[GATO: @2]	dois gatos
quantidade	[GATO: @1]	exactamente um gato
medida	[PESO: @2 kg]	um peso de 2 quilos

Os referentes básicos são os *marcadores individuais* `#1`, `#2`, ...[3], e o *marcador genérico* `*`. Como se vê pela tabela dada, o marcador genérico pode ser omitido. Todos os outros referentes são definidos à custa dos marcadores. Por exemplo, `[GATO: Garfield]` é uma abreviatura de `[GATO]->(NOME)->["Garfield"]` em que `"Garfield"` é um subtipo de `PALAVRA`[4]. Por sua vez, a forma expandida de `[PESO: @2 kg]` é `[PESO] -> (MED) -> [MEDIDA] -> (NOME) -> ["2 kg"]`. Estas expansões definidas

[2] O leitor interessado poderá encontrar resumos mais alargados em [Sowa, 1992] e [Way, 1992a].

[3] O referente `#` é usado quando o indivíduo exacto é desconhecido ou único.

[4] Sowa abandonou entretanto esta concepção das palavras como tipos conceptuais, passando a encara-las como literais: `[PALAVRA: 'Garfield']`.

por Sowa são ontologicamente discutíveis mas uma implementação de Estruturas Conceptuais pode facilmente adoptar outra convenção, pois é apenas disso que se trata.

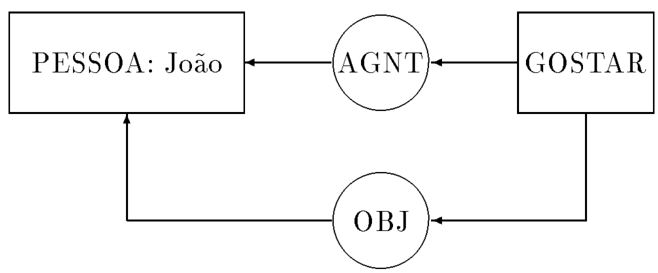
Sempre que a notação linear de um grafo não couber numa única linha, as relações (conceitos) ligadas ao mesmo conceito (relação) podem ser escritas em linhas separadas. O hífen é usado para introduzir uma tal lista de itens, os quais são separados por caracteres de mudança de linha (ver o exemplo seguinte).

Um *variável* é simplesmente um marcador genérico ao qual se deu um nome. A notação é '**nome*'. As variáveis aparecem no campo do referente e, entre outras coisas, são usadas na notação linear para ter em conta circularidades no grafo. Por exemplo, a frase "João gosta de si próprio" poderia ser escrita de qualquer das seguintes maneiras

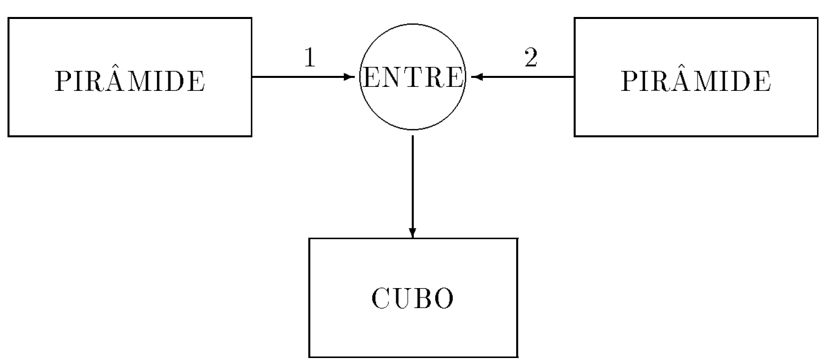
```
(AGNT) -
  ->[PESSOA: João = *x]
  <-[GOSTAR]->(OBJ)->[PESSOA: *x].
```

```
[GOSTAR] -
  ->(AGNT)->[PESSOA: João = *x]
  ->(OBJ)->[PESSOA: *x].
```

que resolvi individualizar para que o leitor possa ver o que acontece quando um grafo não cabe numa só linha. Estas são formas lineares do grafo



Uma relação diz-se n-ária se estiver ligada a exactamente n conceitos. Embora formalmente os grafos conceptuais não sejam direccionados, é preciso conseguir saber a ordem dos argumentos. Nesse sentido, convencionou-se que o arco ligado ao último argumento aponta sempre na direcção deste, enquanto no caso dos restantes $n-1$ argumentos os arcos desenham-se em direcção à relação[5]. Se $n > 2$ então esses arcos devem ser numerados. Por exemplo, "um cubo encontra-se entre duas pirâmides" pode ser representado de pelo menos três maneiras:



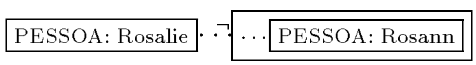
[5]Esta convenção tem a vantagem de ser bastante intuitiva no caso da relação ser funcional: o último argumento representa o valor da função.

```
(ENTRE) -                      [PIRÂMIDE] 1 -> (ENTRE) -
    1 <- [PIRÂMIDE]                -> [CUBO]
    2 <- [PIRÂMIDE]                2 <- [PIRÂMIDE].
    -> [CUBO].
```

Um *contexto* é um conceito cujo tipo é `PROPOSIÇÃO`, o qual pode ser omitido como acontece nos exemplos seguintes, e cujo referente ou é o marcador genérico (*) ou um conjunto de grafos. Os contextos são usados para exprimir afirmações, crenças e modalidades. Nesse sentido, algumas relações como `NEG` (negação), `NECS` (necessidade) e `PSBL` (possibilidade) só podem ser ligadas a contextos. A relação `NEG` pode ser abreviada por $\neg$ ou ~.

Os *elos de correferência* são usados para ligar conceitos que se encontram em grafos e/ou contextos diferentes mas que denotam o mesmo indivíduo. Visualmente são representados por linhas pontilhadas e a notação linear usa referentes múltiplos. Por exemplo (tirado de [Sowa, 1984]), "Rosalie não é Rosann" é representada pelo grafo

ou, na notação que adoptarei,



cuja notação linear é

```
[PESSOA: Rosalie] ; (NEG)->[PROPOSITION: [PESSOA: Rosann = Rosalie] ].
```

que pode ser abreviada para

```
[PESSOA: Rosalie] ; ~[ [PESSOA: Rosann = Rosalie] ].
```

(Os pontos-e-vírgula separam os diversos subgrafos.)

Os tipos dos conceitos formam uma hierarquia, mais precisamente um reticulado cuja relação de ordem é representada por $\leq$[6]. O topo é o tipo universal $\top$ e a base é o tipo absurdo $\bot$. Por exemplo, `COMUNICAR` $<$ `DAR` $<$ `ACTO` significa que `COMUNICAR` é um subtipo de `DAR` que, por sua vez, é um subtipo de `ACTO`, pelo que `COMUNICAR` $<$ `ACTO`. O facto de t ser um subtipo de t' significa que qualquer indivíduo de tipo t também é de tipo t'. Por outras palavras, as instâncias de t são instâncias de t'.

Para impedir a formação de grafos sem sentido, a TEC oferece ao engenheiro do conhecimento dois mecanismos para especificar restrições sobre os grafos permitidos: a *relação de conformidade* e os *grafos canónicos*. O primeiro limita as combinações possíveis entre tipos e marcadores. Dito de outro modo, um conceito $\boxed{t\colon m}$ só será permitido se o marcador m for conforme ao tipo t. Ao especificar a relação de conformidade, o engenheiro do conhecimento está pois a definir quais os conceitos que fazem sentido para a aplicação em causa. No entanto é preciso observar algumas restrições: o marcador genérico é conforme a qualquer tipo; e se um marcador individual

[6] No entanto, os exemplos apresentados nesta dissertação são da forma $t < t'$ porque t e t' serão sempre tipos distintos.

for conforme a um dado tipo t, também o é em relação a todos os supertipos de t. Significa isto que para qualquer tipo t, o conceito $\boxed{t\colon *}$ é permitido. Por outro lado, se por exemplo $\boxed{\text{PESSOA: \#123}}$ for considerado conforme, então $\boxed{\text{ANIMAL: \#123}}$ também o é, visto ser `PESSOA` $<$ `ANIMAL`. A razão destas restrições é que, basicamente, a relação de conformidade serve para especificar quais as instâncias possíveis de cada tipo.

O segundo mecanismo não actua no nível local dos conceitos, mas antes no nível dos subgrafos. Enquanto a relação de conformidade enumera as combinações possíveis entre tipos e marcadores, os grafos canónicos impõem restrições de selecção sobre conceitos e relações, evitando assim a geração de grafos que não fazem sentido (na opinião de quem especifica os grafos canónicos). É por isso que tem que existir um grafo canónico para cada tipo de conceito ou de relação, indicando como é que outros conceitos e relações se relacionam com ele. Por exemplo, os grafos canónicos para os três tipos acima mencionados poderiam ser

1. `[ACTO] -> (AGNT) -> [ANIMADO].`
   Um acto tem como agente um ser animado.

2. `[ENTIDADE] <- (OBJ) <- [DAR] -> (RCPT) -> [ANIMADO].`
   Dar envolve um objecto (uma entidade) e um recipiente (um ser animado).

3. `[INFORMAÇÃO] <- (OBJ) <- [COMUNICAR] -> (INST) -> [ENTIDADE].`
   O objecto comunicado é informação e o instrumento utilizado é uma entidade.

No que se segue, assume-se que as relações hierárquicas entre os tipos ocorrentes são `INFORMAÇÃO` $<$ `ENTIDADE` e `ANIMADO` $<$ `ENTIDADE`, para além de `COMUNICAR` $<$ `DAR` e `DAR` $<$ `ACTO`. Não se considera ser `ACTO` $<$ `ENTIDADE` nem `INFORMAÇÃO` $<$ `ANIMADO` ou vice-versa.

As chamadas *regras canónicas de formação* derivam novos grafos canónicos a partir dos já existentes. Elas também definem um hierarquia sobre os grafos conceptuais: se o grafo g_1 for canonicamente derivável do grafo g_2 então g_1 é uma especialização de g_2 (escrevendo-se $g_1 \leq g_2$) e g_2 diz-se uma generalização de g_1. As regras são extremamente simples:

cópia Um grafo pode ser duplicado.

simplificação Se duas relações estiverem ligadas exactamente aos mesmos conceitos, então pode-se eliminar uma delas.

restrição Pode-se substituir o tipo de um conceito por um subtipo ou o referente genérico por um marcador individual, caso o conceito resultante seja conforme.

junção Conceitos idênticos (no mesmo grafo ou em grafos diferentes) podem ser juntos, passando os arcos de um deles a estar ligados ao outro.

Repare-se que as duas primeiras regras mantêm o novo grafo equivalente ao original, sendo as duas outras regras responsáveis pela obtenção de grafos mais especializados que aqueles de onde se partiu.

Para os exemplos acima dados, no grafo 2 pode-se restringir `DAR` a `COMUNICAR` e de seguida juntar o grafo obtido com o terceiro grafo, obtendo-se

```
[COMUNICAR] -
    -> (OBJ) -> [ENTIDADE]
    -> (RCPT) -> [ANIMADO]
    -> (OBJ) -> [INFORMAÇÃO]
    -> (INST) -> [ENTIDADE].
```

Visto `ENTIDADE` ser um supertipo de `INFORMAÇÃO`, fazendo de novo uma restrição seguida de uma junção chega-se a

```
[COMUNICAR] -
    -> (OBJ) -> [INFORMAÇÃO: *x]
    -> (RCPT) -> [ANIMADO]
    -> (OBJ) -> [*x]
    -> (INST) -> [ENTIDADE].
```

Como se vê, há duas relações de tipo `OBJ` (obtidas da junção dos grafos para `DAR` e `COMUNICAR`) ligadas aos mesmos conceitos. Aplicando a regra da simplificação tem-se

```
[COMUNICAR] -
    -> (OBJ) -> [INFORMAÇÃO]
    -> (RCPT) -> [ANIMADO]
    -> (INST) -> [ENTIDADE].
```

Finalmente, restringe-se no grafo 1 o tipo `ACTO` e junta-se ao grafo anterior. O resultado é o seguinte grafo canónico completo de `COMUNICAR`, uma especialização comum dos três grafos acima dados.

```
[COMUNICAR] -
    -> (OBJ) -> [INFORMAÇÃO]
    -> (RCPT) -> [ANIMADO]
    -> (INST) -> [ENTIDADE]
    -> (AGNT) -> [ANIMADO].
```

Este exemplo também mostra que, juntando os grafos canónicos respectivos, os tipos podem herdar novos "atributos" dos seus supertipos (neste caso `COMUNICAR` herda o recipiente de `DAR` e o agente de `ACTO`) e restringir os existentes (o objecto passa de `ENTIDADE` a `INFORMAÇÃO`). As regras canónicas de formação, além de gerarem novos grafos que não violam as restrições de selecção, implementam pois um mecanismo de herança entre tipos.

Há ainda as regras de inferência que derivam sempre grafos "verdadeiros" (i.e. que expressam afirmações verdadeiras) a partir de grafos verdadeiros. Isto contrasta com as regras canónicas de formação em que os grafos derivados fazem sempre sentido mas podem não ser verdadeiros mesmo que as premissas o sejam. Por exemplo (tirado de [Sowa, 1984]), assumindo que `RAPARIGA` $<$ `PESSOA` e que

```
[PESSOA: Sue] <- (AGNT) <- [COMER] -> (OBJ) -> [TARTE].
[RAPARIGA] <- (AGNT) <- [COMER] -> (MANR) -> [DEPRESSA].
```

são grafos verdadeiros, aplicando as regras canónicas de formação obtém-se a especialização

```
[PESSOA: Sue] <- (AGNT) <- [COMER] -
    -> (MANR) -> [DEPRESSA]
    -> (OBJ) -> [TARTE].
```

que pode ser falsa, pois o facto de existir uma rapariga que come depressa não implica que ela seja Sue. Há dois tipos de regras de inferência: as proposicionais, que manipulam os grafos como unidades indivisíveis, e as de primeira ordem, que também têm em conta a estrutura dos grafos sobre os quais operam. Claro que as regras de inferência de primeira ordem incluem as proposicionais.

Pode-se demonstrar que as regras de inferência são consistentes (isto é, não derivam grafos falsos a partir de verdadeiros) usando o operador ϕ que traduz grafos conceptuais para fórmulas de primeira ordem com igualdade. Um conceito $\boxed{t\text{: } m}$ é traduzido para $\exists x\ t(x)$ se $m = *$, caso contrário obtém-se simplesmente o termo $t(m)$. Os contextos resumem-se a parêntesis, os elos de correferência a igualdades, e os grafos no mesmo contexto são ligados através da conjunção. Por exemplo,

$\phi(\boxed{\text{PESSOA: Rosalie}} \cdot \neg\boxed{\cdots \boxed{\text{PESSOA: Rosann}}}) \equiv$ pessoa(Rosalie) $\wedge \neg$(pessoa(Rosann) $\wedge$ Rosalie=Rosann)

Para além dos grafos "normais" e dos grafos canónicos, a TEC inclui outros géneros de grafos, nomeadamente

definições de tipos para especificar novos tipos de conceito e de relação;

esquemas para especificar conhecimento por defeito;

agregações para especificar indivíduos de um dado tipo;

protótipos para especificar indivíduos típicos;

grafos de fluxo de dados para especificar algoritmos.

Todos estes grafos foram considerados como não fazendo parte do núcleo da teoria, pelo que não serão abordados nesta dissertação. No entanto, segue-se uma breve explicação das definições de tipos por serem usadas em exemplos nos restantes capítulos.

As *abstracções* permitem definir novos tipos (de conceito ou de relação), deste modo sendo possível introduzir novo conhecimento num sistema de grafos conceptuais. A notação linear de uma definição de um tipo de conceito é: **tipo** $t(v)$ **é** u. Nesta notação t é o novo tipo e v é uma variável que indica o seu supertipo no grafo u. Por exemplo **tipo** `GATO-GRANDE(x)` **é** `[GATO:*x]->(ATRIB)->[GRANDE]` implicitamente declara `GATO-GRANDE` $<$ `GATO`. A definição **relação** $t(v_1, \ldots, v_n)$ **é** u introduz um novo tipo de relação t de aridade n. Tendo a relação de descendência entre pessoas pode-se ter **relação** `ANTEPASSADO(x,y)` **é** `[PESSOA: *x]<-(DESCENDENTE)<-[PESSOA: *y]`.

Usando as operações de *contracção de tipo* e de *expansão de tipo* pode-se por exemplo transformar os grafos

[GRANDE] ← (ATRIB) ← [GATO: Felix] ← (AGNT) ← [CAÇAR]

[PESSOA: João] → (DESCENDENTE) → [PESSOA: Pedro] nos grafos

[GATO-GRANDE: Felix] ← (AGNT) ← [CAÇAR]
[PESSOA: João] ← (ANTEPASSADO) ← [PESSOA: Pedro]

e vice-versa.

1.3 Panorâmica Geral

A base da teoria a desenvolver nos capítulos seguintes é uma divisão rigorosa dos grafos conceptuais em quatro categorias: os grafos arbitrários, os bem-formados, os canónicos e os verdadeiros. Cada um destes conjuntos inclui o seguinte, isto é, os grafos verdadeiros são um subconjunto dos canónicos que por sua vez têm que ser bem-formados. Isto significa que não faz sentido falar da veracidade de um grafo não canónico. Do mesmo modo, a noção de canonicidade não se aplica a grafos mal-formados.

Cada nível nesta hierarquia de significação obedece a restrições adicionais às do nível anterior. O primeiro nível, o dos grafos arbitrários, obviamente que não tem quaisquer restrições. Os grafos bem-formados têm que obedecer a restrições impostas pelas hierarquias de tipos, enquanto os canónicos obedecem a restrições ontológicas dadas pela base de conhecimentos. Para além da ontologia a usar, a base de conhecimentos especifica um conjunto de grafos que representam factos sobre o domínio da aplicação. Os grafos verdadeiros serão então todos aqueles que podem ser derivados a partir desse conjunto inicial pelas regras de inferência.

Por outras palavras, um grafo arbitrário está sintacticamente correcto mas pode não ter significado nenhum para a aplicação em causa, um grafo bem-formado também pode não fazer sentido mas pelo menos está bem tipado, um grafo canónico já está ontologicamente correcto mas pode ser falso, enquanto os grafos verdadeiros estão bem tipados, fazem sentido e ainda por cima correspondem a factos verídicos segundo a informação contida na base de conhecimentos. Vejamos então o que caracteriza cada uma das espécies de grafos.[7]

As noções de conceito, relação e grafo conceptual (arbitrário) mantêm-se como na abordagem de Sowa. Assim, continuam a existir duas categorias de tipos: os tipos de conceito e os tipos de relação. Estes últimos são classificados de acordo com a sua *aridade* e *ordem*. A aridade indica o número de argumentos da relação e a ordem restringe os argumentos possíveis. Basicamente, uma relação de ordem n tem como argumentos outras relações de ordem inferior a n, excepto se n valer 1. Nesse caso os argumentos terão que ser entidades não-relacionais. Por exemplo, imaginemos que `INVERSA-DE` é uma relação entre duas relações de primeira ordem. Significa isto que tanto a aridade como a ordem de `INVERSA-DE` são 2. Por outro lado, se `LOC` é uma relação binária entre objectos e lugares, então é de primeira ordem. Mas se não há problema em escrever por exemplo [PESSOA: `#123`] → (LOC) → [CASA: `#456`], já

[7]O texto que se segue não pretende ser um modelo de rigor e precisão, sendo o seu objectivo apenas o de transmitir as ideias principais.

(MAIS-ALTO)→(INVERSA-DE)→(MAIS-BAIXO) não é um grafo conceptual válido porque os arcos não ligam relações com conceitos.

A existência de relações de ordem superior obriga pois a dois *géneros* de tipos de conceito: os *relacionais* e os *não-relacionais*. Por exemplo, `PESSOA` e `CASA` são tipos de conceito não-relacionais. Para representar correctamente o outro exemplo é necessário um tipo relacional, seja ele `RELAÇÃO`: [RELAÇÃO: mais-alto]→(INVERSA-DE)→[RELAÇÃO: mais-baixo]. Os tipos de conceito também se classificam de acordo com a sua *ordem*. Tanto o género como a ordem restringem os referentes que combinam com esses tipos. Simplificando um pouco, o referente de um conceito tem que ser do mesmo género do tipo e não pode ser de ordem superior à do tipo. Caso contrário o conceito não está *bem-formado*. De modo semelhante, uma relação r está *bem-formada* se todos os conceitos que são seus argumentos também estiverem bem-formados e se nenhum deles representar uma relação de ordem superior à de r. Assim, os grafos *bem-formados* são aqueles em que todas as relações e conceitos estão bem-formados.

Tendo visto os tipos necessários, passemos aos referentes. Estando o âmbito deste trabalho limitado aos fundamentos da teoria, os únicos referentes que vão ser tratados são os marcadores, deixando de fora as quantidades, as medidas e outros referentes não primitivos. Ao marcador genérico `*` e aos marcadores `#1`, ... que representam indivíduos do domínio de aplicação, foi acrescentado o *marcador absurdo* $\bar{*}$. Assim, tal como os tipos, os marcadores também podem formar um reticulado, com `*` no topo e $\bar{*}$ na base. Desta forma os conceitos, que são pares tipo-marcador, também formam um reticulado, o que por sua vez induz uma hierarquia sobre os grafos em que um grafo mais específico que outro se designa *instância*. A organização dos vários componentes da teoria em reticulados ou simples ordens parciais, para além de tornar o formalismo mais elegante e regular, tem vantagens computacionais.

Para além destes marcadores, o campo do referente de um conceito pode também conter tipos. A razão vem da própria noção de "ordem superior". Um tipo não-relacional representa um conjunto de entidades. Se for de primeira ordem, as entidades são indivíduos, se for de segunda ordem, as entidades serão tipos de primeira ordem, etc. Por exemplo, o tipo de primeira ordem `GATO` denota o conjunto de todos gatos, mas o tipo de segunda ordem `ESPÉCIE` denota todas as espécies possíveis, as quais são dadas por tipos de primeira ordem tais como `GATO` e `CÃO`. Fará sentido então escrever [ESPÉCIE: gato]. Por outro lado, um tipo relacional denota um conjunto de relações. Por exemplo, `RELAÇÃO` representa todas as relações possíveis, `SIMÉTRICA` o subconjunto das simétricas, `TERNÁRIA` as relações de aridade 3, etc. Assim como convém poder escrever conceito [RELAÇÃO: mais-baixo] também é útil ter [SIMÉTRICA: inversa-de] e [TERNÁRIA: entre] para especificar as classes a que pertencem essas relações. Resumindo, se no conceito $\boxed{t : m}$ o tipo t for não-relacional, então m é um indivíduo ou um tipo de conceito não-relacional. Se t for relacional então m terá que ser um tipo de relação. Como se disse anteriormente, em qualquer caso a ordem de m não poder ser superior à de t. Outra coisa a ter mente é que se t for um subtipo de t' então as entidades denotadas por t são um subconjunto das representadas por t'.

Antes de passar ao terceiro nível de significação, os grafos canónicos, vejamos como serão classificados os grafos segundo a sua forma. Sowa distingue apenas os grafos *simples* por oposição aos *compostos*. Estes últimos têm contextos e elos de correferência, os primeiros não. Como se viu na secção anterior, um contexto é um conceito especial, em que o tipo é fixo e o referente é um conjunto de grafos, e um elo de correferência indica que dois conceitos (eventualmente em contextos diferentes) têm o mesmo referente. A definição de contexto vai ser mantida mas a de elo de correferência vai ser estendida para se poder fazer uma melhor utilização dos tipos de ordem superior. Por exemplo, para definir relações simétricas convém poder escrever a seguinte regra, em que $\top_c$ é o tipo de conceito *universal*:

$$\begin{array}{ll} \text{se} & \boxed{\text{SIMÉTRICA: *r}} \text{ e } \boxed{\top_c\text{: *x}} \rightarrow (\text{*r}) \rightarrow \boxed{\top_c\text{: *y}} \\ \text{então} & \boxed{\top_c\text{: *y}} \rightarrow (\text{*r}) \rightarrow \boxed{\top_c\text{: *x}} \end{array}$$

Esta regra envolve quatro igualdades: duas entre referentes de conceitos, uma entre dois tipos de relação, e outra entre um referente e um tipo de relação. As duas primeiras igualdades são representadas pelas variáveis '*x' e '*y', e as outras igualdades são no fundo uma só, representada pela partilha da variável '*r'. Estas igualdades entre dois tipos, entre um tipo e um referente, ou entre dois referentes serão designadas por *dependências*. Segundo o que foi dito anteriormente, um *elo de correferência* é simplesmente o caso particular de uma dependência entre dois referentes.

A classificação de grafos conceptuais quanto à sua forma passa a ser feita segundo três critérios: a existência de contextos, a existência de dependências e a conectividade. Um grafo diz-se *hierárquico* se contiver contextos, caso contrário é *raso*. Se um *grafo com dependências* continuar conexo depois de remover todas as dependências, então diz-se *simples*, senão *composto*. Estes dois termos têm pois significados diferentes dos dados por Sowa. A categorização dos grafos baseada na sua forma é independente da classificação de acordo com o seu significado. Assim, um grafo conceptual hierárquico está bem-formado se cada um dos grafos contidos nos contextos estiver bem-formado. Um grafo com dependências bem-formado tem pelo menos uma instância bem-formada que satisfaz as igualdades. Quanto aos grafos compostos, cada um dos seus subgrafos tem que estar bem-formado. Na base destas definições recursivas estão os grafos "normais", isto é simples e rasos, para os quais a noção de boa-formação já foi tratada anteriormente.

Passando por fim aos *grafos canónicos*, a sua definição é feita em três passos, como em [Sowa, 1984]: *relação de conformidade*, *cânone* e *regras canónicas de formação*. Um grafo canónico é um grafo bem-formado que está ontologicamente correcto. O cânone fornece a ontologia e as regras permitem derivar todos os grafos canónicos a partir de um conjunto inicial designado *base canónica*. Esta base, juntamente com as hierarquias de tipos e com a relação de conformidade, constitui o cânone. A relação de conformidade permite seleccionar de todos os conceitos bem-formados aqueles que vão ser considerados ontologicamente correctos. Os grafos da base canónica só poderão então usar conceitos que estejam conformes. Para simplificar a abordagem, a base canónica será apenas composta por grafos sem contextos nem dependências. Assim, um grafo hierárquico será canónico se os grafos contidos nos contextos o forem. De modo semelhante, um grafo com dependências canónico tem que estar bem-formado (isto é, as dependências são satisfazíveis) e os subgrafos têm que ser canónicos.

A definição do quarto e último nível de significação, os grafos verdadeiros, faz-se igualmente em três passos. O primeiro consiste em definir uma linguagem de primeira ordem e uma tradução de grafos conceptuais para fórmulas dessa linguagem. De seguida é preciso dizer como essas fórmulas serão interpretadas e finalmente, o terceiro passo é a definição de regras de inferência. De novo simplifiquei o trabalho, abordando apenas os grafos cujas dependências sejam elos de correferência pois eles são suficientes para cobrir a lógica de primeira ordem. Como pretendo que os grafos verdadeiros sejam um subconjunto dos canónicos, segui a mais recente abordagem de Sowa [1995] que consiste em definir as regras de inferência como combinações particulares de regras canónicas de formação. A tradução de grafos conceptuais mantém-se praticamente idêntica à de Sowa. As únicas diferenças devem-se à correcção de pequenos erros e à sintaxe diferente da linguagem de primeira ordem devido ao sistema de tipos usado. Quanto à interpretação, ela segue as condições acima dadas para as hierarquias de tipos.

1.4 Notação e Estrutura da Tese

Assumo que o leitor está familiarizado com alguma da terminologia e notação matemática habitual, sobretudo em relação a conjuntos ordenados (para boas introduções ver [Sowa, 1984, Apêndice A] e [Davey e Priestley, 1990]). Para além da simbologia apresentada na página 9, serão adoptadas as seguintes convenções notacionais:

- para qualquer reticulado finito L, $topo(L)$ designa o seu maior elemento, $base(L)$ o menor elemento, $x \wedge y$ o ínfimo dos elementos x e y, e $x \vee y$ o seu supremo[8];
- para qualquer conjunto parcialmente ordenado, o símbolo $\leq$ designa a ordem parcial[9], e as relações $<$, $>$ e $\leq$ são sempre definidas pelas seguintes equivalências: $x \geq y \Leftrightarrow y \leq x$, $x < y \Leftrightarrow x \leq y \wedge x \neq y$ e $x > y \Leftrightarrow x \geq y \wedge x \neq y$;
- para um dado operador $op \in \{<, \leq\}$, a notação $t_1, \ldots, t_n \; op \; t'_1, \ldots, t'_m$ é uma forma compacta de indicar que $t_i \; op \; t'_j$ para todas as combinações possíveis de $i = 1, \ldots, n$ e $j = 1, \ldots, m$.

De modo a simplificar a notação e evitar a proliferação de nomes e/ou índices, os símbolos de função serão "sobrecarregados", isto é, funções distintas poderão ter o mesmo nome se forem parecidas. Por exemplo, $f : A \rightarrow B$ e $f : C \rightarrow D$ significa que $f(x)$ é um elemento de B se x for elemento de A, e que $f(x) \in D$ se $x \in C$. Para evitar confusões, garante-se que A e C são domínios disjuntos.

A estrutura deste documento tenta ser o mais linear possível: começando nos componentes primitivos, os tipos e os marcadores, passa-se pelos vértices, os conceitos e as relações, para finalmente chegar aos grafos. Deste modo garante-se que todas as definições, hipóteses[10] e teoremas se baseiam estritamente em definições, hipóteses e teoremas anteriores. Apenas no caso dos exemplos é que se abriram excepções, pois para melhor explicar certas noções convém por vezes utilizar noções que formalmente só serão apresentadas mais adiante. Uma situação deste género ocorre logo no início do próximo capítulo, em que a explicação da hierarquia de tipos envolve o paradoxo

[8] Também utilizarei o símbolo $\wedge$ para a conjunção lógica e $\vee$ para a disjunção, mas o uso pretendido será sempre claro a partir do contexto.

[9] Os operandos permitirão distinguir a ordem particular a que o símbolo se refere.

[10] As definições que são fundamentais para a TEC, tais como as de grafo conceptual e hierarquia de tipos, serão designadas por hipóteses. Esta separação das definições vem de [Sowa, 1984].

de Russell, pelo que é preciso recorrer a grafos, contextos e negação — noções que, formalmente, só serão apresentadas em capítulos posteriores. Um dos objectivos da introdução aos grafos conceptuais (Secção 1.2 na página 16) é exactamente o de poder permitir ao leitor entender os exemplos de forma intuitiva, mesmo sem conhecer ainda os detalhes formais.

Outra regra estrutural seguida consistiu em tentar juntar na mesma secção todas os teoremas, definições e hipóteses que digam respeito à mesma noção, mesmo que apenas sejam utilizados muito mais adiante. Parece-me que esta compartimentação das matérias a serem tratadas facilita a sua compreensão e a consulta deste documento. Em relação aos fundamentos lógicos da teoria, esta regra da compartimentação podia ser interpretada de duas formas: ou a denotação dos vários componentes e a sua tradução para lógica de primeira ordem era apresentada juntamente com as restantes definições para esses componentes, ou então juntava-se tudo o que dissesse respeito à lógica num capítulo à parte. Segui esta segunda via por me parecer a que tornava a estrutura global da dissertação mais equilibrada além de facilitar a exposição.

A desvantagem da compartimentação de matérias é a separação de noções que estão relacionadas entre si. Esta deficiência foi colmatada pela abundância de referências cruzadas, que não só permitem ao leitor uma melhor compreensão das interacções entre os vários componentes da teoria, como também possibilitam a rápida revisão de definições anteriores. Além disso, este documento também poderá ser usado como "manual de referência" da nova teoria proposta graças ao índice remissivo.

Cada capítulo começa com um breve sumário da matéria que vai tratar. Cada secção começa pela apresentação das definições originais. Após uma discussão dos méritos e dos problemas da abordagem original, são apresentadas as novas definições que eu proponho, tendo sempre a preocupação de explicar a razão das alterações e dos acrescentos introduzidos. A exposição dos detalhes formais da teoria é entremeada com exemplos da sua aplicação, muitos deles tirados de artigos publicados por outros investigadores de modo a mostrar como a formalização proposta nesta tese incorpora as noções informais apresentadas nesses artigos.

As definições da teoria original encontram-se marcadas com a menção **(Sowa)** e foram traduzidas literalmente, excepto quando o uso da notação e nomenclatura de Sowa causaria confusão com as convenções adoptadas nesta dissertação. Nesses casos optei por usar a notação e nomenclatura deste documento, tornando assim a apresentação mais uniforme. Para o caso do leitor querer consultar rapidamente a definição original não "deturpada", mantive a numeração dada por Sowa: a definição/hipótese/teorema $c.s.n$ é a n-ésima da secção s do capítulo c de [Sowa, 1984].

O facto das definições originais terem sido incluídas textualmente neste documento, em vez de ter optado por uma simples referência, prende-se com as mesmas razões que levaram à abundância de exemplos, à inclusão da Secção 1.2 na página 16 e às opções estruturais acima expostas: a tentativa de tornar esta dissertação auto-contida e facilmente compreensível, mesmo para leitores não familiarizados com a Teoria das Estruturas Conceptuais.

Capítulo 2

Tipos e Marcadores

As noções básicas da Teoria das Estruturas Conceptuais (TEC) são os tipos e os marcadores, a partir dos quais se podem definir conceitos e relações, que por sua vez formam os grafos conceptuais. Isto obriga a ter, tal como na teoria original, duas espécies de tipos: os tipos de conceito e os tipos de relação. No entanto, em ambos os casos são introduzidos tipos de ordem superior, o que leva à distinção de dois géneros de tipos de conceito: os relacionais e os não-relacionais. Tanto os tipos de conceito como os de relação são organizados em reticulados. A classificação hierárquica faz-se, para os tipos de conceito, de acordo com a ordem e o género, e os tipos de relação são ordenados de acordo com a ordem e a aridade. Quanto aos marcadores, para além dos habituais marcadores individuais e marcador genérico, será introduzido um marcador absurdo para que os marcadores também formem um reticulado. Além disso, devido à própria noção de ordem superior, os marcadores também podem ser tipos.

2.1 Tipos de Conceito

A componente mais utilizada em qualquer sistema de Estruturas Conceptuais é certamente a hierarquia de tipos: desde a verificação dos grafos introduzidos pelo utilizador quanto à sua correcção até as regras de inferência, passando pelas regras canónicas de formação, não há praticamente operação que não consulte a hierarquia de tipos.

Em [Sowa, 1984] a noção básica é a de "conceito", a partir da qual se extrai o conjunto de tipos, os quais devem estar organizados num reticulado.

Hipótese 3.2.1 (Sowa). A função *tipo* associa conceitos a um conjunto T, cujos elementos são chamados *tipos*. Os conceitos c e d são do mesmo tipo se $tipo(c) = tipo(d)$.

Hipótese 3.2.3 (Sowa). A *hierarquia de tipos* é uma ordem parcial definida sobre o conjunto dos tipos. O símbolo $\leq$ designa a relação de ordem. Sejam s, t e u tipos:

- Se $s \leq t$, então diz-se que s é um *subtipo* de t, e t é um *supertipo* de s, escrevendo-se $t \geq s$.
- Se $s \leq t$ e $s \neq t$, então diz-se que s é um *subtipo próprio* de t, escrevendo-se $s < t$, e t é um *supertipo próprio* de s, e escreve-se $t > s$.

- Se s é um subtipo de t e um subtipo de u ($s \leq t$ e $s \leq t$), então s diz-se um *subtipo comum* de t e u.

- Se s é um supertipo de t e um supertipo de u ($s \geq t$ e $s \geq t$), então s diz-se um *supertipo comum* de t e u.

Hipótese 3.2.5 (Sowa). A hierarquia de tipos forma um reticulado chamado *reticulado de tipos*.

- Qualquer par de tipos s e t tem um *supertipo comum mínimo*, escrito $s \vee t$. Para qualquer tipo u, se $u \geq s$ e $u \geq t$, então $u \geq s \vee t$.

- Qualquer par de tipos s e t tem um *subtipo comum máximo*, escrito $s \wedge t$. Para qualquer tipo u, se $u \leq s$ e $u \leq t$, então $u \leq s \wedge t$.

- Há dois tipos primitivos: o *tipo universal* $\top$ e o *tipo absurdo* $\bot$. Para qualquer tipo t, $\bot \leq t \leq \top$.

Para perceber o significado destas definições e das que se seguirão, convém já explicar com maior precisão o que são os tipos de conceito não-relacionais[1]. No fundo, os tipos servem para classificar, isto é organizar em classes, as entidades do universo de discurso. Dito de outro modo, um tipo de conceito t representa um conjunto S de entidades. Se essas entidades forem indivíduos, t diz-se de *primeira ordem*. Se as entidades forem elas próprias tipos, então t é de *ordem superior*. Mais precisamente, e se encararmos os indivíduos como tipos de ordem zero, o conjunto S tem que conter entidades da mesma ordem n e então o tipo t que representa S será de ordem $n+1$ (o exemplo que se segue ajuda a perceber estes detalhes). Resumindo, um tipo de ordem n representa um conjunto de entidades de ordem $n-1$ e por sua vez pode ser elemento de um tipo de ordem $n+1$. Por outro lado, visto os tipos serem conjuntos, podem ser organizados segundo a relação de inclusão. Assim, t é subtipo de t' se e só se o conjunto denotado por t for um subconjunto do conjunto representado por t'. Do mesmo modo, t é subtipo comum de t_1 e t_2 se t for um subconjunto da intersecção de t_1 e t_2. Resumindo, a hierarquia de tipos não é mais do que a organização dos tipos segundo a ordem parcial de inclusão de conjuntos. Por sua vez, isto significa que se $t_1 < t_2$ e $t_2 < t_3$ então $t_1 < t_3$ porque a inclusão de conjuntos é transitiva. Além disso, se $t_1 < t_2$ então ambos os tipos terão que ser da mesma ordem, caso contrário os conjuntos que representam serão necessariamente distintos e portanto incomparáveis.

À primeira vista os tipos de ordem superior parecem ter apenas interesse matemático. No entanto, o próprio Sowa, embora apenas use tipos de primeira ordem na teoria original, mostra [1992] a necessidade da existência de tipos de ordem superior para uma melhor modelização de domínios relevantes. O exemplo que ele usa é o seguinte. Imaginemos que queremos representar na hierarquia de tipos os seguintes factos sobre a ordenação taxonómica usada em Biologia:

1. Um gato é um felino.

2. "Felino" é uma família.

[1] Os relacionais comportam-se de maneira ligeiramente diferente e os tipos de relação serão apresentados na próxima secção.

3. “Família” é uma categoria (taxonómica).

A representação `GATO` $<$ `FELINO` $<$ `FAMÍLIA` $<$ `CATEGORIA` está errada porque, devido à transitividade da relação $<$, obtém-se que um gato é uma categoria. Concretamente, o erro consiste em assumir que as palavras “é um(a)” nas três frases anteriores representam sempre a mesma relação e que portanto os tipos são todos da mesma ordem. Na verdade, enquanto na primeira frase a relação é de facto a inclusão, nas outras duas já se trata da relação “elemento-de”. Assim, se `GATO` e `FELINO` forem de ordem n, `FAMÍLIA` será de ordem $n+1$ e `CATEGORIA` de ordem $n+2$. Ora como as entidades denotadas por `GATO` e `FELINO` são indivíduos (que podem ser fictícios, tais como o gato Garfield e o tigre Hobbes, ou reais), n vale 1. Assim, `FAMÍLIA` é um tipo de segunda ordem visto denotar um conjunto de outros tipos (de primeira ordem), tais como `FELINO`, `BOVINO`, `SUÍNO`, etc. É por isso que o tipo `FELINO` é um elemento (e não um subtipo) do tipo `FAMÍLIA`. Do mesmo modo, `CATEGORIA` é um tipo de terceira ordem que denota um conjunto de tipos de segunda ordem tais como `ESPÉCIE`, `FAMÍLIA`, `ORDEM`, `CLASSE`, `REINO`, etc.

Outra das razões para introduzir tipos de ordem superior na Teoria das Estruturas Conceptuais é o facto de querer usar os grafos conceptuais como a sua própria metalinguagem. Um dos grandes objectivos da comunidade que trabalha com Estruturas Conceptuais é representar a própria teoria com grafos conceptuais. Por exemplo, é útil poder escrever grafos conceptuais que especificam propriedades de relações usadas noutros grafos conceptuais. Concretamente, uma maneira expedita de exprimir que a relação “maior-que” é transitiva seria $\boxed{\text{TRANSITIVA: >}}$. Note-se que `TRANSITIVA` é um tipo que não denota indivíduos (tal como `GATO` denota) nem outros tipos de conceito (tal como `FAMÍLIA`) mas sim tipos de relação. Por isso vai haver dois géneros de tipos de conceito: os não-relacionais, tais como os apresentados no exemplo da Biologia, e os relacionais, tais como `TRANSITIVA`, `SIMÉTRICA`, `REFLEXIVA`, etc. Claro que para estes tipos convém ter regras que permitam inferir nova informação. Por exemplo, sabendo que $<$ é transitiva, que $3 < 4$ e que $4 < 8$, deve-se poder inferir $3 < 8$. Como esta espécie de regras envolve noções avançadas, em particular dependências entre os vértices de um grafo, só serão abordadas na Secção 4.4 na página 58. Em particular, veja-se o Exemplo 30 na página 58.

Para terminar esta discussão introdutória sobre tipos de ordem superior, note-se que não atendendo às diferenças de ordem obtém-se o famoso paradoxo de Russell, pois seria possível definir o tipo `S` de todos os tipos que não são instâncias de si próprios: **tipo** `S(x)` **é** `[TIPO: *x] ¬[ [*x: *x] ]`. Assume-se que `TIPO` é o tipo de todos os tipos, o que impede a variável `*x` de ser substituída por um indivíduo como “João”. O paradoxo surge ao tentar verificar se $\boxed{\text{S: S}}$ ou $\neg\boxed{\boxed{\text{S: S}}}$. Substituindo `*x` pelo referente `S`, fazendo a expansão do tipo `S` segundo a sua definição, e por fim eliminando $\boxed{\text{TIPO: S}}$ porque é sempre verdade, chega-se em ambos os casos à negação do conceito original:

$$\boxed{\text{S: S}} \;\rightarrow\; \boxed{\text{TIPO: S}}\;\neg\boxed{\boxed{\text{S: S}}} \;\rightarrow\; \neg\boxed{\boxed{\text{S: S}}}$$

$$\neg\boxed{\boxed{\text{S: S}}} \;\rightarrow\; \neg\boxed{\boxed{\text{TIPO: S}}\;\neg\boxed{\boxed{\text{S: S}}}} \;\rightarrow\; \neg\boxed{\neg\boxed{\boxed{\text{S: S}}}} \;\rightarrow\; \boxed{\text{S:S}}$$

Por todas estas razões estendo a definição da hierarquia de tipos de primeira ordem dada pela Hipótese 3.2.3 na página 27 e pela Hipótese 3.2.5 na página oposta de modo a incorporar tipos de ordem superior, mantendo no entanto a nomenclatura dada por essas duas hipóteses. A nova definição será detalhadamente explicada mais adiante, depois da introdução de mais alguma nomenclatura que facilitará essa tarefa.

Hipótese 1. Uma *hierarquia de tipos de conceito* é um tuplo $\langle \mathcal{T}_\mathcal{C}, género, ordem \rangle$ tal que

1. $\mathcal{T}_\mathcal{C}$ é um reticulado finito e não vazio com $topo(\mathcal{T}_\mathcal{C}) = \top_c$ e $base(\mathcal{T}_\mathcal{C}) = \bot_c$;
2. $género : \mathcal{T}_\mathcal{C} - \{\top_c, \bot_c\} \rightarrow \{\mathtt{rel}, \mathtt{nonrel}\}$ é uma função total;
3. $ordem : \mathcal{T}_\mathcal{C} - \{\top_c, \bot_c\} \rightarrow \mathbb{N}$ é uma função total;
4. $\forall t \in \mathcal{T}_\mathcal{C} \quad ordem(t) > 1 \Rightarrow \exists t' \in \mathcal{T}_\mathcal{C} \quad género(t') = género(t) \wedge ordem(t') = ordem(t) - 1$;
5. para quaisquer $t, t' \in \mathcal{T}_\mathcal{C}$, se $t \leq t'$ então uma das seguintes afirmações tem que ser verdadeira:
   (a) $t' = \top_c \vee t = \bot_c$;
   (b) $género(t) = género(t') = \mathtt{rel} \wedge ordem(t) \leq ordem(t')$;
   (c) $género(t) = género(t') = \mathtt{nonrel} \wedge ordem(t) = ordem(t')$.

Definição 2. Para cada $i \in \mathbb{N}$, a hierarquia dos tipos de conceito *não-relacionais* de ordem i é $T_i^{nc} = \{t \in \mathcal{T}_\mathcal{C} | género(t) = \mathtt{nonrel} \wedge ordem(t) = i\}$ e a hierarquia dos tipos de conceito *relacionais* de ordem i é $T_i^{rc} = \{t \in \mathcal{T}_\mathcal{C} | género(t) = \mathtt{rel} \wedge ordem(t) = i\}$.

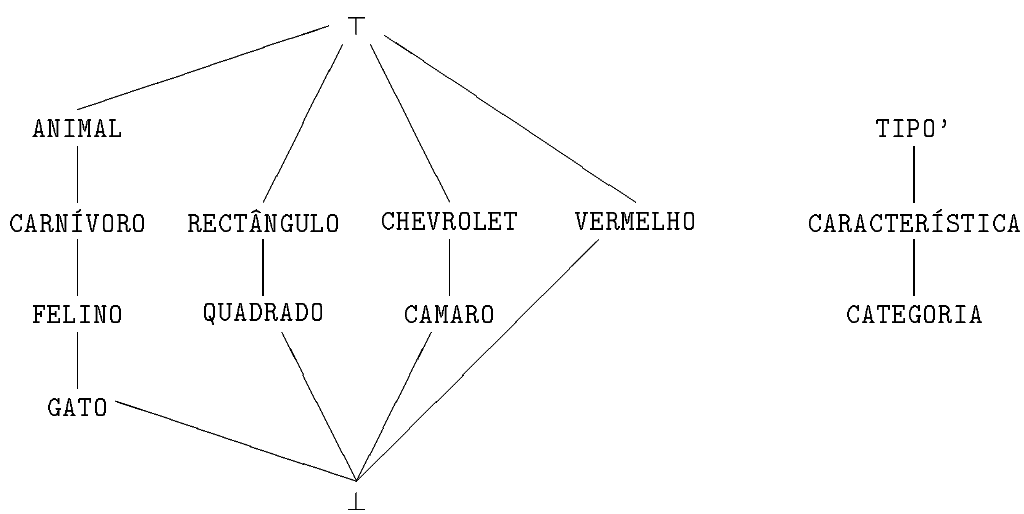
Definição 3. O tipo de conceito *universal* é $\top_c$ e $\bot_c$ é o tipo de conceito *absurdo*.

Segundo a primeira condição da Hipótese 1, o requisito da hierarquia ser um reticulado mantém-se, o que tem vantagens computacionais [Aït-Kaci *et al.*, 1989] além de tornar a operação de junção de conceitos determinista, pois cada par de conceitos terá um único "sub-conceito" comum máximo (Proposição 23 na página 44). Assumo também a finitude da hierarquia de tipos (condição 1) pois ela tem que ser dada explicitamente pelo utilizador. Note-se que embora os tipos tenham sido todos reunidos numa única hierarquia, a estratificação é obtida pela função *ordem* que associa um número natural a cada tipo excepto aos tipos universal e absurdo (condição 3) visto esses tipos, por estarem acima e abaixo de todos os outros, não terem uma ordem específica. Além disso a definição requer que não se salte por cima de ordens (condição 4), isto é, não é possível ter um tipo de terceira ordem sem ter um de segunda ordem do mesmo género. A função *género* permite distinguir os tipos relacionais, que denotam tipos de relação, dos não-relacionais, que denotam outros tipos de conceito não-relacionais (condição 2). Como a denotação é diferente, um tipo relacional não pode ser subtipo de um tipo não-relacional e vice-versa (condições 5(b) e (c)). Para além disso, um subtipo não-relacional tem que ter a mesma ordem dos seus supertipos (condição 5(c)), ou seja, não é possível a um tipo não-relacional ser de ordem 3 e um seu subtipo ser de ordem 2, embora isso seja possível no caso dos tipos relacionais (condição 5(b)).

A razão para este tratamento diferenciado é a seguinte. As propriedades mais interessantes das relações, tais como simetria e transitividade, são independentes da ordem.

Assim, se um tipo de conceito relacional apenas pudesse denotar um conjunto de tipos de relação todos da mesma ordem, então teria que haver tipos `TRANSITIVA-1`, `TRANSITIVA-2`, ... para representar as relações transitivas de primeira ordem, as de segunda ordem e assim adiante. Do mesmo modo, a aridade é independente da ordem. Convém pois ter um único tipo de conceito relacional `BINÁRIA` para denotar todas as relações binárias, em vez de ter tipos separados paras as binárias de primeira ordem, para as de segunda ordem, etc. Note-se que um tipo de conceito não-relacional representa um conjunto de entidades do mesmo género (isto é, outros tipos não-relacionais). Assim, se um tipo de ordem n representasse um conjunto de tipos também de ordem n isso levaria a paradoxos. Mas como um tipo relacional denota entidades de espécie completamente diferente, não há esse perigo. Resumindo, um tipo de conceito relacional de ordem n representa um conjunto de tipos de relação de ordem não superior a n. Como os subtipos são subconjuntos, se t for de ordem n, então um seu subtipo t' pode ser apenas de ordem $n-2$, por exemplo. Basta imaginar que t representa tipos de relação de todas as ordens de 1 a n e que t' selecciona apenas o subconjunto dos tipos de relação de ordem 1 a $n-2$. Como exemplo concreto, pode-se ter `TRANSITIVA-1` $<$ `TRANSITIVA-12`, em que `TRANSITIVA-12` representa todas as relações transitivas de ordem 1 ou 2, e `TRANSITIVA-1` denota apenas as de primeira ordem. Assim, consegue-se também representar as relações transitivas de segunda ordem: **tipo** `TRANSITIVA-2(x)` **é** $\boxed{\text{TRANSITIVA-12: *x}}\ \neg\boxed{\boxed{\text{TRANSITIVA-1: *x}}}$. Claro que `TRANSITIVA-2` $<$ `TRANSITIVA-12`.

Exemplo 1. Eis uma possível organização de alguns tipos não-relacionais, a maioria deles tirados de [Sowa, 1992]:



- o sub-reticulado[2] T_1^{nc} é dado pelo diagrama esquerdo;
- `TIPO` é o topo dos tipos não-relacionais de segunda ordem, isto é `TIPO` $= topo(T_2^{nc})$;
- os tipos `ESPÉCIE`, `GÉNERO`, `FAMÍLIA`, `ORDEM`, `CLASSE`, `REINO`, `COR`, `FORMA`, `MODELO`, `MARCA` pertencem a T_2^{nc} mas não estão relacionados entre si;

[2]Note-se que os conjuntos T_i^{nc} e T_i^{rc} não têm necessariamente que ser reticulados, embora o sejam frequentemente. Pelo menos convém que tenham um topo.

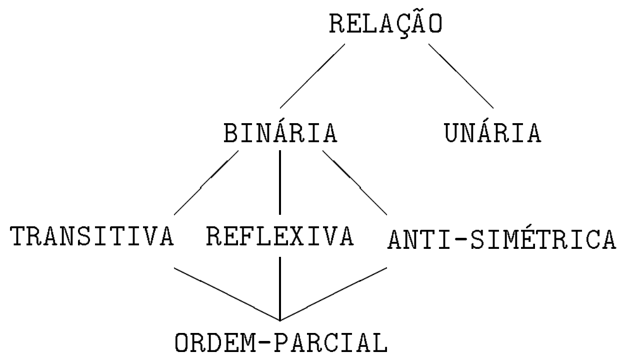
- a sub-hierarquia T_3^{nc} é dada pelo diagrama direito[3].

O Exemplo 13 na página 43 mostra a relação entre os tipos das diferentes ordens, isto é, quais são instâncias de quais.

Para o exemplo seguinte, e para simplificar definições futuras, vamos definir duas variáveis para as ordens mais elevadas tanto nos tipos relacionais como nos não-relacionais.

Definição 4. Os naturais $on, or \in \mathbb{N}_0$ são definidos como $on = max(\{i|T_i^{nc} \neq \emptyset\})$ e $or = max(\{i|T_i^{rc} \neq \emptyset\})$ com a ressalva que $max(\emptyset) = 0$.

Exemplo 2. O topo de T_{or}^{rc} é `RELAÇÃO` com os seguintes subtipos da mesma ordem



em que `TRANSITIVA` poderá ser definido por (ver Exemplo 7 na página 36)

tipo `TRANSITIVA(x)` **é** `[RELAÇÃO: *x]->(ATRIB)->[TRANSITIVIDADE]`

e ter associado a regra do Exemplo 30 na página 58.

2.2 Tipos de Relação

A definição e utilização de hierarquias de tipos de relação é bastante confusa e omissa em [Sowa, 1984]. Tudo o que se diz a esse respeito resume-se às seguintes hipóteses. A primeira trata dos tipos de relação no geral e a segunda aborda o caso de eles formarem uma hierarquia aristotélica, isto é quando a base de conhecimentos tem que conter definições explícitas para todos os tipos excepto um.

Hipótese 3.2.7 (Sowa). A função *tipo* pode ser estendida de modo a associar relações a tipos.

- As relações r e s dizem-se do mesmo tipo se $tipo(r) = tipo(s)$.
- Se r e s são do mesmo tipo, então têm que ter o mesmo número de arcos.

[3]Uma alternativa seria considerar `CATEGORIA`, tal como `COR` e `FORMA` (Exemplo 13 na página 43), uma instância de `CARACTERÍSTICA`. Então este tipo passaria a ser de ordem 4, pelo que `COR` e `FORMA` seriam de terceira ordem.

- Para qualquer conceito c e relação r, $tipo(c) \neq tipo(r)$.
- A ordenação parcial dos tipos também se estende aos tipos de relações, mas os tipos dos conceitos não têm supertipo comum com os tipos das relações.

Hipótese 3.6.13 (Sowa). Numa hierarquia de tipos aristotélica T, existe um tipo `ELO` para uma relação binária. Se t for um tipo de uma relação e $t \neq$ `ELO`, então existe uma definição **relação** $t(a_1, \ldots, a_n)$ **é** u.

Segundo a Hipótese 3.2.1 na página 27, a função *tipo* define o conjunto T dos tipos de conceito e a Hipótese 3.2.5 na página 28 diz que T é um reticulado. Ora como aqui se fala em extensão da função, em supertipos comuns, e novamente em T, a interpretação óbvia parece ser a seguinte: os tipos de conceito e de relação encontram-se todos no mesmo reticulado T, não havendo "misturas" entre os dois géneros de tipos, e sendo `ELO` o topo dos tipos de relação[4]. Assim sendo, faz perfeitamente sentido que os grafos conceptuais de uma base de conhecimentos usem apenas os tipos contidos em T (Hipótese 3.4.5 na página 73), pois estão lá todos os necessários. No entanto, este modo de encarar as hierarquias de tipos levanta vários problemas. A começar pela exigência de T ser um reticulado. Isso implica que, na formalização de Sowa, $\top$ é supertipo de `ELO`, pelo que os tipos de conceito e de relação passam a ter um supertipo em comum ($\top$) contrariamente ao que foi dito. E isto é grave, pois a regra canónica da restrição (Hipótese 3.4.3 na página 76) permite então obter conceitos com tipos de relação. Por exemplo, a partir de $\boxed{\top\text{: João}}$ deriva-se o conceito $\boxed{\text{OBJ: João}}$ que obviamente não faz sentido pois `OBJ` é o tipo de uma relação binária. Uma maneira de resolver esta contradição é dizer que T é um conjunto constituído por dois reticulados, um para os tipos de conceito e outro contendo os tipos de relação.

Mas como está esse segundo reticulado organizado? Como é usado? O que significa exactamente um tipo de relação ser subtipo de outro? Sintoma dos problemas levantados por estas perguntas é o facto da hierarquia de tipos de relação não ser usada no resto de [Sowa, 1984]. Em particular, as regras canónicas de formação não manipulam relações e o Catálogo Conceptual [Sowa, 1984, Apêndice B.3] dá exemplos de tipos de conceito e de relação mas só indica as relações hierárquicas para os primeiros. De facto, uma tal hierarquia conteria algo como `ENTRE` $<$ `ELO`, em que `ENTRE` é ternária enquanto `ELO` é binária. Uma semântica, de preferência simultaneamente precisa e intuitiva, para explicar tais fenómenos continua por definir, tanto quanto eu saiba. Assim compreende-se que ninguém tenha ainda dado uma definição geral de "restrição de relações" — até [Sabah e Vilnat, 1993] omite o problema do número de argumentos diferentes — embora seja por vezes dito que `ELO` pode ser restrito a qualquer outro tipo de relação.

Por estas razões classifico os tipos de relação segundo a sua aridade e ordem. Uma relação de aridade i, também chamada de relação i-ária, tem exactamente i argumentos. Por analogia à noção matemática de "função de ordem superior", uma relação de ordem $i+1$ é uma relação que tem pelo menos um argumento que é uma relação de ordem i. Mais precisamente, e porque os grafos conceptuais são bipartidos (isto é, não se pode

[4]Uma hierarquia aristotélica apenas exige que `ELO` seja o único tipo de relação primitivo, isto é, sem definição explícita. Portanto mesmo que haja outros tipos primitivos, `ELO` continua a ser o tipo de relação básico, isto é, o topo da hierarquia.

ligar relações com relações e conceitos com conceitos), uma relação de ordem $i+1$ tem pelo menos um conceito relacional de ordem i como argumento. Por definição, a ordem máxima dos tipos relacionais é or. Isto significa que tem que haver $or+1$ ordens de tipos de relação, porque senão os conceitos relacionais de ordem or não poderiam ser ligados a nenhum outro conceito.

Como se pode ver pela definição que a seguir se apresenta, os tipos de relação continuam todos numa só hierarquia, mas de novo existem funções que permitem estratificar os tipos segundo a sua *assinatura*, ou seja, ordem e número de argumentos. Assim, para que o tipo t seja um subtipo de t' as suas assinaturas têm que ser compatíveis, nomeadamente t e t' têm a mesma aridade e a ordem de t não pode ser superior à de t'. A razão é a mesma que a dada para os tipos de conceito relacionais: flexibilidade de representação. Por exemplo, pode-se ter `INVERSA-DE-1` $<$ `INVERSA-DE-12` em que `INVERSA-DE-12` tem como argumentos relações de primeira e segunda ordem que são inversas umas das outras, ou pode-se ter simplesmente um único tipo `INVERSA-DE` de ordem máxima que englobe todos os casos.

Hipótese 5. Uma *hierarquia de tipos de relação* é um tuplo $\langle \mathcal{T}_\mathcal{R}, aridade, ordem\rangle$ tal que

- $\mathcal{T}_\mathcal{R}$ é um reticulado finito e não vazio com $topo(\mathcal{T}_\mathcal{R}) = \top_r$ e $base(\mathcal{T}_\mathcal{R}) = \perp_r$;
- $aridade : \mathcal{T}_\mathcal{R} - \{\top_r, \perp_r\} \rightarrow \mathbb{N}$ é uma função total;
- $ordem : \mathcal{T}_\mathcal{R} - \{\top_r, \perp_r\} \rightarrow \mathbb{N}$ é uma função total com imagem $\{1, \ldots, or+1\}$;
- $\forall t, t' \in \mathcal{T}_\mathcal{R} \;\; t \leq t' \Rightarrow t = \perp_r \vee t' = \top_r \vee (aridade(t) = aridade(t') \wedge ordem(t) \leq ordem(t'))$;
- para cada $i \in \mathbb{N}$ o conjunto $\{t \in \mathcal{T}_\mathcal{R} | aridade(t) = i\}$ é um reticulado.

Para facilitar a escrita vamos introduzir mais alguma notação e nomenclatura associada.

Definição 6. O conjunto dos tipos de relação de ordem i é $T_i^r = \{t \in \mathcal{T}_\mathcal{R} | ordem(t) = i\}$ e o reticulado dos tipos de relação de *aridade* i é $T_{\langle i\rangle}^r = \{t \in \mathcal{T}_\mathcal{R} | aridade(t) = i\}$.

Definição 7. [5] Se um tipo de relação é de aridade i diz-se ser i-ária. Relações 1-árias, 2-árias e 3-árias chamam-se unárias, binárias e ternárias, respectivamente.

Definição 8. O número $a \in \mathbb{N}$ representa a *aridade máxima* e define-se como $a = max(\{i | T_{\langle i\rangle}^r \neq \emptyset\})$.

Definição 9. O tipo de relação *universal* é $\top_r$ e o tipo de relação *absurdo* é $\perp_r$.

Gostaria de chamar a atenção para dois pormenores. Em primeiro lugar, a notação introduzida permite especificar de uma forma sucinta o conjunto de todos os tipos de relação de ordem i e aridade j: basta escrever $T_i^r \cap T_{\langle j\rangle}^r$. Em segundo lugar note-se

[5] Esta definição corresponde a parte da Hipótese 3.1.2 na página 50.

que T_i^r e T_i^{rc} têm significados distintos. O primeiro é o conjunto de todos os tipos de *relação* de ordem i, enquanto o segundo é o conjunto de todos os tipos de *conceito* relacionais de ordem i. Como seria de esperar, e como já foi dito de maneira informal, há uma relação entre ambos os conjuntos (Hipótese 68 na página 105).

Em termos formais falta apenas assumir (como na Hipótese 3.2.7 na página 32) que os tipos de conceito e os tipos de relação são disjuntos.

Hipótese 10. $\mathcal{T}_\mathcal{C} \cap \mathcal{T}_\mathcal{R} = \emptyset$.

Passemos agora aos exemplos.

Exemplo 3. Na presente abordagem, o tipo `ELO`, por ser binário, apenas é o topo do reticulado $T_{\langle 2 \rangle}^r$, e não de toda a hierarquia $\mathcal{T}_\mathcal{R}$.

Exemplo 4. Numa base de conhecimentos normal, a maior parte dos elementos de $\mathcal{T}_\mathcal{R}$ provavelmente pertence ao conjunto $T_{\langle 2 \rangle}^r \cap T_1^r$ dos tipos de relação binários de primeira ordem. De facto, a maioria das relações usadas em artigos sobre Estruturas Conceptuais e no Catálogo Conceptual [Sowa, 1984, Apêndice B.3] são binárias e relacionam dois conceitos de primeira ordem. Casos típicos são `AGNT` (agente), `LOC` (local), `OBJ` (objecto), e `PARTE`.

Exemplo 5. O conjunto $T_{\langle 2 \rangle}^r \cap T_1^r$ contém também os tipos de relação `CLASSE` e `CARAC` (característica) que foram apresentados em [Sowa, 1992]. No entanto, estes tipos são fundamentalmente diferentes dos do exemplo anterior, pois relacionam tipos de conceito[6] de ordens diferentes, enquanto ambos os argumentos de `AGNT` e das outras relações são conceitos de primeira ordem. A relação `CLASSE` (definida no Exemplo 31 na página 59) associa uma entidade ao seu tipo, e por isso o segundo argumento é sempre de ordem imediatamente superior à do primeiro. A relação `CARAC` também tem argumentos de ordens diferentes porque foi definida por Sowa à custa de `CLASSE`.

relação `CARAC(x,y)` **é** [$\top_c$: *x] → (ATRIB) → [$\top_c$] → (CLASSE) → [$\top_c$: *y]

Eis dois exemplos de utilização destas relações.

[ANIMAL: Garfield] → (CLASSE) → [ESPÉCIE: gato]

[BOLA] → (CARAC) → [COR: vermelho]

Como estes exemplos deixam antever, não há nenhuma relação hierárquica entre ambos os tipos, isto é, nem `CLASSE` $<$ `CARAC` nem vice-versa (ver o Exemplo 37 na página 65). No entanto, à primeira vista pode parecer que `CARAC` é subtipo de `CLASSE` por estar definido à sua custa. Mas também se pode argumentar que `CLASSE` é uma característica particular de uma entidade, nomeadamente o seu tipo, pelo que devia ser subtipo de `CARAC`.

Exemplo 6. Um tipo de relação de ordem $or + 1$ é `INVERSA-DE`, uma relação binária entre duas outras relações (de ordem inferior).

[6] Não-relacionais, porque senão `CLASSE` e `CARAC` não seriam de primeira ordem.

Exemplo 7. Em [Sowa, 1992], a relação `ATRIB` (atributo) é usada no grafo

[BOLA] → (ATRIB) → [VERMELHO]

o que indica que `ATRIB` é uma relação entre dois conceitos não-relacionais de primeira ordem. No entanto a mesma relação já é usada entre dois conceitos de géneros diferentes (um relacional e outro não-relacional) em [Sowa, 1993]: o exemplo concreto é o grafo [RELAÇÃO: maior-que] → (ATRIB) → [TRANSITIVIDADE]. Se a base de conhecimentos contivesse apenas estes dois grafos, a classificação de `ATRIB` em $\mathcal{T}_{\mathcal{R}}$ teria que ser *aridade*(`ATRIB`) = *ordem*(`ATRIB`) = 2, pois pelo menos um dos argumentos é um conceito relacional de primeira ordem.

2.3 Marcadores

Como se viu na Secção 1.2 na página 16, um conceito é constituído por um tipo e um referente. É este último que contém a maior parte da informação, pois é o referente que descreve o indivíduo representado, enquanto o tipo apenas indica a classe a que esse indivíduo pertence. O referente básico é o marcador individual.

Hipótese 3.3.1 (Sowa). Existe um conjunto $I = \{\#1, \#2, \#3, \ldots\}$ cujos elementos se chamam *marcadores individuais*. A função *referente* pode ser aplicada a qualquer conceito c:

- *referente*(c) é um marcador individual de I ou o *marcador genérico* $*$.
- Quando *referente*(c) pertence a I diz-se que c é um *conceito individual*.
- Quando *referente*(c) é $*$, então diz-se que c é um *conceito genérico*.

Para além destes marcadores, só há dois outros tipos de referentes definidos formalmente por Sowa [1984]: grafos (ver Secção 4.2 na página 52) e conjuntos de marcadores. No entanto, as definições destes dois casos são bastante menos precisas que a hipótese anterior. Para além disso, não se encontra dito em lado algum o que a função *referente* é suposta devolver nesses casos, e nem as regras canónicas de formação nem as regras de inferência se aplicam a esse referentes. Para complicar ainda mais as coisas, muitos autores usam e abusam dos referentes no intuito de dar maior expressividade às Estruturas Conceptuais, aproximando-as do poder expressivo das línguas naturais. Como se pode ver facilmente em qualquer colecção de artigos sobre grafos conceptuais existem referentes para denotar quantificadores generalizados ("todo", "alguns", etc.), medidas (tempo, comprimento, etc.), indexicais ("eu", etc.), conjuntos (distributivos, respectivos, colectivos e outros), sequências (listas de referentes), cadeias de caracteres, etc. Embora a semântica de alguns deles seja relativamente fácil de definir (como no caso do quantificador universal "todo"), e embora tenha havido algum trabalho por parte de alguns investigadores[7] que infelizmente não foi adoptado pela comunidade em

[7] Por exemplo, [Tjan *et al.*, 1992; Gardiner *et al.*, 1992; Pfeiffer e Hartley, 1992] tratam dos referentes que sejam conjuntos.

geral, o caos formal mantém-se. O próprio Sowa reconhece o problema e no seu novo livro [Sowa, 1995] dá um tratamento mais sistemático e aprofundado aos referentes.

Como se pode pois constatar a partir do trabalho realizado por dezenas de investigadores nesta primeira década de existência da TEC, um tratamento completo e rigoroso dos referentes está muito para além do âmbito desta dissertação, envolvendo questões filosóficas, ontológicas e sobre a própria lógica subjacente à semântica dos referentes. Nesse sentido abordar-se-ão apenas os referentes mais simples, nomeadamente os marcadores individuais.

Já foi dito informalmente, e a Secção 6.2 na página 103 tornará o assunto mais claro, que os tipos de ordem superior denotam conjuntos de tipos de ordem inferior. Ou seja, é preciso tratar os próprios tipos (classes de indivíduos) como indivíduos. Para usar os grafos conceptuais como meta-linguagem, temos que ter marcadores que correspondam aos indivíduos do universo de discurso. Portanto, se os tipos também são indivíduos, o conjunto único de marcadores individuais da Hipótese 3.3.1 na página anterior tem que ser estendido para múltiplos conjuntos, um por cada ordem. Para além disso, os marcadores individuais devem reflectir a ordenação hierárquica dos tipos a que dizem respeito. O método mais simples e natural para obter esta correspondência entre tipos e marcadores consiste em deixar os tipos serem também marcadores individuais, pelo que passam a aparecer num conceito tanto à esquerda como à direita do ":". De facto, visto um tipo ser um conjunto de entidades, o conceito $\boxed{t : m}$ corresponde à afirmação $m \in t$. Assim, se t for um tipo de ordem superior, então tem como elementos outros tipos, e portanto nada mais natural do que o marcador m ser um tipo.

Além disto, assumo, tal como [Mugnier e Chein, 1993b], que existe um marcador absurdo o qual, juntamente com o marcador genérico, permite-nos classificar os marcadores em reticulados, seguindo a proposta de [Schmidt e Kocura, 1993].

Hipótese 11. Os *marcadores* $\mathcal{M} = \bigcup_{i=0}^{on} T_i^{nc} \cup T_{\leq or}^{r} \cup \{*, \bar{*}\}$ formam um reticulado tal que

- $topo(\mathcal{M}) = *$ e $base(\mathcal{M}) = \bar{*}$;
- T_0^{nc} é um conjunto finito;
- para quaisquer $m, m' \in \mathcal{M}$, se $m \leq m'$ então tem-se um dos seguintes casos:
  - $m = \bar{*}$ ou $m' = *$;
  - $m, m' \in T_i^{nc}$ com $0 < i \leq on$;
  - $m \in T_i^r \cap T_{\langle j \rangle}^r$ e $m' \in T_k^r \cap T_{\langle j \rangle}^r$ com $i \leq k \leq or$.

Definição 12. O *marcador genérico* é $*$ e o *marcador absurdo* é $\bar{*}$. Todos os outros marcadores chamam-se *marcadores individuais*.

Repare-se que $\mathcal{M}$ não é mais que a união dos conjuntos parcialmente ordenados dos tipos de conceito não-relacionais e dos tipos de relação, juntamente com um conjunto de indivíduos mutuamente incomparáveis, aos quais se acrescenta um topo (o marcador genérico) e uma base (o marcador absurdo). Esse conjunto de indivíduos designou-se por T_0^{nc} porque podem ser encarados como tipos de ordem zero, no fundo da estratificação por ordens. No entanto, não são verdadeiramente tipos porque não podem aparecer à esquerda do ":" nos conceitos. É de notar igualmente que nem todos os tipos de relação podem servir de marcadores, nomeadamente os de ordem $or + 1$. A razão é simples. Para um tipo de relação aparecer no referente de um conceito, é porque o tipo desse conceito é um tipo de conceito relacional. Ora a ordem máxima de um tipo relacional é or pelo que será também essa a ordem máxima do tipo de relação que aparece no referente.

A Hipótese 11 na página anterior é omissa em relação à forma dos elementos de T_0^{nc}. Podia-se adoptar a convenção da Hipótese 3.3.1 na página 36, que consiste simplesmente em numerá-los `#1`, `#2`, etc. No entanto, para facilitar a leitura, seguir-se-á neste texto o uso corrente de dar etiquetas descritivas (tais como `#João`) aos marcadores. Estas etiquetas devem ser únicas. Se por exemplo a base de conhecimentos contiver informação sobre duas pessoas, ambas chamadas "João", deverá utilizar-se duas etiquetas `#João1` e `#João2`. Quando o marcador é um tipo, é praticamente inevitável usar uma etiqueta em vez de um número. Neste caso adoptou-se também a grafia com `#` e letras minúsculas para realçar o facto do tipo estar a ser usado como marcador individual (ver Exemplo 9 na página 42).

Capítulo 3

Conceitos e Relações

Como foi dito no capítulo anterior, os marcadores representam entidades e os tipos agrupam-nas em conjuntos. As relações entre entidades normalmente dependem dessa classificação (por exemplo, `AGNT` relaciona apenas seres animados com acções). A mesma entidade pode pois fazer parte de várias relações, conforme a sua classificação. Nesse sentido, a representação de entidades não vai ser feita unicamente com marcadores, mas sim com pares tipo-marcador, os chamados *conceitos*. Uma *relação n-ária* será então formada pelo seu tipo e por n conceitos, os seus argumentos. No entanto, o agrupamento e o relacionamento das entidades têm em conta o género e a ordem dessas entidades (por exemplo, um tipo de conceito não-relacional de segunda ordem apenas pode denotar um conjunto de tipos não-relacionais de primeira ordem). Dir-se-á então que um conceito está *bem-formado* se a entidade representada pelo marcador puder ser um elemento do conjunto representado pelo tipo do conceito. De modo análogo, uma relação estará *bem-formada* se todos os seus argumentos estiverem bem-formados e se puderem pertencer à denotação do tipo da relação.

Neste capítulo, os conceitos e relações são tratados em secções separadas mas estruturadas de forma idêntica. Primeiro dá-se uma definição geral de conceito (e de relação) como tuplo e apresentam-se algumas funções auxiliares para extrair os elementos do tuplo. No caso de um conceito, esses elementos são um tipo de conceito e um marcador, enquanto para uma relação os elementos são um tipo de relação e um ou mais conceitos. De seguida, as características dos tipos são estendidas aos conceitos e às relações. Assim, sabendo a ordem e género do tipo e do marcador, pode-se calcular a ordem e o género do conceito. Por sua vez, sabendo quantos argumentos tem uma relação e qual o género e a ordem de cada um, e tendo a aridade e a ordem do tipo da relação, pode-se calcular a ordem e a aridade de toda a relação. Isto leva-nos a uma definição formal e facilmente computável dos conceitos e relações bem-formados: são aqueles em que nem a ordem nem o género (ou aridade) estão indefinidos. Finalmente mostra-se como as hierarquias dos tipos e dos marcadores induzem hierarquias sobre os conceitos e relações, o que terá vantagens para as regras canónicas de formação (Secção 5.3 na página 76).

3.1 Conceitos

Na TEC original os tipos e marcadores são definidos à custa dos conceitos, nomeadamente através das funções *tipo* (Hipótese 3.2.1 na página 27) e *referente* (Hipótese 3.3.1 na página 36). Isto significa no entanto que a definição de conceito é *implícita*. De facto, nada em [Sowa, 1984] impede um conceito de conter outras coisas para além do tipo e do marcador. Por esta razão, e porque penso serem os tipos e marcadores entidades mais básicas do que os conceitos, segui a ordem inversa. Isto permite ter uma definição *explícita* de conceitos como pares ordenados, cujos elementos são estraídos pelas funções auxiliares *tipo* e *referente*.

Definição 13. O conjunto de todos os *conceitos* é $\mathcal{C} = \mathcal{T}_\mathcal{C} \times \mathcal{M}$.

Definição 14. A função total *tipo*: $\mathcal{C} \to \mathcal{T}_\mathcal{C}$ devolve para cada conceito o seu tipo: $tipo(\langle t, m \rangle) = t$.

Definição 15. A função total *referente*: $\mathcal{C} \to \mathcal{M}$ devolve para cada conceito o seu referente, um marcador: $referente(\langle t, m \rangle) = m$.

Os conceitos podem ser classificados segundo o referente. A seguinte definição apenas redefine e estende a nomenclatura introduzida pela Hipótese 3.3.1 na página 36.

Definição 16. Um conceito c diz-se *genérico* se e só se $referente(c) = *$. É designado por conceito *individual* se e só se $referente(c)$ for um marcador individual, e é um conceito *absurdo* se e só se $referente(c) = \bar{*}$.

De acordo com a "perspectiva" dada pelo tipo, um conceito representa simplesmente a entidade correspondente ao marcador. Assim, um conceito individual representa uma entidade particular, o conceito absurdo representa uma entidade inexistente — isto é, que não faz parte do domínio, razão pela qual os conceitos absurdos são sempre falsos (Capítulo 6 na página 91) — e um conceito genérico representa uma entidade desconhecida.

Neste último caso é pois possível que o género (relacional ou não-relacional) do conceito, que não é mais do que o género da entidade representada, seja desconhecido. Por convenção, o género de entidades inexistentes também é desconhecido. Claro que se souber a que classe pertence uma entidade, o género já poderá ser determinado com precisão. Por outro lado, se o género do marcador for incompatível com o do tipo, teremos um conceito de género indefinido. Resumindo, o género de um conceito pode ser relacional, não-relacional, desconhecido, ou indefinido. Em vez de acrescentar dois símbolos especiais aos já existentes `rel` e `nonrel`, sigo uma via alternativa: o género de um conceito é o conjunto de todos os géneros possíveis das entidades que esse conceito possa representar. Assim, o conjunto vazio corresponde ao género indefinido e o conjunto $\{\texttt{rel}, \texttt{nonrel}\}$ é o género desconhecido. O conjunto associado a um dado conceito é determinado pela função *género*. A sua definição é extremamente simples: calcula-se os géneros possíveis segundo o tipo e o marcador e depois intersecta-se esses dois conjuntos, o que equivale a tentar compatibilizá-los.

Definição 17. Seja $c = \langle t, m \rangle$ um conceito. Então a função total *género*: $\mathcal{C} \rightarrow \wp(\{\texttt{rel}, \texttt{nonrel}\})$ que devolve para cada conceito os seus géneros possíveis define-se como $género(c) = f(t) \cap g(m)$ em que

$$f(t) = \begin{cases} \{\texttt{rel}, \texttt{nonrel}\} & \text{se } t \in \{\top_c, \bot_c\} \\ \{género(t)\} & \text{caso contrário} \end{cases} \qquad g(m) = \begin{cases} \{\texttt{rel}, \texttt{nonrel}\} & \text{se } m \in \{*, \bar{*}\} \\ \{\texttt{rel}\} & \text{se } m \in T^r_{\leq or} \\ \{\texttt{nonrel}\} & \text{caso contrário} \end{cases}$$

A abordagem adoptada para a ordem de um conceito é igual à seguida para o género. A função *ordem* devolve para cada conceito o conjunto das ordens de todas as entidades que ele possa representar. De novo, o conjunto vazio indica que o conceito, e portanto a sua ordem, está indefinido. O cálculo da função faz-se também por intersecção das contribuições dos componentes do conceito, o tipo e o referente. Repare-se que visto a ordem ser dada por um número natural, estes conjuntos são subconjuntos de $\mathbb{N}_0$ porque existem entidades de ordem zero, os indivíduos. Em particular são intervalos, como se verá de seguida. Por esta razão, utilizarei a notação $[x, \ldots, y]$ para representar o conjunto de todos os naturais de x a y, inclusive, sendo $x \leq y$. Vejamos então como são calculados estes conjuntos e porque são intervalos.

Seja x a ordem máxima possível, isto é $x = max(\{on, or\})$. Se o marcador for $*$ ou $\bar{*}$, então, por convenção, as ordens possíveis são $[0, \ldots, x]$ porque a entidade é desconhecida ou inexistente. Para as entidades concretas, a ordem é dada pelo marcador individual. Quanto ao tipo, se ele for $\top_c$ ou $\bot_c$ a entidade pode ser qualquer uma: um indivíduo de ordem zero, um tipo de relação de ordem não superior a or, ou um tipo de conceito não-relacional de ordem não superior a on. Portanto, a ordem será de novo $[0, \ldots, x]$. Por outro lado, se o tipo do conceito for relacional e de ordem i, então a entidade denotada terá que ser uma relação de ordem não superior a i. Neste caso o intervalo é pois $[1, \ldots, i]$ porque não há relações de ordem zero. Finalmente, se o tipo for não-relacional de ordem i, a entidade representada pelo marcador só poderá ser de ordem $i - 1$. Juntando isto tudo tem-se a seguinte definição formal.

Definição 18. Seja $c = \langle t, m \rangle$ um conceito e seja $x = max(\{or, on\})$. Então a função total *ordem*: $\mathcal{C} \rightarrow \wp(\mathbb{N}_0)$ que devolve para cada conceito as suas ordens possíveis define-se como $ordem(c) = f(t) \cap g(m)$ em que

$$f(t) = \begin{cases} [0, \ldots, x] & \text{se } t \in \{\top_c, \bot_c\} \\ [1, \ldots, i] & \text{se } t \in T^{rc}_i \\ [ordem(t) - 1] & \text{caso contrário} \end{cases} \qquad g(m) = \begin{cases} [0, \ldots, x] & \text{se } m \in \{*, \bar{*}\} \\ [i] & \text{se } m \in T^{nc}_i \\ [i] & \text{se } m \in T^r_i \end{cases}$$

Visto o género e a ordem de um conceito serem determinados a partir dos seus componentes, a especificação das hierarquias de tipos por parte do engenheiro do conhecimento reveste-se de grande importância. Se os tipos forem organizados de maneira intuitiva, então as funções *género* e *ordem* permitirão, de forma algorítmica, determinar os conceitos que fazem sentido, segundo essa mesma intuição. Vejamos alguns exemplos.

Exemplo 8. O conceito $c = \boxed{\text{GATO: }\texttt{\#Garfield}}$ representa um indivíduo. Será pois um conceito não-relacional de ordem zero. De facto, aplicando as definições formais tem-se

$género(c) = \{\texttt{nonrel}\} \cap \{\texttt{nonrel}\} = \{\texttt{nonrel}\}$ e $ordem(c) = [1-1] \cap [0] = [0]$, porque $ordem(\texttt{GATO}) = 1$ e $género(\texttt{GATO}) = \texttt{nonrel}$ (Exemplo 1 na página 31) e $\texttt{\#Garfield} \in T_0^{nc}$.

Exemplo 9. O conceito $c = \boxed{\text{BINÁRIA: } \texttt{\#agnt}}$ representa uma relação de primeira ordem. Tem-se pois $género(c) = \{\texttt{rel}\} \cap \{\texttt{rel}\} = \{\texttt{rel}\}$ e $ordem(c) = [1, \ldots, or] \cap [1] = [1]$ visto $género(\texttt{BINÁRIA}) = \texttt{rel}$ e $ordem(\texttt{BINÁRIA}) = or$ (Exemplo 2 na página 32) e $\texttt{AGNT} \in T_1^r$ (Exemplo 4 na página 35).

Exemplo 10. O conceito $c = \boxed{\text{GATO: } \texttt{\#espécie}}$ é não-relacional mas é de ordem indefinida. Formalmente, visto $\texttt{GATO} \in T_1^{nc}$ e $\texttt{ESPÉCIE} \in T_2^{nc}$ (Exemplo 1 na página 31) tem-se $ordem(c) = [0] \cap [2] = \emptyset$. No entanto, $género(\texttt{GATO}) = género(\texttt{ESPÉCIE}) = \texttt{nonrel}$ implica $género(c) = \{\texttt{nonrel}\}$.

Exemplo 11. O género de $c = \boxed{\text{BINÁRIA: } \texttt{\#gato}}$ está indefinido. De facto, $género(c) = \{\texttt{rel}\} \cap \{\texttt{nonrel}\} = \emptyset$. A ordem não coloca problemas: $ordem(c) = [1, \ldots, or] \cap [1] = [1]$.

Exemplo 12. Nos exemplos anteriores os conceitos são individuais. Vejamos agora o conceito absurdo $\boxed{\bot_c : \bar{*}}$ e o conceito genérico $\boxed{\top_c : *}$. Em nenhum caso são impostas quaisquer restrições quanto ao género ou ordem, pelo que a função *género* devolve para ambos os conceitos o conjunto $\{\texttt{rel}, \texttt{nonrel}\}$ e a função *ordem* retorna $[0, \ldots, max(\{on, or\})]$. Repare-se que como não se sabe se os conceitos são relacionais ou não, a sua ordem máxima é a maior das ordens relacionais e das ordens não-relacionais, e a ordem mínima é zero, a ordem dos indivíduos.

Formalizando esta noção do sentido de um conceito, dir-se-á que um conceito está bem-formado se nem o seu género nem a sua ordem estiverem indefinidos.

Definição 19. O conjunto dos conceitos *bem-formados* é

$$\mathcal{C}_f = \{c \in \mathcal{C} | género(c) \neq \emptyset \wedge ordem(c) \neq \emptyset\}$$

Esta definição impõe condições extremamente simples, donde fáceis de calcular computacionalmente, sobre os conceitos bem-formados. Trata-se portanto de uma típica definição "gera-e-testa" que poderia ser adoptada por uma implementação para verificar os conceitos introduzidos pelo utilizador. Devido ao raciocínio usado para obter as funções *género* e *ordem*, a definição formal de $\mathcal{C}_f$ corresponde à noção intuitiva de boa-formação: um conceito $\boxed{t : m}$ estará bem-formado se a entidade representada por m puder pertencer ao conjunto denotado por t. Segundo a denotação dos tipos informalmente apresentada no capítulo anterior obtém-se as seguintes possibilidades:

- se t for um tipo não-relacional de ordem n, então m é um tipo não-relacional de ordem $n-1$;
- se t for um tipo relacional de ordem n, então m é um tipo de relação de ordem não superior a n;
- se m for $*$ então a entidade é desconhecida pelo que t pode ser qualquer tipo;

- o tipo universal $\top_c$ representa o conjunto de todas as entidades pelo que m pode ser qualquer marcador.

Os casos que faltam, $t = \bot_c$ ou $m = \bar{*}$, são tratados por convenção. O marcador absurdo representa uma entidade que não faz parte do domínio, e o tipo absurdo denota o conjunto vazio. Assim, os conceitos da forma $\boxed{\bot_c : m}$ e $\boxed{t\colon \bar{*}}$ são sempre falsos. Como a entidade representada pelo referente não pode pertencer ao conjunto dado pelo tipo, estes conceitos não deveriam estar bem-formados. No entanto, um formalismo de representação de conhecimentos que não seja capaz de representar afirmações falsas tem menor interesse, razão pela qual os conceitos bem-formados deverão ser todos os casos aqui descritos. A proposição seguinte confirma que assim é, dando deste modo uma visão construtiva dos conceitos bem-formados.

Proposição 20. *O conjunto dos conceitos* bem-formados *é*

$$\mathcal{C}_f = \bigcup_{i=1}^{on}(T_i^{nc} \times T_{i-1}^{nc}) \cup \bigcup_{i=1}^{or}(T_i^{rc} \times T_{\leq i}^{r}) \cup \mathcal{T_C} \times \{*, \bar{*}\} \cup \{\top_c, \bot_c\} \times \mathcal{M}$$

Demonstração. Segundo a definição, um conceito c não está bem formado se $\mathit{género}(c) = \emptyset$ ou $\mathit{ordem}(c) = \emptyset$. Olhando para a Definição 17 na página 41, só há duas maneiras de o género de um conceito não estar definido: o género do tipo é `rel` e o do referente é `nonrel` ou vice-versa. Ou seja, o conjunto dos conceitos de género indefinido é $T_{\leq or}^{rc} \times T_{\leq on}^{nc} \cup T_{\leq on}^{nc} \times T_{\leq or}^{rc}$. Quanto à ordem, segundo a Definição 18 na página 41, há quatro possibilidades de se obter o conjunto vazio, conforme o género do tipo e do referente do conceito. No entanto, já só interessam os casos em que tipo e referente são do mesmo género. Se o tipo for relacional e o referente for uma relação, basta a sua ordem ser estritamente maior que a do tipo para a ordem do conceito estar indefinida. Se tanto o tipo como o referente forem não-relacionais, e se a ordem do referente for i e a do tipo for j, basta ter $i \neq j - 1$. O conjunto dos conceitos mal-formados no que diz respeito à ordem é portanto $\bigcup_{i=1}^{or} T_i^{rc} \times T_{>i}^{r} \cup \bigcup_{i=1}^{on} T_i^{nc} \times (T_{\geq i}^{nc} \cup T_{<i-1}^{nc})$. Se retirarmos este conjunto e o anterior ao conjunto $\mathcal{C}$ de todos os conceitos, obteremos o resultado pretendido. $\square$

Exemplo 13. No Exemplo 1 na página 31 foram apresentados tipos de primeira, segunda e terceira ordem. Os seguintes conceitos bem-formados, se introduzidos como factos na base de conhecimentos, servem para estabelecer as relações entre as várias ordens, pois $\boxed{t\colon \#t'}$ significa que t' é uma entidade de tipo t.

- $\boxed{\text{ESPÉCIE: }\texttt{\#gato}}$ $\boxed{\text{FAMÍLIA: }\texttt{\#felino}}$ $\boxed{\text{REINO: }\texttt{\#animal}}$
- $\boxed{\text{COR: }\texttt{\#vermelho}}$
- $\boxed{\text{FORMA: }\texttt{\#rectângulo}}$ $\boxed{\text{FORMA: }\texttt{\#quadrado}}$
- $\boxed{\text{MARCA: }\texttt{\#chevrolet}}$ $\boxed{\text{MODELO: }\texttt{\#camaro}}$
- $\boxed{\text{CARACTERÍSTICA: }\texttt{\#cor}}$ $\boxed{\text{CARACTERÍSTICA: }\texttt{\#forma}}$
- $\boxed{\text{CATEGORIA: }\texttt{\#espécie}}$ $\boxed{\text{CATEGORIA: }\texttt{\#família}}$ $\boxed{\text{CATEGORIA: }\texttt{\#reino}}$
- $\boxed{\text{CATEGORIA: }\texttt{\#marca}}$ $\boxed{\text{CATEGORIA: }\texttt{\#modelo}}$

Eis mais uma definição que será útil na especificação de relações bem-formadas.

Definição 21. Para cada $i > 0$, os conceitos *relacionais* de ordem i são $RC_i = \{c \in \mathcal{C} | \texttt{rel} \in género(c) \wedge i \in ordem(c)\}$ e os conceitos *não-relacionais* de ordem i são $NRC_i = \{c \in \mathcal{C} | \texttt{nonrel} \in género(c) \wedge i \in ordem(c)\}$.

Há duas coisas a salientar em relação a esta definição. Primeiro, e como foi dito anteriormente, o mesmo conceito pode pertencer a vários RC_i e/ou NRC_i. Um exemplo típico é $\boxed{\top_c\texttt{: *}}$. Segundo, pela própria natureza da definição, um conceito (não-)relacional de ordem i é bem-formado. Nomeadamente, também poderia ter definido $\mathcal{C}_f = NRC_{\leq on} \cup RC_{\leq or}$.

Para finalizar o tratamento dos conceitos, mostro como a ordenação dos tipos e referentes induz uma hierarquia sobre os conceitos.

Definição 22. Um conceito c é uma *restrição* (ou *especialização*) do conceito c' e inversamente c' é uma *generalização* de c (escrevendo-se $c \leq c'$) se e só se $tipo(c) \leq tipo(c')$ e $referente(c) \leq referente(c')$.

Proposição 23. *$\mathcal{C}$ é um reticulado.*

Demonstração. A afirmação é trivial: visto $\mathcal{T_C}$ e $\mathcal{M}$ serem ambos reticulados (Hipótese 1 na página 30 e Hipótese 11 na página 37), o seu produto cartesiano $\mathcal{C}$ também o é. Obviamente que a ordem parcial do reticulado de conceitos não é mais do que a relação de especialização. □

É óbvio que o topo de $\mathcal{C}$ é $\boxed{\top_c\texttt{: *}}$ e a base é $\boxed{\bot_c\texttt{: }\bar{*}}$. Ao contrário de $\mathcal{C}$, o conjunto $\mathcal{C}_f$ não é um reticulado.

Exemplo 14. Sejam $c = \boxed{\top_c\texttt{: \#João}}$ e $c' = \boxed{\text{RELAÇÃO: *}}$. Então $c \wedge c' = \boxed{\text{RELAÇÃO: \#João}}$ não é um conceito bem-formado. Mais concretamente, c e c' têm duas especializações bem-formadas incomparáveis em comum, nomeadamente $\boxed{\bot_c\texttt{: \#João}}$ e $\boxed{\text{RELAÇÃO: }\bar{*}}$.

3.2 Relações

Tal como aconteceu com os conceitos, e pelas mesmas razões, a sequência de definições seguida por Sowa é invertida. Primeiro, e baseado na Hipótese 3.1.2 na página 50, defino explicitamente as relações como tuplos, em que o primeiro elemento é o tipo da relação e os restantes elementos (os argumentos) são conceitos. De seguida, as funções auxiliares *tipo* (baseada na Hipótese 3.2.7 na página 32), *arg* e *args* extraiem esses elementos. Esta última, ao juntar todos os argumentos de uma relação num só conjunto, elimina os conceitos duplicados.

Definição 24. O conjunto de todas as *relações* é

$$\mathcal{R} = \{\langle t, c_1, \ldots, c_i \rangle | t \in \mathcal{T_R} \wedge i > 0 \wedge \forall j \in \{1, \ldots, i\} \;\; c_j \in \mathcal{C}\}$$

Definição 25. A função total *tipo*: $\mathcal{R} \to \mathcal{T}_{\mathcal{R}}$ devolve para cada relação o seu tipo: $tipo(\langle t, c_1, \ldots, c_i \rangle) = t$.

Definição 26. A função parcial *arg*: $\mathcal{R} \times \mathbb{N} \to \mathcal{C}$ devolve para uma dada relação r e um dado número natural j o j-ésimo *argumento* de r:

$$arg(r, j) = c_j \Leftrightarrow r = \langle t, c_1, \ldots, c_j, \ldots, c_i \rangle$$

A função total *args*: $\mathcal{R} \to \wp(\mathcal{C})$ associa a cada relação os seus argumentos:

$$args(r) = \{c_1, \ldots, c_i\} \Leftrightarrow r = \langle t, c_1, \ldots, c_i \rangle$$

Assim como estendi as funções associadas aos tipos de conceito para os conceitos em geral, também irei adaptar às relações as funções *aridade* e *ordem* que definiam a hierarquia de tipos de relação. Quanto à aridade de uma relação, a intuição por trás da definição formal que se segue é simples: uma relação tem uma única aridade e não um conjunto delas; se a relação estiver bem formada então a aridade é simplesmente o número de argumentos, senão *aridade* devolve um valor especial que se convencionou ser zero, visto uma relação ter sempre pelo menos um argumento.

Definição 27. Seja $r = \langle t, c_1, \ldots, c_i \rangle$ uma relação. Então a função total *aridade*: $\mathcal{R} \to \mathbb{N}_0$ que devolve para cada relação a sua aridade define-se como

$$aridade(r) = \begin{cases} i & \text{se } t \in \{\top_r, \bot_r\} \vee aridade(t) = i \\ 0 & \text{caso contrário} \end{cases}$$

Um conceito representa uma única entidade, pelo que a sua ordem é a ordem dessa entidade. No entanto uma relação tem como argumentos várias entidades. Por convenção, a ordem de uma relação será a maior das ordens dessas entidades. Mas as entidades não-relacionais não são relevantes para o cálculo da ordem. Por exemplo, uma relação binária em que um dos argumentos é um conceito não-relacional de ordem 5 e o outro é um conceito relacional de ordem 2, será uma relação de ordem 2 e não 5. A razão dada no capítulo anterior é a analogia com a noção de "função de ordem superior": uma função de ordem n tem pelo menos um argumento que é uma função de ordem $n-1$ e não tem nenhum argumento que seja uma função de ordem n ou mais. Substituindo a palavra "função" por "relação" obtém-se a definição de relação de ordem superior. Mas existe outra razão para ignorar os argumentos não-relacionais. Considere-se por exemplo o tipo de relação `CLASSE` do Exemplo 5 na página 35. Dá jeito poder escrever $\boxed{\text{GATO: \texttt{\#Garfield}}} \to (\text{CLASSE}) \to \boxed{\text{ESPÉCIE: \texttt{\#gato}}} \to (\text{CLASSE}) \to \boxed{\text{CATEGORIA: \texttt{\#espécie}}}$. Ou seja, `CLASSE` é usado para relacionar qualquer entidade não-relacional com o seu tipo, independentemente da ordem.

Resumindo, a ordem de uma relação é a maior das ordens dos argumentos relacionais. Se a relação só tiver argumentos não-relacionais, então por convenção será de ordem zero. Visto a ordem de uma relação se basear na dos seus argumentos, que são conceitos, ela poderá ser desconhecida. Neste caso, calcula-se o intervalo das ordens possíveis. Deste modo pode-se representar de novo a ordem indefinida como o intervalo vazio. Mas como é calculado esse intervalo de ordens? Imaginemos uma relação binária entre conceitos relacionais, um de ordem $[1, \ldots, 4]$ e outro de ordem $[2, 3]$. A relação nunca poderá ser de ordem inferior a 2 por causa do segundo argumento, mas poderá chegar à quarta ordem por causa do primeiro argumento. A partir desta observação chega-se ao algoritmo pretendido. Para cada conceito, as funções li e ls calculam respectivamente o limite inferior e superior das ordens possíveis. Se o conceito for não-relacional, esses limites serão zero. De seguida, guarda-se em x o maior dos valores devolvidos por li, e y será o maior dos limites superiores. Deste modo, a ordem relação terá que estar no intervalo $[x, \ldots, y]$ segundo os argumentos. Passando ao tipo da relação, a função o calcula as ordens por ele permitidas. Os tipos universal e absurdo permitem relacionar quaisquer conceitos. O intervalo irá então de 0, a ordem dos conceitos não-relacionais, a or, a maior ordem dos conceitos relacionais. Caso contrário, isto é, se o tipo da relação tiver uma ordem específica n, então o intervalo só poderá ir até $n - 1$. A ordem da relação será então determinada pela intersecção dos intervalos obtidos. No entanto, caso um dos argumentos tenha ordem indefinida (ou seja, se não estiver bem-formado), a ordem da relação também estará indefinida.

Definição 28. A função total $ordem\colon \mathcal{R} \rightarrow \wp(\mathbb{N})$ que devolve para cada relação r de tipo t as suas ordens possíveis define-se como

$$ordem(r) = \begin{cases} \emptyset & \text{se } \exists c \in args(r) \;\; c \notin \mathcal{C}_f \\ o(t) \cap [x, \ldots, y] & \text{caso contrário} \end{cases}$$

em que $x = max(\{li(c) | c \in args(r)\})$ e $y = max(\{ls(c) | c \in args(r)\})$ sendo

$$o(t) = \begin{cases} [0, \ldots, or] & \text{se } t \in \{\top_r, \bot_r\} \\ [0, \ldots, ordem(t) - 1] & \text{caso contrário} \end{cases}$$

$$li(c) = \begin{cases} 0 & \text{se } \texttt{nonrel} \in g\acute{e}nero(c) \\ min(ordem(c)) & \text{caso contrário} \end{cases}$$

$$ls(c) = \begin{cases} 0 & \text{se } \texttt{nonrel} \in g\acute{e}nero(c) \\ max(ordem(c)) & \text{caso contrário} \end{cases}$$

Tal como acontecia com os conceitos, estas funções permitem distinguir as relações bem-formadas das mal-formadas. Intuitivamente, numa relação bem-formada o tuplo representado pelos argumentos poderá pertencer à denotação do tipo da relação. Assim, o número de argumentos tem que ser igual à aridade do tipo e cada argumento relacional tem que ser de ordem inferior à permitida pelo tipo. Claro que a relação está mal-formada se algum dos seus argumentos o estiver. Como para os conceitos, a definição formal é sucinta, facilmente computável, e corresponde à intuição devido à própria maneira como foram definidas as funções *aridade* e *ordem*.

Definição 29. O conjunto das relações *bem-formadas* é

$$\mathcal{R}_f = \{r \in \mathcal{R} | aridade(r) > 0 \wedge ordem(r) \neq \emptyset\}$$

Proposição 30. *O conjunto $\mathcal{R}_f$ das relações* bem-formadas *contém apenas as relações $\langle t, c_1, \ldots, c_i \rangle$ que satisfazem uma das seguintes condições:*

- $t \in T^r_{\langle i \rangle} \wedge \forall j \in \{1, \ldots, i\}\ \ c_j \in \mathcal{C}_f \wedge (\textit{género}(c_j) = \{\texttt{rel}\} \Rightarrow c_j \notin RC_{\leq ordem(t)})$;
- $t \in \{\top_r, \bot_r\} \wedge \forall j \in \{1, \ldots, i\}\ \ c_j \in \mathcal{C}_f$.

Demonstração. O estilo da demonstração é idêntico ao da Proposição 20 na página 43: constroi-se o conjunto das relações mal-formadas e imediatamente vê-se que o seu complementar é o conjunto pretendido. Em particular, para uma relação ter aridade zero, só há uma possibilidade segundo a Definição 27 na página 45: o tipo não é o tipo universal nem o tipo absurdo e a sua aridade é diferente do número de argumentos da relação. Ou seja, o conjunto das relações mal-formadas quanto à aridade é $\{\langle t, c_1, \ldots, c_i \rangle | t \in T^r_{\langle j \rangle} \wedge j \neq i\}$. Quanto à ordem (Definição 28 na página oposta), uma das possibilidades de obter o conjunto vazio é um dos argumentos não estar bem-formado. Outra possibilidade é a ordem máxima de um dos argumentos relacionais (que são conceitos e portanto podem ter várias ordens) ser superior à ordem do tipo da relação. De facto repare-se que os conceitos não-relacionais contribuem sempre com 0 para a ordem da relação e portanto são sempre compatíveis com o conjunto $[0, \ldots, ordem(t) - 1]$ contribuido pelo tipo da relação. Donde a única maneira de obter o conjunto vazio é ter um conceito relacional de ordem superior ou igual a $ordem(t)$.□

Os exemplos usam a habitual notação gráfica $\boxed{c_1} \to (\ r\ ) \to \boxed{c_2}$ em vez da notação matemática $\langle r, c_1, c_2 \rangle$.

Exemplo 15. O tipo de relação `CLASSE` faz a passagem do nível objecto para o meta-nível ao relacionar um indivíduo com o seu tipo. A relação

$$\boxed{\text{GATO: \texttt{\#Garfield}}} \to (\text{ CLASSE }) \to \boxed{\text{ESPÉCIE: \texttt{\#gato}}}$$

é de aridade dois, porque $aridade(\texttt{CLASSE})$ é igual ao número de argumentos (ver Exemplo 5 na página 35). Sendo estes não-relacionais, as funções auxiliares li e ls devolvem sempre 0, donde $x = y = 0$. Além disso, tem-se $ordem(\texttt{CLASSE}) = 1$ pelo que se obtém $[0, \ldots, 1 - 1] \cap [0, \ldots, 0]$ o que dá $[0]$ para a ordem da relação.

Exemplo 16. Uma relação que usa o tipo `INVERSA-DE` (Exemplo 6 na página 35) é $r = \boxed{\text{TRANSITIVA: \texttt{\#antepassado}}} \to (\text{ INVERSA-DE }) \to \boxed{\text{TRANSITIVA: \texttt{\#descendente}}}$. Tem-se $aridade(r) = 2$ e $ordem(r) = 1$ pois tanto `ANTEPASSADO` como `DESCENDENTE` são relações entre pessoas e portanto são de primeira ordem. Mesmo sendo `TRANSITIVA` de ordem or isso não altera a ordem dos conceitos que neste caso é determinada pelos referentes. Além disso, `INVERSA-DE` é de ordem $or + 1$ mas a relação é de ordem 1. Por aqui se vê que a ordem de uma relação é a ordem das entidades relacionadas e não a ordem dos tipos que são usados para classificar essas entidades ou a própria relação.

Exemplo 17. A relação $r = \boxed{\text{FELINO}} \to (\text{ ENTRE }) \to \boxed{\text{CARACTERÍSTICA}}$ é de ordem zero mas de aridade indefinida. Aplicando a definição obtém-se $aridade(r) = 0$ porque $aridade(\texttt{ENTRE}) = 3$ mas r tem dois argumentos. De modo análogo ao Exemplo 15, $ordem(r) = [0, \ldots, 1 - 1] \cap [0, \ldots, 0] = [0]$.

Exemplo 18. A relação $r = \boxed{\text{FELINO}} \to (\text{ AGNT }) \to \boxed{\text{ORDEM-PARCIAL}}$ é binária mas de ordem indefinida. Pelo Exemplo 4 na página 35 tem-se $ordem(\texttt{AGNT}) = 1$ e $aridade(\texttt{AGNT}) = 2$.

Como a relação tem de facto dois argumentos vem que $aridade(r) = 2$. De acordo com o Exemplo 2 na página 32, `ORDEM-PARCIAL` $\in T^{rc}_{or}$ pelo que $ordem(\boxed{\text{ORDEM-PARCIAL}}) = [1, \ldots, or]$. Então $ordem(r) = [0, \ldots, 1-1] \cap [max(\{0,1\}), \ldots, max(\{0, or\})] = [0] \cap [1, \ldots, or] = \emptyset$. Numa relação com argumentos relacionais, o tipo tem pois que ser pelo menos de segunda ordem.

Para finalizar defino uma ordem parcial sobre as relações, induzida pela hierarquia dos conceitos obtida anteriormente e pela hierarquia dos tipos de relação.

Definição 31. Uma relação r é uma *restrição* (ou *especialização*) da relação r' e inversamente r' é uma *generalização* de r, escrevendo-se $r \leq r'$, se e só se

- r e r' tiverem o mesmo número de argumentos;
- $tipo(r) \leq tipo(r')$;
- $arg(r, i) \leq arg(r', i)$ para cada i possível.

Proposição 32. *$\mathcal{R}$ é uma partição de reticulados.*

Demonstração. As relações de aridade i formam um reticulado pois são o produto cartesiano do reticulado $\mathcal{T}_{\mathcal{R}}$ (Hipótese 5 na página 34) com i vezes o reticulado $\mathcal{C}$ (Proposição 23 na página 44). Donde $\mathcal{R}$ é uma partição de reticulados, cada um deles contendo apenas as relações da mesma aridade. Por definição, a ordem parcial destes reticulados é a relação de especialização. $\square$

Por exemplo, o topo das relações binárias é $\boxed{\top_c\text{: *}} \rightarrow (\top_r) \rightarrow \boxed{\top_c\text{: *}}$ sendo a respectiva base $\boxed{\bot_c\text{: }\bar{*}} \rightarrow (\bot_r) \rightarrow \boxed{\bot_c\text{: }\bar{*}}$. Mas como os conceitos bem-formados não formam um reticulado, as relações bem-formadas (da mesma aridade) também não o fazem.

Exemplo 19. A maior especialização comum das relações bem-formadas

$$\boxed{\text{ACTO}} \rightarrow (\text{AGNT}) \rightarrow \boxed{\top_c\text{: \#João}} \quad \text{e} \quad \boxed{\text{RELAÇÃO}} \rightarrow (\text{INVERSA-DE}) \rightarrow \boxed{\text{RELAÇÃO}}$$

é a relação mal-formada $\boxed{\bot_c} \leftarrow (\bot_r) \leftarrow \boxed{\text{RELAÇÃO: \#João}}$. Há no entanto duas especializações comuns bem-formadas, mas incomparáveis (Exemplo 14 na página 44).

Capítulo 4

Grafos

Tendo os ingredientes básicos (conceitos e relações) pode-se passar aos grafos conceptuais. Como foi dito na introdução desta dissertação, os grafos vão ser classificados segundo o seu significado e a sua forma. Dos quatro níveis de significação, este capítulo apenas trata os dois primeiros: grafos sintacticamente correctos e grafos bem-formados. Quanto à forma, seguirei três critérios: a (in)existência de contextos, a (in)existência de dependências e a conectividade do grafo.

Os contextos são conceitos cujo referente não é uma entidade individual mas sim um conjunto de grafos. A sua utilidade principal consiste na representação da negação e da conjunção. Assim, os grafos com contextos conseguem representar todas as fórmulas da lógica proposicional (Capítulo 6 na página 91). A ligação de conceitos em contextos distintos é feita através dos *elos de correferência*, os quais, juntamente com os contextos, permitem aos grafos conceptuais representarem todas as fórmulas de primeira ordem. Basicamente, um elo de correferência é uma igualdade entre os referentes dos dois conceitos por ele ligados. No entanto, devido à introdução de tipos de ordem superior são necessárias outras igualdades (chamadas *dependências*) entre os vértices de um grafo, por exemplo entre os tipos de dois conceitos[1]. Ao contrário de um grafo "normal", um grafo com dependências representa todos os grafos que satisfazem as igualdades. As dependências também permitem ligar vértices de grafos distintos, juntando-os deste modo num só grafo, ao qual chamarei *grafo composto*. Removendo as dependências a um grafo composto obtém-se pois um grafo desconexo.

A estrutura deste capítulo é extremamente simples. As secções são dedicadas às várias formas de grafos. Dentro de cada secção é feita a distinção dos dois primeiros níveis de significação (grafos sintacticamente correctos e bem-formados). As duas primeiras secções tratam os grafos sem e com contextos. A terceira e quarta secções abordam os grafos com dependências, sendo o caso dos elos de correferência tratado isoladamente na terceira secção. A última secção é sobre os grafos compostos. Por sua vez, cada secção tem uma estrutura semelhante. Depois da definição geral vem o caso bem-formado, tratando-se por fim a hierarquia formada pelos grafos. A verificação da boa-formação de um grafo será computacionalmente eficiente.

[1] Sowa [1993] introduziu os operadores τ e ρ para equacionar o referente de um conceito com o tipo de um conceito ou relação. Os operadores foram formalmente definidos em [Wermelinger e Lopes, 1994] mas a presente abordagem não só é uma extensão como uma simplificação desse trabalho.

4.1 Grafos Rasos

No caso mais simples, um grafo conceptual é apenas constituído por conceitos e relações ligados entre si por arcos numerados, o que permite distinguir os vários argumentos de uma relação.

Hipótese 3.1.2 (Sowa). Um *grafo conceptual* é um grafo finito, conexo e bipartido.

- Os dois géneros de nós do grafo bipartido são *conceitos* e *relações*.
- Toda a relação tem um ou mais *arcos*, cada um ligado a um conceito.
- Se uma relação tem n arcos é chamada de n-ária, e os seus arcos são etiquetados com os números naturais $1, 2, \ldots, n$. O termo *unária* é sinónimo de 1-ária, *binária* de 2-ária e *ternária* de 3-ária.
- Um único conceito pode formar um grafo conceptual, mas cada arco de uma relação tem que estar ligado a algum conceito.

Esta hipótese introduz três noções: grafo, conceito e relação. Visto estar a seguir uma ordem de apresentação completamente diferente da de [Sowa, 1984], e por estar a seguir uma metodologia em que "para cada noção, a sua definição", já introduzi anteriormente as noções de conceito e de relação. Assim, segue-se a definição de grafo conceptual, que se mantém equivalente à de Sowa, embora esteja formulada de maneira diferente. Por oposição aos grafos com contextos (ver a secção seguinte), designo os grafos "normais" de rasos.

Hipótese 33. Um *grafo conceptual raso* $\langle V_C, V_R, E, etiqueta\rangle$ é um grafo finito, bipartido e conexo tal que V_C é o conjunto não vazio de *vértices conceptuais*, V_R é o conjunto de *vértices relacionais* e a função total *etiqueta*: $V_C \cup V_R \rightarrow \mathcal{C} \cup \mathcal{T}_\mathcal{R}$ associa a cada vértice conceptual um conceito e a cada vértice relacional um tipo de relação. Além disso, o conjunto dos *arcos* E é um subconjunto de $V_R \times \mathbb{N} \times V_C$ cujos elementos satisfazem a condição $\forall r \in V_R \;\; \forall c \in V_C \;\; \forall i > 1 \;\; \langle r, i, c\rangle \in E \Rightarrow \exists c' \in V_C \;\; \langle r, i-1, c'\rangle \in E$.

As duas definições de grafo conceptual assumem o mínimo possível. Basicamente, apenas se exige que os arcos liguem vértices de espécie diferente e que sejam numerados de forma consecutiva, isto é, que uma relação não salte sobre argumentos. Quanto às diferenças, a minha definição não diz explicitamente que um vértice relacional tem que ter um ou mais arcos e que esses arcos têm que estar ligados a conceitos, porque isso é implicado pelo facto do grafo ser conexo e bipartido. Também não é necessário dizer explicitamente que um conceito por si só pode formar um grafo conceptual, pois esse é o caso em que V_R está vazio, mas V_C não. Por outro lado, dois ou mais vértices conceptuais sem vértices relacionais não constituem um grafo pois violam a condição da conectividade. Resumindo, se V_R estiver vazio só pode haver mesmo um único conceito.

A separação entre vértices conceptuais e conceitos é importante porque vértices distintos podem estar etiquetados com o mesmo conceito. No entanto, se o conceito for genérico (por exemplo, $\boxed{\text{PESSOA: } *}$), as duas ocorrências do mesmo conceito podem representar entidades diferentes. Devido a esta distinção entre vértices, por um lado, e conceitos e relações, por outro, convém ter as seguintes funções auxiliares que serão bastante usadas no resto da dissertação.

Definição 34. Seja $\langle V_C, V_R, E, etiqueta\rangle$ um grafo conceptual. A função total $tp\colon V_C \to \mathcal{T}_\mathcal{C}$ que devolve para cada vértice conceptual o seu tipo é a composição de *etiqueta* com *tipo*, ou seja, $tp(c) = tipo(etiqueta(c))$ para qualquer $c \in V_C$.

Definição 35. Seja $\langle V_C, V_R, E, etiqueta\rangle$ um grafo conceptual. A função total $rf\colon V_C \to \mathcal{M}$ que devolve para cada vértice conceptual o seu referente é a composição de *etiqueta* com *referente*, isto é, $rf(c) = referente(etiqueta(c))$ para qualquer $c \in V_C$.

Definição 36. Seja $\langle V_C, V_R, E, etiqueta\rangle$ um grafo conceptual. A função total $rl\colon V_R \to \mathcal{R}$ devolve para vértice relacional r a relação por ele formada:

$$rl(r) = \langle etiqueta(r), etiqueta(c_1), \ldots, etiqueta(c_n)\rangle \Leftrightarrow \forall i = 1, \ldots, n \;\; \langle r, i, c_i\rangle \in E$$

A noção de boa-formação para os grafos rasos é imediata: um grafo está bem-formado se os conceitos e relações representados pelos vértices também o estiverem. Em termos práticos, se $V_R \neq \emptyset$ basta verificar que as relações do grafo estão bem-formadas pois isso inclui automaticamente a verificação de todos os vértices conceptuais.

Definição 37. Um grafo conceptual raso $\langle V_C, V_R, E, etiqueta\rangle$ está *bem-formado* se $\forall c \in V_C \;\; etiqueta(c) \in \mathcal{C}_f$ e se $\forall r \in V_R \;\; rl(r) \in \mathcal{R}_f$.

Tal como aconteceu com os tipos, os marcadores, os conceitos e as relações, vamos querer comparar grafos e organizá-los em conjuntos parcialmente ordenados. Essa relação de comparação chama-se de novo especialização, ou generalização, dependendo do ponto de vista que se queira tomar. A hierarquia dos conceitos foi induzida pelas hierarquias de tipos e marcadores, e a hierarquia de relações foi obtida a partir das hierarquias dos conceitos e dos tipos de relação. Assim, a hierarquia de grafos vai ser definida à custa das hierarquias dos seus vértices. Um grafo g' será uma especialização de g se as etiquetas de g' forem uma especialização das de g (usando a Definição 22 na página 44 e a Hipótese 5 na página 34). Mas g' também tem que manter a topologia — os arcos — de g. Para tal usa-se uma noção bastante vulgar da teoria de grafos, a de projecção[2].

Definição 38. Sejam $g = \langle V_C, V_R, E, etiqueta\rangle$ e $g' = \langle V'_C, V'_R, E', etiqueta'\rangle$ dois grafos conceptuais. Uma *projecção* $\pi\colon g \to g'$ é uma função que associa a g um subgrafo de g' chamado *projecção de g em g'* tal que

- $\forall c \in V_C \;\; \pi(c) \in V'_C \wedge etiqueta'(\pi(c)) \leq etiqueta(c)$
- $\forall r \in V_R \;\; \pi(r) \in V'_R \wedge etiqueta'(\pi(r)) \leq etiqueta(r)$

[2]Em [Sowa, 1984] a projecção e a especialização de grafos são usadas apenas no âmbito das regras canónicas de formação (Definição 3.5.1 na página 86 e Teorema 3.5.4 na página 86). Penso no entanto que se tratam de duas noções genéricas, pelo que devem ser aplicadas aos grafos conceptuais em geral e não apenas a um seu subconjunto, o dos grafos canonicamente deriváveis.

- $\forall e = \langle r, i, c \rangle \in E \;\; \pi e = \langle \pi(r), i, \pi(c) \rangle \in E'$

Por outras palavras, existe uma projecção de g em g' quando g' pode ser obtido de g especializando os vértices e, eventualmente, acrescentando novos vértices e arcos. Por isso g' será chamado uma especialização de g. No caso mais simples, especializar um grafo g resume-se apenas a especializar os vértices existentes, sem acrescentar novos vértices ou arcos, pelo que se obtém um grafo g' isomórfico ao original g. A este género de especialização chamarei "instância".

Definição 39. Um grafo conceptual g' é uma *especialização* (ou *restrição*) de um grafo conceptual g e inversamente g é uma *generalização* de g', escrevendo-se $g' \leq g$, se existir uma projecção de g em g'.

Definição 40. Diz-se que g' é uma *instância* de g se houver uma projecção de g em g' que seja uma função bijectiva.

Exemplo 20. Considere os seguintes grafos:

$$g_1 = \boxed{\top_c} \rightarrow (\text{AGNT}) \rightarrow \boxed{\top_c} \qquad g_2 = \boxed{\top_c} \rightleftarrows (\text{AGNT})$$

$$g_3 = \boxed{\text{ACTO}} \rightarrow (\text{AGNT}) \rightarrow \boxed{\text{ANIMADO}} \qquad g_4 = \boxed{\text{COMER}} \rightarrow (\text{AGNT}) \rightarrow \boxed{\text{PESSOA: \#João}}$$

As especializações são $g_4 \leq g_3 \leq g_1$ e $g_2 \leq g_1$. Mais concretamente, g_4 é uma instância de g_3 que por sua vez é uma instância de g_1. No entanto, g_2 não é instância de g_1 porque a projecção de g_1 em g_2 não é injectiva, donde não é bijectiva.

Um caso particular de instância é a cópia: dois grafos iguais são instâncias um do outro. Note-se também que se os dois grafos a comparar se resumirem a um único conceito ou a uma única relação caímos no caso particular da especialização de conceitos e relações (Definição 22 na página 44 e Definição 31 na página 48).

4.2 Grafos Hierárquicos

A par dos referentes, uma das fontes de maior expressividade na TEC são os contextos, o que os torna problemáticos do ponto de vista formal. Existem várias concepções do que deve ser um contexto, pelo que um tratamento aprofundado do assunto está para além do âmbito deste texto[3]. Assim, limitar-me-ei a adoptar a definição de [Sowa, 1984], sem alterações, pois ela serve para o uso limitado que farei dos contextos. De facto, a teoria original encara-os apenas como um conjunto de grafos que representam proposições (verdadeiras ou falsas) que podem ser crenças de algum indivíduo ou descrever uma situação. Do ponto de vista lógico, os contextos traduzem-se meramente na conjunção dos grafos contidos (Hipótese 4.2.3 na página 93), o que os torna pouco expressivos comparando com outras abordagens.

[3] A maior parte do trabalho sobre contextos tem sido feito por John Esch [1993a; 1993b; 1994a; 1994b; 1994c] que influenciou parte das extensões propostas pelo próprio Sowa [1992; 1995]. Em relação a trabalho feito fora da comunidade das Estruturas Conceptuais, veja-se [Guha, 1991; McCarthy, 1993; Buvač e Mason, 1993].

Definição 4.2.1 (Sowa). Seja p um conceito de tipo `PROPOSIÇÃO` cujo referente é o conjunto $\{u_1, \ldots, u_n\}$ de grafos conceptuais: `[PROPOSIÇÃO:` $\{u_1, \ldots, u_n\}$`]`. Então cada grafo u_i em *referente*(p) diz-se asserido pela *proposição* p, e u_i diz-se ocorrer no *contexto* de p.

Definição 4.2.2 (Sowa). Sejam os grafos $\{u_1, \ldots, u_n\}$ asseridos pela proposição p com a relação (NEG) ligada a p: `(NEG)->[PROPOSIÇÃO:` $\{u_1, \ldots, u_n\}$`]`. Então o grafo de (NEG) ligada a p chama-se um *contexto negativo*. Um contexto negativo contendo um único grafo que também é um contexto negativo chama-se uma *dupla negação*.

Para simplificar a escrita, Sowa convencionou que se pode omitir o tipo `PROPOSIÇÃO` e a relação `NEG` pode ser abreviada como ¬.

Exemplo 21. A afirmação "João não é uma pessoa" é representada pelo contexto negativo (NEG)→[PROPOSIÇÃO: [PESSOA: `#João`]]. A proposição "Nenhuma pessoa come" é dada por (NEG)→[PROPOSIÇÃO: [COMER]→(AGNT)→[PESSOA]]. A conjunção "Ninguém come e João não é uma pessoa" corresponde ao contexto (positivo)

[PROPOSIÇÃO: ¬[[COMER]→(AGNT)→[PESSOA]] ¬[[PESSOA: `#João`]]]

Se cada uma das afirmações é falsa individualmente, a sua conjunção também o é: ¬[[COMER]→(AGNT)→[PESSOA] [PESSOA: `#João`]].

Exemplo 22. O uso principal que darei aos contextos será para a representação de regras "se-então", ou seja, implicações. Visto que segundo a lógica clássica "p implica q" é equivalente a "não se tem simultaneamente p e não q", o grafo ¬[g_1 ¬[g_2]] representa a implicação pretendida, assumindo que o grafo conceptual g_1 representa a proposição p e g_2 representa q. Nomeadamente pode-se definir

tipo `SE(x)` **é** ¬[PROPOSIÇÃO: *x]   **tipo** `ENTÃO(x)` **é** ¬[PROPOSIÇÃO: *x]

o que permite escrever implicações na sua forma mais geral do seguinte modo:

[SE: $g_1, g_2, \ldots, g_n$ [ENTÃO: $g'_1, g'_2, \ldots, g'_m$]]

No entanto podia-se trocar o tipo `SE` pelo tipo `ENTÃO` e vice-versa, pois semanticamente são equivalentes.

Exemplo 23. Os contextos também são usados para descrever situações, tanto estáticas (estados) como dinâmicas (eventos e processos). Um exemplo tirado de [Sowa, 1992] é

[SITUAÇÃO: [GATO]←(AGNT)←[CAÇAR]→(OBJ)→[RATO]]→(DUR)→[PERÍODO: @13 seg]

Para uma representação detalhada de discursos em língua natural, em particular de informação temporal, veja-se o trabalho de Moulin [1993; 1994].

A definição de boa-formação para os grafos com contextos é simples.

Definição 41. Um grafo conceptual diz-se *hierárquico* se pelo menos um dos vértices conceptuais for um contexto.

Definição 42. Um grafo conceptual hierárquico está *bem-formado* se cada grafo contido num contexto estiver bem-formado.

Quanto à projecção, ela deve obviamente manter o encaixe dos contextos. No entanto, não se obriga a que o contexto $\pi(c)$ tenha exactamente o mesmo número de grafos que o contexto c. Por um lado, dois (ou mais) grafos no contexto c podem ser projectados num único grafo do contexto $\pi(c)$, mas também é possível que $\pi(c)$ tenha novos grafos que não estejam relacionados com os do contexto c. Em particular c pode estar vazio, mas $\pi(c)$ não.

Definição 43. Sejam g e g' dois grafos conceptuais hierárquicos. Uma projecção $\pi : g \to g'$ tem que ser tal que cada grafo contido num contexto c de g é projectado num grafo do contexto $\pi(c)$ de g'.

Exemplo 24. O grafo ¬[[COMER]→(AGNT)→[PESSOA] [PESSOA: #João]] é uma generalização de ¬[[COMER]→(AGNT)→[PESSOA: #João]].

Ao contrário do que acontece no Exemplo 23 na página precedente, nesta dissertação os contextos apenas são usados para representar a conjunção e a negação de proposições, pelo que as definições originais de Sowa acima apresentadas são suficientes. Deste modo um grafo hierárquico poderá apenas ser da forma[4] [PROPOSIÇÃO: $\{g_1, \ldots, g_n\}$] ou (NEG)→[PROPOSIÇÃO: $\{g_1, \ldots, g_n\}$] em que os g_i são grafos rasos ou hierárquicos. Daqui resulta que um contexto pode conter um ou mais contextos que por sua vez podem incluir outros e assim por diante. A definição seguinte introduz alguma nomenclatura associada a esta noção de encaixe.

Definição 4.2.4 (Sowa). O *contexto exterior* é a colecção de todos os grafos conceptuais que não ocorrem no referente de qualquer conceito.

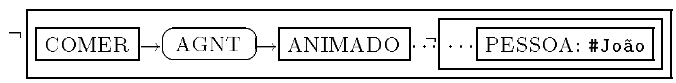
- Se um conceito ou grafo conceptual ocorre no contexto exterior, diz-se que está encaixado à *profundidade* 0.
- Se x é um contexto negativo que está encaixado à profundidade n, então qualquer grafo ou conceito que ocorre no contexto de x diz-se estar encaixado à profundidade $n + 1$.
- Para qualquer inteiro $n \geq 0$, um grafo ou conceito encaixado no nível $2n$ diz-se estar *com encaixe par*, e um grafo ou conceito encaixado no nível $2n + 1$ diz-se estar *com encaixe ímpar*.
- Se um contexto y ocorre num contexto x, então diz-se que x *domina* y. Se y dominar outro contexto z, então x também domina z. O contexto exterior domina todos os outros contextos.

[4] Na notação abreviada omitirei as chavetas e as vírgulas entre os g_i.

Note-se que o nível de encaixe só aumenta quando se transpõe um contexto negativo. Isto tem a ver com o facto desta definição ser usada nas regras de inferência de primeira ordem, em que é preciso distinguir os grafos que se encontram negados dos que não o estão, e neste último caso não interessa quantos contextos positivos se encontram à sua volta.

4.3 Grafos Correferentes

Muitas vezes é necessário indicar que dois vértices conceptuais distintos representam a mesma entidade. Isso acontece sobretudo quando os vértices estão em contextos diferentes, mas também podem estar no mesmo contexto. Em qualquer dos casos, o mecanismo dado pela TEC para representar a igualdade de referentes é o elo de correferência. Imaginemos por exemplo que queremos representar a frase "se alguém come, então esse alguém é João". O grafo conceptual correspondente é



O elo de correferência indicado pela linha pontilhada mostra que os conceitos $\boxed{\text{ANIMADO}}$ e $\boxed{\text{PESSOA: \#João}}$ representam a mesma entidade. Um elo de correferência pode ser visto como um arco não orientado que liga dois conceitos, os quais por sua vez podem estar ligados através de outros elos de correferência a outros conceitos e assim por diante, formando o todo um grafo ao qual se chama linha de identidade. Neste exemplo, a linha de identidade é $\boxed{\text{ANIMADO}}\cdots\boxed{\text{PESSOA: \#João}}$. Como o contexto de $\boxed{\text{PESSOA: \#João}}$ está encaixado no contexto de $\boxed{\text{ANIMADO}}$, diz-se que $\boxed{\text{ANIMADO}}$ domina $\boxed{\text{PESSOA: \#João}}$ e que os dois conceitos são correferentes. Vejamos a definição formal destas noções.

Hipótese 4.2.5 (Sowa). Uma *linha de identidade* é um grafo conexo e não orientado g cujos nós são conceitos e cujos arcos são pares de conceitos, chamados *elos de correferência*.

- Nenhum conceito pode pertencer a mais do que uma linha de identidade.
- Um conceito a em g diz-se *dominar* outro conceito b se existir um caminho $\langle a_1, a_2, \ldots, a_n \rangle$ em g com $a = a_1$, $b = a_n$ e para todo o i, ou a_i e a_{i+1} ocorrem ambos no mesmo contexto ou o contexto de a_i domina o contexto de a_{i+1}.
- Dois conceitos a e b são *correferentes* se a dominar b ou b dominar a.
- Um conceito a é *dominante* se a dominar todos os conceitos que dominarem a.

Um colecção de grafos conceptuais ligados por uma ou mais linhas de identidade é chamado um *grafo composto*. Um grafo conceptual sem linhas de identidade nem contextos encaixados é chamado um *grafo simples*.

Há dois pontos a salientar nesta definição. Em primeiro lugar, repare-se que segundo a formulação apresentada, um conceito é dominante se não houver nenhum conceito que o domine ou se os conceitos que o dominarem estiverem no mesmo contexto. Em segundo lugar, a classificação de grafos proposta não cobre a totalidade das possibilidades (como por exemplo os grafos com apenas um nível de contextos) e a distinção simples vs. composto baseia-se em dois critérios diferentes (conectividade e linhas de identidade). Irei pois adoptar a terminologia introduzida pela Hipótese 4.2.5 na página anterior à excepção da respeitante aos grafos, para os quais uso a seguinte definição.

Definição 44. Um grafo conceptual com uma ou mais linhas de identidade chama-se um grafo *correferente*.

Os exemplos de grafos correferentes que se seguem ilustram a nomenclatura usada, além de mostrarem as subtilezas da correferência e como o "senso comum" pode ser enganador.

Exemplo 25 ([Sowa, 1984]). Tendo a negação e a conjunção (o operador por defeito num contexto), consegue-se definir qualquer outra relação lógica, tal como `OU`.

$$\textbf{relação}\ \texttt{OU(x, y)}\ \textbf{é}\ \boxed{{}^*\text{x}}\cdots\neg\boxed{\cdots\neg\boxed{{}^*\text{x}}\ \neg\boxed{{}^*\text{y}}\cdots}\cdots\boxed{{}^*\text{y}}$$

Repare-se que os argumentos de `OU` são contextos, pois o tipo `PROPOSIÇÃO` foi omitido. Além disso, devido à maneira como a expansão relacional está definida [Sowa, 1984, p. 115], na definição de um tipo de conceito ou tipo de relação os argumentos têm que estar no contexto exterior, o que faz com que neste caso sejam necessários elos de correferência. Mais precisamente, o grafo correferente usado pela definição do tipo de relação `OU` tem dois elos de correferência, duas linhas de identidade e dois conceitos dominantes.

Exemplo 26 ([Esch, 1993b]). No grafo $\boxed{\text{GATO}}\cdots\boxed{\text{GATO}}$ temos uma linha de identidade (que é o próprio grafo) e dois conceitos correferentes, sendo ambos dominantes. Quanto a $\boxed{\text{GATO}}\cdots\boxed{\cdots\boxed{\text{GATO}}}$, apenas o conceito da esquerda é dominante, embora os dois conceitos estejam à profundidade 0 (Definição 4.2.4 na página 54). Continuando com o processo de encaixe obtém-se $\boxed{\boxed{\text{GATO}}\cdots}\cdots\boxed{\cdots\boxed{\text{GATO}}}$. Embore a linha de identidade continue a ser $\boxed{\text{GATO}}\cdots\boxed{\text{GATO}}$, os conceitos já não são correferentes mas ambos são de novo dominantes (porque não há nenhum conceito que os domine). A razão pela qual não são correferentes é que o elo de correferência primeiro sai de um contexto para depois entrar noutro.

Exemplo 27 ([Sowa, 1984]). O exemplo clássico da utilização de contextos negativos e linhas de identidade é a representação da frase "Existem pelo menos duas pessoas distintas", ou, dita de modo equivalente embora mais rebuscado, "Existem pessoas x e y mas não existe pessoa que seja idêntica a ambas". Esta segunda versão pode-se traduzir directamente para $\boxed{\text{PESSOA}}\cdots\neg\boxed{\cdots\boxed{\text{PESSOA}}\cdots}\cdots\boxed{\text{PESSOA}}$. Este grafo tem uma única linha de identidade mas dois elos de correferência e dois conceitos dominantes, os que estão fora do contexto negativo. No entanto esses dois conceitos não são correferentes, porque nenhum deles domina o outro, embora cada um deles seja correferente com o conceito do meio.

Como se vê, as noções de profundidade, dominância e correferência não são tão dependentes umas das outras como se possa julgar à primeira vista. Outra coisa que os exemplos mostram é que a ideia intuitiva de que "linhas de identidade expressam igualdade" [Sowa, 1984, p. 147] não pode ser levada à letra. Nem uma linha de identidade liga conceitos necessariamente idênticos (veja-se o exemplo com que começou esta secção), nem a correferência é transitiva (Exemplo 27 na página oposta) embora a igualdade o seja.

Veremos na Secção 6.1 na página 92, em que se mostra a semântica dos grafos conceptuais, que um elo de correferência indica a igualdade das entidades representadas pelos conceitos ligados. Assim, os referentes desses conceitos devem ser iguais, mas os tipos não precisam de o ser. É por isso que os conceitos ligados por uma linha de identidade não são necessariamente idênticos. Por outro lado, uma linha de identidade pode atravessar contextos com profundidades diferentes (isto é, contextos negativos). Por isso é que a correferência não é transitiva. Devido a estas características dos grafos hierárquicos, os elos de correferência não podem ser usados de qualquer maneira. A discussão que se segue, cujo objectivo é chegar à definição de grafo correferente bem-formado, tenta explicar melhor o significado dos elos de correferência e a sua utilização.

A definição de linha de identidade (e por conseguinte a de grafo correferente) não impede um elo de correferência entre dois conceitos no mesmo contexto. Do ponto de vista prático pode ser útil caso os dois conceitos pertençam a grafos diferentes e se desejar-se mantê-los separados. No entanto, do ponto de vista teórico, o significado desses dois grafos é equivalente ao de um único grafo, obtido por junção dos conceitos correferentes.

Exemplo 28. O grafo $\boxed{\text{GULOSO}} \leftarrow (\text{ATRIB}) \leftarrow \boxed{\text{GATO}} \cdots \boxed{\text{FELINO: }\texttt{\#Garfield}} \rightarrow (\text{AGNT}) \rightarrow \boxed{\text{COMER}}$ tem o mesmo significado que $\boxed{\text{GULOSO}} \leftarrow (\text{ATRIB}) \leftarrow \boxed{\text{GATO: }\texttt{\#Garfield}} \rightarrow (\text{AGNT}) \rightarrow \boxed{\text{COMER}}$.

Pode-se pois assumir que um elo de correferência denota o maior sub-conceito comum dos conceitos ligados. Este facto foi utilizado por Esch [1993b] para a definição de novos subtipos. Eis um exemplo seu.

Exemplo 29. O um quadrado pode ser definido como sendo um rectângulo (um quadrilátero cujos lados formam ângulos rectos) que é simultaneamente um losango (um quadrilátero cujos lados têm comprimento igual):

$$\textbf{tipo}\ \texttt{QUADRADO(x)}\ \textbf{é}\ \boxed{\text{RECTÂNGULO: *x}} \cdots \boxed{\text{LOSANGO: *x}}$$

Então quais são as correferências não permitidas num grafo bem-formado? Para a resposta voltemos ao Exemplo 26 na página oposta, em que a única linha de identidade sem significado era a do terceiro grafo $\boxed{\boxed{\text{GATO}}\cdots} \cdots \boxed{\cdots\boxed{\text{GATO}}}$. Se traduzirmos este grafo para uma fórmula em lógica de predicados, usando as hipóteses de Sowa apresentadas na Secção 6.1 na página 92, obtemos $\exists x\ Gato(x) \wedge \exists y\ Gato(y)$. Ou seja, embora o utilizador tenha querido forçar a identidade dos dois conceitos, isso é impedido pelo facto dos contextos em que se encontram estarem completamente separados. Por isso, para um elo de correferência fazer sentido tem que ligar um conceito dominante a um dominado. Usando a terminologia da Hipótese 4.2.5 na página 55, tem que ligar conceitos correferentes.

Definição 45. Um grafo correferente está *bem-formado* se cada elo de correferência ligar dois conceitos correferentes.

Outra maneira de entender a correferência é através da comparação com as variáveis de linguagens de programação tradicionais e regras de escopo associadas. Duas variáveis com o mesmo nome que ocorrem em dois blocos de programa diferentes designam a mesma entidade se (1) o bloco de uma variável contém o bloco da outra ou (2) se houver uma declaração para essa variável num bloco que contenha esses dois blocos (e se não houver redeclarações pelo meio). De modo análogo, dois conceitos c_1 e c_2 em contextos diferentes cx_1 e cx_2 representam a mesma entidade (1) se houver um elo de correferência entre c_1 e c_2 e um dos contextos contiver o outro ou (2) se houver um conceito c_3 num contexto cx_3 tal que c_1 e c_2 estão ligados a c_3 por elos de correferência e cx_3 contém tanto cx_1 como cx_2. Em qualquer dos casos, os elos de correferência são sempre entre conceitos dominantes e dominados.

A função de projecção introduzida na Secção 4.1 na página 50 adapta-se facilmente aos grafos correferentes. Assim como no caso normal a projecção de dois vértices deve também incluir o arco entre eles, no caso de dois vértices conceptuais ligados por um elo de correferência, a projecção deve manter esse elo.

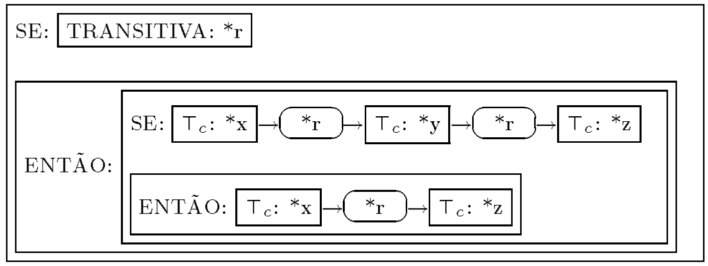
Definição 46. Sejam g e g' dois grafos conceptuais correferentes. Uma projecção $\pi : g \to g'$ tem que ser tal que para cada elo de correferência entre conceitos c_1 e c_2 de g existe um elo de correferência entre os conceitos $\pi(c_1)$ e $\pi(c_2)$ de g'.

4.4 Grafos com Dependências

Na secção anterior disse que os conceitos ligados por um elo de correferência representam a mesma entidade e como tal devem ter o mesmo referente. Ou seja, um elo de correferência não é mais que uma equação (isto é, uma igualdade) entre dois referentes. No entanto, esta espécie de *dependências*[5] entre vértices conceptuais de um grafo não é suficiente para tirar partido dos tipos relacionais e de ordem superior, os quais também podem ser usados como marcadores (Secção 2.3 na página 36). Para estes casos serão precisas igualdades entre o tipo de um conceito e o referente de outro, entre um tipo de uma relação e o referente de um conceito, etc. Os exemplos que se seguem motivam a necessidade de novas dependências para além dos elos de correferência. Para facilitar o trabalho de composição do texto, as dependências serão visualizadas através de variáveis.

Exemplo 30. Uma das aplicações mais úteis das dependências é a especificação de esquemas de axiomas para tipos de ordem superior. Por exemplo, para o tipo de conceito relacional `TRANSITIVA` (Exemplo 2 na página 32) tem-se o seguinte grafo, em que a variável '*r' representa uma dependência "mista" entre o referente de um vértice conceptual e o tipo de um vértice relacional.

[5] Para simplicar este trabalho, as dependências entre vértices serão sempre igualdades. O termo "correferências" seria pois mais apropriado, mas preferi usá-lo apenas no sentido da TEC original.



Exemplo 31. O tipo de relação `CLASSE`, introduzido no Exemplo 5 na página 35, pode ser definido da forma

relação `CLASSE(x,y)` **é** [*y: *x] [$\top_c$: *x]→(ELO)→[$\top_c$: *y]

o que permite escrever grafos tais como [PESSOA: `#João`]→(CLASSE)→[REINO: `#animal`] em que no referente do segundo conceito aparece um supertipo do tipo do primeiro conceito. Isto é possível em parte devido à dependência estabelecida pela variável '*y'. Mas sem o elo de correferência representado pela variável '*x' a definição seria

relação `CLASSE(x,y)` **é** [*y: *x]→(ELO)→[$\top_c$: *y]

e então os dois campos teriam que ser exactamente iguais:

[PESSOA: `#João`]→(CLASSE)→[TIPO: `#pessoa`]

Exemplo 32. O EXPRESS é porventura a linguagem de programação mais usada em *Computer Integrated Manufacturing*. Uma tradução parcial de EXPRESS para Estruturas Conceptuais foi apresentada em [Wermelinger e Bejan, 1993]. Entre as funcionalidades que ficaram por tratar encontram-se as funções genéricas. Por exemplo, o cabeçalho `FUNCTION f(x: GENERIC:t) : GENERIC:t;` indica que a função f aceita um argumento de qualquer tipo (de primeira ordem) t e devolve um valor do mesmo tipo t. Tendo dois conceitos [$\top_c$: *x] e [$\top_c$: *f] para representar respectivamente o argumento e o resultado da função, como dizer que têm o mesmo tipo mas não necessariamente o mesmo referente (pois isso seria dizer que o valor de saída é igual ao valor de entrada)? Uma hipótese será utilizando a relação `CLASSE` que, como se viu no Exemplo 31, envolve um tipo de ordem superior e duas dependências:

[$\top_c$: *x]→(CLASSE)→[TIPO]←(CLASSE)←[$\top_c$: *f]

No entanto a tradução mais simples e directa é [*t: *x] [*t: *f] que requer apenas uma dependência (entre os tipos de dois conceitos) e nenhum tipo de ordem superior.

Exemplo 33. Um grafo com dependências também pode ser encarado como uma pergunta à base de conhecimentos, consistindo a resposta de todos os grafos que satisfazem as igualdades. Por exemplo, a pergunta "Qual a espécie de Garfield?" traduz-se em [ESPÉCIE: *x] [*x: `#Garfield`]. A resposta será [ESPÉCIE: `#gato`] [GATO: `#Garfield`]. Se ao invés da espécie pretendermos saber a família, basta substituir `ESPÉCIE` por `FAMÍLIA` na pergunta, sendo a resposta [FAMÍLIA: `#felino`] [FELINO: `#Garfield`] (ver Exemplo 13 na página 43).

Dada a necessidade de estender os elos de correferência a dependências mais flexíveis entre os vértices de um grafo, o resto desta secção discute em pormenor as dependências necessárias e qual o seu significado, além de adaptar as noções de boa-formação e de projecção a grafos com dependências.

Começando pelas possíveis equações entre os vértices de um grafo, verifica-se facilmente que há seis combinações no total. Ei-las, em que c e c' são vértices conceptuais, e r e r' são vértices relacionais[6]:

1. $tp(c) = tp(c')$	4. $rf(c) = rf(c')$
2. $tp(c) = rf(c')$	5. $rf(c) = etiqueta(r)$
3. $tp(c) = etiqueta(r)$	6. $etiqueta(r) = etiqueta(r')$

A equação 3 é obviamente impossível de satisfazer visto os tipos de conceito e os tipos de relação serem disjuntos (Hipótese 10 na página 35). A equação 4 corresponde a um elo de correferência[7]. Por outro lado, será que as equações da forma 1 e 6 são dispensáveis? Mais especificamente, pode-se reescrever $tp(c) = tp(c')$ como a conjunção das dependências $tp(c) = rf(c'')$ e $rf(c'') = tp(c')$ e pode-se encarar $etiqueta(r) = etiqueta(r')$ como uma abreviatura de $etiqueta(r) = rf(c)$ e $rf(c) = etiqueta(r')$? A resposta é não porque as dependências 1 ou 6 admitem os tipos universais e absurdos como solução, o que não acontece com as formas reescritas pois nem $\top_c$, $\bot_c$, $\top_r$ ou $\bot_r$ podem ocorrer no referente de um conceito.

É pois ponto assente que, à excepção das equações da forma 3, todas as outras são necessárias. O Exemplo 30 na página 58 usa as formas 4, 5 e 6, enquanto a forma 1 é usada no Exemplo 32 na página anterior e a forma 6 aparece no Exemplo 33 na página precedente e no Exemplo 31 na página anterior. Um grafo com dependências poderá ser então encarado como um grafo normal ao qual se acrescenta um conjunto não vazio de equações entre referentes e tipos.

Hipótese 47. Um grafo conceptual com *dependências* é um grafo $\langle V_C, V_R, E, etiqueta\rangle$ com um conjunto não vazio de equações (as dependências), cada uma delas sendo de uma das seguintes formas, em que $c, c' \in V_C$ e $r, r' \in V_R$:

$$tp(c) = tp(c') \qquad rf(c) = rf(c') \qquad tp(c) = rf(c')$$
$$rf(c) = etiqueta(r) \qquad etiqueta(r) = etiqueta(r')$$

Mas qual a diferença entre grafos com dependências e sem dependências? O significado dado neste trabalho às dependências advém do Exemplo 33 na página precedente: um grafo com dependências representa todas as suas soluções, isto é, todas as instâncias (Definição 38 na página 51) do grafo que satisfaçam as equações. Como um elo de correferência é uma dependência particular, é importante que esta interpretação inclua a dos grafos correferentes. E de facto a noção de linha de identidade como denotando a entidade comum das representadas pelos conceitos ligados adapta-se bem à ideia intuitiva de dependências.

[6]Relembro que $tp(c) = tipo(etiqueta(c))$ e $rf(c) = referente(etiqueta(c))$ (51).

[7]Na TEC original, um elo de correferência corresponde na prática a um par de equações 1 e 4 porque, como veremos na Secção 5.1 na página 68, para cada indivíduo tem que existir um único tipo mínimo ao qual esse indivíduo pertence. Assim, a igualdade de referentes implica a igualdade de tipos.

Visto as dependências serem uma generalização da correferência, os grafos com dependências bem-formados devem incluir os grafos correferentes bem-formados. Por essa razão as dependências bem-formadas terão que obedecer ao encaixe dos contextos. Por outro lado, o significado dado no parágrafo anterior a um grafo com dependências implica que ele só estará bem-formado se pelo menos uma das suas soluções o estiver. No entanto, esta definição de boa-formação depende dos tipos usados. Por exemplo, imaginemos que os tipos de primeira ordem `VERMELHO` e `AZUL` não têm nenhum subtipo em comum além de $\perp_c$. Então um grafo que contenha os conceitos $c_1 = \boxed{\text{VERMELHO}}$, $c_2 = \boxed{\text{AZUL}}$ e $c_3 = \boxed{\text{COR}}$ juntamente com a equação $tipo(c_1) = tipo(c_2) = referente(c_3)$ não tem solução possível, porque a única maneira de os tipos de c_1 e c_2 serem iguais é torná-los $\perp_c$ que no entanto não pode ser o referente de c_3. Como não existe solução (bem-formada), o grafo com dependências também não estará bem-formado. Isto não me parece muito natural, porque o problema reside apenas no facto de não existir um subtipo comum (por exemplo, `VIOLETA`). De resto está tudo bem: as dependências ligam dois tipos de primeira ordem com um referente também ele de primeira ordem (porque `COR` é de segunda ordem).

Resumindo, tal como aconteceu com os conceitos, as relações e portanto com os grafos rasos, a boa-formação dos grafos com dependências apenas depende da ordem, do género e da aridade dos tipos envolvidos. No entanto, ao contrário do que aconteceu com os grafos rasos, a determinação da boa-formação não pode ser feita localmente (isto é, verificando os vértices um por um) pois as dependências podem abarcar a globalidade do grafo. Deste modo a definição formal de grafos com dependências bem-formados baseia-se no seguinte método. A cada vértice conceptual associam-se três variáveis (uma para a ordem do tipo, outra para a ordem do referente, e a terceira para o género do conceito) e a cada vértice relacional duas variáveis (uma para a ordem e outra para a aridade). A partir das dependências e do próprio grafo extrai-se uma série de condições, sobretudo (in)equações, que têm que ser resolvidas. As soluções indicam as ordens, géneros e aridades possíveis dos vértices das instâncias do grafo que satisfazem as dependências. Deste modo, um grafo está bem-formado apenas se o sistema de condições associado for solúvel.

Vejamos com algum detalhe as condições extraídas do grafo. Seja c um vértice conceptual e sejam $c_{\mathtt{t}}$, $c_{\mathtt{r}}$ e $c_{\mathtt{g}}$ as variáveis associadas ao tipo, referente e género de c, respectivamente. De modo semelhante, se r é um vértice relacional, as variáveis $r_{\mathtt{t}}$ e $r_{\mathtt{a}}$ representam a ordem e a aridade do tipo da relação. À excepção de $c_{\mathtt{g}}$ as variáveis são inteiras porque representam ordens ou aridades. Para facilitar a formalização convencionei que os géneros `rel` e `nonrel` são codificados pelos números 1 e 2, respectivamente. Assim, a variável $c_{\mathtt{g}}$ também será inteira. Por outro lado, a ordem dos tipos e marcador absurdos será por convenção -1, e a dos tipos universais e do marcador genérico será -2, deste modo não havendo confusão com as ordens dos restantes tipos e marcadores, as quais nunca são negativas. As condições são obtidas sempre pelo mesmo princípio: como é que uma dependência entre dois vértices ou a ordem, o género ou aridade de um vértice restringe as especializações possíveis desse(s) vértice(s)? Vejamos alguns exemplos.

Se o tipo de um conceito c for $\perp_c$, então não pode ser mais especializado, pelo que qualquer instância do grafo terá que satisfazer $c_{\mathtt{t}} = -1$. Por outro lado, se o tipo for $\top_c$, então está tudo em aberto para as possíveis instâncias. Neste caso não se lhe

$c \in V_C$	$c_\mathtt{t} = -1 \;\vee\; c_\mathtt{t} = -2 \;\vee\; c_\mathtt{r} = -1 \;\vee\; c_\mathtt{r} = -2 \;\vee$ $c_\mathtt{g} = 1 \vert c_\mathtt{r} \leq c_\mathtt{t} \;\vee\; c_\mathtt{g} = 2 \vert c_\mathtt{t} = c_\mathtt{r} + 1$
$tp(c) = \bot_c$	$c_\mathtt{t} = -1$
$tp(c) \in T_i^{nc}$	$c_\mathtt{g} = 2 \;\wedge\; c_\mathtt{t} = i \;\vee\; c_\mathtt{t} = -1$
$tp(c) \in T_i^{rc}$	$c_\mathtt{g} = 1 \;\wedge\; 1 \leq c_\mathtt{t} \leq i \;\vee\; c_\mathtt{t} = -1$
$rf(c) = \bar{*}$	$c_\mathtt{r} = -1$
$rf(c) \in T_i^{nc}$	$c_\mathtt{g} = 2 \;\wedge\; c_\mathtt{r} = i \;\vee\; c_\mathtt{r} = -1$
$rf(c) \in T_i^{r}$	$c_\mathtt{g} = 1 \;\wedge\; 1 \leq c_\mathtt{r} \leq i \;\vee\; c_\mathtt{r} = -1$
$tp(c) = tp(c')$	$c_\mathtt{t} = c'_\mathtt{t} \;\wedge\; 1 \leq c_\mathtt{t} \vert c_\mathtt{g} = c'_\mathtt{g}$
$rf(c) = rf(c')$	$c_\mathtt{r} = c'_\mathtt{r} \;\wedge\; 0 \leq c_\mathtt{r} \vert c_\mathtt{g} = c'_\mathtt{g}$
$tp(c) = rf(c')$	$c_\mathtt{g} = c'_\mathtt{g} = 2 \;\wedge\; 1 \leq c_\mathtt{t} = c'_\mathtt{r} \leq on$
$etiqueta(r) = rf(c)$	$c_\mathtt{g} = 1 \;\wedge\; 1 \leq r_\mathtt{t} = c'_\mathtt{r} \leq or + 1$
$etiqueta(r) = etiqueta(r')$	$r_\mathtt{t} = r'_\mathtt{t}$
$etiqueta(r) = \bot_r$	$r_\mathtt{t} = -1$
$etiqueta(r) \in T_i^r$	$1 \leq r_\mathtt{t} \leq i \;\vee\; r_\mathtt{t} = -1$
$etiqueta(r) \in T_{\langle i \rangle}^r$	$r_\mathtt{a} = i \;\vee\; r_\mathtt{t} = -1$
$\langle r, i, c \rangle \in E$	$i \leq r_\mathtt{a} \;\wedge$ $(r_\mathtt{t} = -1 \;\vee\; c_\mathtt{g} = 2 \;\vee\; c_\mathtt{t} < r_\mathtt{t} \;\wedge\; c_\mathtt{r} < r_\mathtt{t})$

Tabela 4.1: Condições associadas a um grafo com dependências.

associa qualquer condição. Se o tipo for não-relacional de ordem i, então há duas possibilidades segundo a hierarquia de tipos de conceito (Hipótese 1 na página 30): ou o tipo na instância será $\bot_c$ ou então também será não-relacional e da mesma ordem. A condição obtida é pois $c_\mathtt{t} = -1 \vee c_\mathtt{g} = 2 \wedge c_\mathtt{t} = i$, assumindo que a conjunção tem prioridade sobre a disjunção. Passando a uma dependência, no caso de ela ser da forma $tp(c) = rf(c')$ uma das condições a satisfazer é obviamente $c_\mathtt{t} = c'_\mathtt{r}$. Mas quando é que um tipo de conceito pode também ser usado como marcador? Consultando a Hipótese 11 na página 37 chega-se à conclusão que $\mathcal{T_C} \cap \mathcal{M} = \bigcup_{i=1}^{on} T_i^{nc}$. Daqui vem pois $c_\mathtt{g} = c'_\mathtt{g} = 2 \wedge 1 \leq c_\mathtt{t} \leq on$. Para finalizar esta série de exemplos, vejamos as condições associadas a um conceito. Basicamente, elas têm que especificar a boa-formação do conceito. Segundo a Proposição 20 na página 43, se o tipo for absurdo ou universal ou se o marcador for genérico ou absurdo, então o conceito estará de certeza bem-formado. Tem-se pois a condição $c_\mathtt{t} = -1 \vee c_\mathtt{t} = -2 \vee c_\mathtt{r} = -1 \vee c_\mathtt{r} = -2$. Adicionalmente, se o conceito for de género não-relacional, então o tipo tem que ser de ordem imediatamente superior à do referente mas se o género for `rel` apenas se exige que a ordem do tipo não seja inferior à do referente. Para esta espécie de condições "se-então" utilizarei guardas. Uma condição guardada é da forma $G|C$ em que a guarda G é uma (in)equação e a condição C não tem outras guardas nem tem conjunções ou disjunções. O exemplo em causa traduz-se em $c_\mathtt{g} = 1 | c_\mathtt{r} \leq c_\mathtt{t} \;\vee\; c_\mathtt{g} = 2 | c_\mathtt{t} = c_\mathtt{r} + 1$. A condição $G|C$ é equivalente à condição C se G for verdadeira, caso contrário é como se $G|C$ não existisse.

Definição 48. Seja $g = \langle V_C, V_R, E, etiqueta \rangle$ um grafo conceptual com dependências. Seja $\mathcal{V} = \{c_\mathtt{t} | c \in V_C\} \cup \{c_\mathtt{r} | c \in V_C\} \cup \{c_\mathtt{g} | c \in V_C\} \cup \{r_\mathtt{t} | r \in V_R\} \cup \{r_\mathtt{a} | r \in V_R\}$ um conjunto de variáveis. O grafo g dá origem a um conjunto de condições segundo a

Tabela 4.1 na página oposta. Uma condição *guardada* $G|C$ é equivalente a C se G for verdadeira, caso contrário $G|C$ é verdadeira se ocorrer numa conjunção e falsa se ocorrer numa disjunção. Então g está *bem-formado* se para cada dependência entre vértices x e y o contexto de x dominar o de y ou vice-versa e se existir uma função total $f : \mathcal{V} \to \mathbb{Z}$ que satisfaça o sistema de condições.

Antes de passar aos exemplos de aplicação desta definição gostaria de chamar a atenção para dois aspectos. Por um lado, o sistema de condições associado a um grafo é um conjunto de fórmulas proposicionais, sendo cada proposição uma (in)equação entre inteiros. Como tal, este sistema de restrições pode ser resolvido em tempo polinomial por um algoritmo híbrido de manutenção de consistência das condições [van Hentenryck *et al.*, 1991; Mackworth e Freuder, 1985]. Em segundo lugar, visto a boa-formação de um grafo com dependências ser independente dos tipos concretos usados, interessando apenas as suas ordens, géneros, e aridades, obtém-se a seguinte definição equivalente, mais "declarativa" e intuitiva.

Proposição 49. *Um grafo conceptual com dependências g está* bem-formado *se cada dependência entre vértices x e y for tal que o contexto de x domine o de y ou vice-versa e se for possível acrescentar tipos às hierarquias $\mathcal{T_C}$ e $\mathcal{T_R}$ usadas por g de maneira a obter uma instância bem-formada de g que satisfaça todas as dependências.*

Nos exemplos que se seguem irei numerar os vértices conceptuais sequencialmente da esquerda para a direita, fazendo o mesmo para os vértices relacionais. Então $c^i_{\mathtt{t}}$, $c^i_{\mathtt{r}}$ e $c^i_{\mathtt{g}}$ serão respectivamente as variáveis associadas ao tipo, referente e género do i-ésimo conceito, sendo $r^i_{\mathtt{t}}$ e $r^i_{\mathtt{a}}$ as variáveis associadas ao tipo e aridade do i-ésimo vértice relacional. Para facilitar a leitura, utilizarei de novo os identificadores `rel` e `nonrel` em vez dos inteiros 1 e 2.

Exemplo 34. Consideremos o grafo constituído apenas pelo conceito $\boxed{\top_c}$ juntamente com a dependência $tp(c1) = rf(c1)$ (isto é o conceito $\boxed{\text{*x: *x}}$ que está na origem do paradoxo de Russell no Capítulo 2 na página 27). Então tem-se as seguintes condições:

- $c^1_{\mathtt{t}} = -1 \;\vee\; c^1_{\mathtt{t}} = -2 \;\vee\; c^1_{\mathtt{r}} = -1 \;\vee\; c^1_{\mathtt{r}} = -2 \;\vee$
  $c^1_{\mathtt{g}} = \mathtt{rel}|c^1_{\mathtt{r}} \leq c^1_{\mathtt{t}} \;\vee\; c^1_{\mathtt{g}} = \mathtt{nonrel}|c^1_{\mathtt{t}} = c^1_{\mathtt{r}} + 1$
- $c^1_{\mathtt{g}} = \mathtt{nonrel} \;\wedge\; 1 \leq c^1_{\mathtt{t}} = c^1_{\mathtt{r}} \leq on$

Como $c^1_{\mathtt{g}} = \mathtt{nonrel}$ e $c^1_{\mathtt{t}} = c^1_{\mathtt{r}}$, a primeira condição reduz-se a $c^1_{\mathtt{t}} = -1 \vee c^1_{\mathtt{t}} = -2 \vee c^1_{\mathtt{t}} = c^1_{\mathtt{t}} + 1$. Visto $1 \leq c^1_{\mathtt{t}}$ obtém-se finalmente a contradição $c^1_{\mathtt{t}} = c^1_{\mathtt{t}} + 1$ pelo que o grafo não está bem-formado, como se esperava.

Exemplo 35. O grafo que define a relação `CLASSE` (Exemplo 31 na página 59) pode ser visto como o grafo $\boxed{\top_c}$ $\boxed{\top_c} \to (\text{ELO}) \to \boxed{\top_c}$ com as dependências $tp(c1) = tp(c2)$, $rf(c1) = \mathtt{rf}(c2)$, $tp(c1) = rf(c3)$. Aplicando as regras da tabela obtêm-se as condições:

- $c^1_{\mathtt{t}} = -1 \;\vee\; c^1_{\mathtt{t}} = -2 \;\vee\; c^1_{\mathtt{r}} = -1 \;\vee\; c^1_{\mathtt{r}} = -2 \;\vee$
  $c^1_{\mathtt{g}} = \mathtt{rel}|c^1_{\mathtt{r}} \leq c^1_{\mathtt{t}} \;\vee\; c^1_{\mathtt{g}} = \mathtt{nonrel}|c^1_{\mathtt{t}} = c^1_{\mathtt{r}} + 1$

- $c^2_{\mathtt{t}} = -1 \;\vee\; c^2_{\mathtt{t}} = -2 \;\vee\; c^2_{\mathtt{r}} = -1 \;\vee\; c^2_{\mathtt{r}} = -2 \;\vee$
  $c^2_{\mathtt{g}} = \mathtt{rel} | c^2_{\mathtt{r}} \leq c^2_{\mathtt{t}} \;\vee\; c^2_{\mathtt{g}} = \mathtt{nonrel} | c^2_{\mathtt{t}} = c^2_{\mathtt{r}} + 1$
- $c^3_{\mathtt{t}} = -1 \;\vee\; c^3_{\mathtt{t}} = -2 \;\vee\; c^3_{\mathtt{r}} = -1 \;\vee\; c^3_{\mathtt{r}} = -2 \;\vee$
  $c^3_{\mathtt{g}} = \mathtt{rel} | c^3_{\mathtt{r}} \leq c^3_{\mathtt{t}} \;\vee\; c^3_{\mathtt{g}} = \mathtt{nonrel} | c^3_{\mathtt{t}} = c^3_{\mathtt{r}} + 1$
- $c^1_{\mathtt{t}} = c^2_{\mathtt{t}} \;\wedge\; 1 \leq c^1_{\mathtt{t}} | c^1_{\mathtt{g}} = c^2_{\mathtt{g}}$
- $c^1_{\mathtt{r}} = c^2_{\mathtt{r}} \;\wedge\; 0 \leq c^1_{\mathtt{r}} | c^1_{\mathtt{g}} = c^2_{\mathtt{g}}$
- $c^1_{\mathtt{g}} = c^3_{\mathtt{g}} = \mathtt{nonrel} \;\wedge\; 1 \leq c^1_{\mathtt{t}} = c^3_{\mathtt{r}} \leq on$
- $1 \leq r^1_{\mathtt{t}} \leq or + 1 \;\vee\; r^1_{\mathtt{t}} = -1$
- $r^1_{\mathtt{a}} = 2 \;\vee\; r^1_{\mathtt{t}} = -1$
- $1 \leq r^1_{\mathtt{a}}$
- $2 \leq r^1_{\mathtt{a}}$
- $r^1_{\mathtt{t}} = -1 \;\vee\; c^1_{\mathtt{g}} = \mathtt{nonrel} \;\vee\; c^1_{\mathtt{t}} < r^1_{\mathtt{t}} \;\wedge\; c^1_{\mathtt{r}} < r^1_{\mathtt{t}}$
- $r^1_{\mathtt{t}} = -1 \;\vee\; c^2_{\mathtt{g}} = \mathtt{nonrel} \;\vee\; c^2_{\mathtt{t}} < r^1_{\mathtt{t}} \;\wedge\; c^2_{\mathtt{r}} < r^1_{\mathtt{t}}$

Este sistema de condições pode ser simplificado da seguinte maneira:

- $c^1_{\mathtt{r}} = -1 \;\vee\; c^1_{\mathtt{r}} = -2 \;\vee\; c^1_{\mathtt{t}} = c^1_{\mathtt{r}} + 1$
- $c^3_{\mathtt{t}} = -1 \;\vee\; c^3_{\mathtt{t}} = -2 \;\vee\; c^3_{\mathtt{t}} = c^3_{\mathtt{r}} + 1$
- $c^1_{\mathtt{t}} = c^2_{\mathtt{t}} \;\wedge\; c^1_{\mathtt{g}} = c^2_{\mathtt{g}}$
- $c^1_{\mathtt{r}} = c^2_{\mathtt{r}} \;\wedge\; 1 \leq c^1_{\mathtt{r}} | c^1_{\mathtt{g}} = c^2_{\mathtt{g}}$
- $c^1_{\mathtt{g}} = c^3_{\mathtt{g}} = \mathtt{nonrel} \;\wedge\; 1 \leq c^1_{\mathtt{t}} = c^3_{\mathtt{r}} \leq on$
- $1 \leq r^1_{\mathtt{t}} \leq or + 1 \;\vee\; r^1_{\mathtt{t}} = -1$
- $r^1_{\mathtt{a}} = 2 \;\vee\; r^1_{\mathtt{t}} = -1$
- $2 \leq r^1_{\mathtt{a}}$

Estas condições são satisfazíveis: para a instância tirada do mesmo exemplo $\boxed{\text{ANIMAL=*y: \texttt{\#João=*x}}}$ $\boxed{\text{PESSOA: \texttt{\#João=*x}}} \rightarrow (\text{ELO}) \rightarrow \boxed{\text{REINO: \texttt{\#animal=*y}}}$ tem-se $c^1_{\mathtt{t}} = c^2_{\mathtt{t}} = c^3_{\mathtt{r}} = 1, c^1_{\mathtt{r}} = c^2_{\mathtt{r}} = 0, c^3_{\mathtt{t}} = 2, c^1_{\mathtt{g}} = c^2_{\mathtt{g}} = c^3_{\mathtt{g}} = \mathtt{nonrel}, r^1_{\mathtt{a}} = 2, r^1_{\mathtt{t}} = or + 1$. Note-se que este grafo, além de bem-formado, é verdadeiro pois uma das classes de `#João` é `ANIMAL`.

Exemplo 36. A condição $1 \leq c_{\mathtt{t}} | c_{\mathtt{g}} = c'_{\mathtt{g}}$ gerada pela dependência $tp(c) = tp(c')$ serve para grafos como $\boxed{\top_c\text{: *x}}$ $\boxed{\top_c\text{=*x: *}}$ $\boxed{\top_c\text{=*x: \texttt{\#agnt}}}$ em que gostaríamos de poder deduzir que $rf(c_3) = \bar{*}$ nos grafos bem-formados que satisfazem as dependências. Vejamos a totalidade das condições.

1. $c^1_{\mathtt{t}} = -1 \;\vee\; c^1_{\mathtt{t}} = -2 \;\vee\; c^1_{\mathtt{r}} = -1 \;\vee\; c^1_{\mathtt{r}} = -2 \;\vee$
   $c^1_{\mathtt{g}} = \mathtt{rel} | c^1_{\mathtt{r}} \leq c^1_{\mathtt{t}} \;\vee\; c^1_{\mathtt{g}} = \mathtt{nonrel} | c^1_{\mathtt{t}} = c^1_{\mathtt{r}} + 1$

2. $c^2_{\mathtt{t}} = -1 \;\vee\; c^2_{\mathtt{t}} = -2 \;\vee\; c^2_{\mathtt{r}} = -1 \;\vee\; c^2_{\mathtt{r}} = -2 \;\vee$
   $c^2_{\mathtt{g}} = \mathtt{rel} | c^2_{\mathtt{r}} \leq c^2_{\mathtt{t}} \;\vee\; c^2_{\mathtt{g}} = \mathtt{nonrel} | c^2_{\mathtt{t}} = c^2_{\mathtt{r}} + 1$
3. $c^3_{\mathtt{t}} = -1 \;\vee\; c^3_{\mathtt{t}} = -2 \;\vee\; c^3_{\mathtt{r}} = -1 \;\vee\; c^3_{\mathtt{r}} = -2 \;\vee$
   $c^3_{\mathtt{g}} = \mathtt{rel} | c^3_{\mathtt{r}} \leq c^3_{\mathtt{t}} \;\vee\; c^3_{\mathtt{g}} = \mathtt{nonrel} | c^3_{\mathtt{t}} = c^3_{\mathtt{r}} + 1$
4. $c^3_{\mathtt{g}} = \mathtt{rel} \;\wedge\; 1 \leq c^3_{\mathtt{r}} \leq 1 \;\vee\; c^3_{\mathtt{r}} = -1$
5. $c^2_{\mathtt{t}} = c^3_{\mathtt{t}} \;\wedge\; 1 \leq c^2_{\mathtt{t}} | c^2_{\mathtt{g}} = c^3_{\mathtt{g}}$
6. $c^1_{\mathtt{g}} = c^2_{\mathtt{g}} = \mathtt{nonrel} \;\wedge\; 1 \leq c^1_{\mathtt{t}} = c^2_{\mathtt{t}} \leq on$

Das condições 5 e 6 vem que $c^3_{\mathtt{g}} = \mathtt{nonrel}$ pelo que $c^3_{\mathtt{r}} = -1$ para que a condição 4 seja verdadeira. Isto torna a terceira condição verdadeira, pelo que pode ser eliminada. Aplicando outras simplificações obtém-se o resultado final

1. $c^1_{\mathtt{t}} = -1 \;\vee\; c^1_{\mathtt{t}} = -2 \;\vee\; c^1_{\mathtt{t}} = c^1_{\mathtt{r}} + 1$
2. $c^2_{\mathtt{t}} = c^2_{\mathtt{r}} + 1$
4. $c^3_{\mathtt{r}} = -1$
5. $c^2_{\mathtt{t}} = c^3_{\mathtt{t}} \;\wedge\; c^2_{\mathtt{g}} = c^3_{\mathtt{g}}$
6. $c^1_{\mathtt{g}} = c^2_{\mathtt{g}} = \mathtt{nonrel} \;\wedge\; 1 \leq c^1_{\mathtt{r}} = c^2_{\mathtt{t}} \leq on$

Termino apresentando a condição adicional para a projecção no caso de ser aplicada a grafos com dependências.

Definição 50. Sejam g e g' dois grafos conceptuais com dependências. Uma projecção $\pi : g \to g'$ tem que ser tal que para cada dependência entre dois vértices v_1 e v_2 de g existe a mesma dependência entre $\pi(v_1)$ e $\pi(v_2)$ em g'.

Exemplo 37. O grafo $\boxed{\top_c} \to (\text{ATRIB}) \to \boxed{\top_c} \to (\text{CLASSE}) \to \boxed{\top_c}$ que define `CARAC` (Exemplo 5 na página 35) e o grafo $\boxed{\text{*y: *x}}$ $\boxed{\top_c\text{: *x}} \to (\text{ELO}) \to \boxed{\top_c\text{: *y}}$ que define `CLASSE` (Exemplo 31 na página 59) são incomparáveis, o que prova que nem `CARAC` < `CLASSE` nem `CLASSE` < `CARAC`. O primeiro não pode ser projectado no segundo porque nem `ATRIB` nem `CLASSE` podem ser projectados no seu supertipo `ELO`. Mas como o primeiro grafo não contém nenhuma dependência, também não pode conter uma projecção do segundo.

4.5 Grafos Simples e Compostos

Como já se viu em vários exemplos anteriores, as dependências (o que inclui as linhas de identidade) podem ligar conceitos que se encontram em grafos distintos.

Definição 51. Um grafo conceptual *composto* é constituído por dois ou mais grafos conceptuais desconexos entre si mas ligados através de dependências. Um grafo conceptual sem nenhuma ligação a qualquer outro diz-se *simples*.

Fica assim completa a classificação dos grafos conceptuais consoante a sua forma. Existem pois três critérios: conectividade, (in)existência de contextos e (in)existência de dependências. A maior parte dos grafos com dependências apenas usa elos de correferência. Estes casos especiais, mas suficientemente importantes, serão designados por "grafos correferentes". Exceptuando o facto de os grafos com dependências incluirem os grafos compostos, não existe relação entre os critérios. Independentemente da forma, os grafos conceptuais podem ser divididos em mal-formados e bem-formados. Eis alguns exemplos das várias combinações possíveis. O grafo conceptual $\boxed{\text{*y: *x}}$ $\boxed{\top_c\text{: *x}} \rightarrow (\text{ELO}) \rightarrow \boxed{\top_c\text{: *y}}$ (Exemplo 31 na página 59) é um grafo raso com dependências composto bem-formado, enquanto $\boxed{\boxed{\text{GATO}}\cdots}\cdots\boxed{\cdots\boxed{\text{GATO}}}$ (o terceiro grafo do Exemplo 26 na página 56) é hierárquico, correferente, composto e mal-formado. O grafo conceptual do Exemplo 30 na página 58 é hierárquico, com dependências, simples e bem-formado.

No caso dos grafos hierárquicos e com dependências, convém por vezes examinar as suas propriedades ignorandos os contextos e as dependências. Nesse sentido, a seguinte definição será útil.

Definição 52. O grafo *subjacente* de um grafo conceptual hierárquico e/ou com dependências é o grafo que se obtém ignorando os grafos encaixados e/ou as dependências.

Note-se que um grafo subjacente não é necessariamente um grafo conceptual. Em particular, o grafo subjacente de um grafo conceptual composto é um grafo desconexo constituído por vários grafos conceptuais simples.

Embora não tenha sido dito explicitamente na Definição 42 na página 54, na Definição 45 na página 58, ou na Definição 48 na página 62, assumiu-se que para um grafo hierárquico ou com dependências estar bem-formado, o seu grafo subjacente também tem que estar. A razão prende-se com os seguintes exemplos.

Exemplo 38. As condições associadas, segundo a Tabela 4.1 na página 62, ao grafo com dependências $\boxed{\text{JOÃO}}\cdots\boxed{\text{PESSOA}}$ permitem uma solução que representa a instância bem-formada $\boxed{\perp_c}\cdots\boxed{\perp_c}$. No entanto, o conceito $\boxed{\text{JOÃO}}$ não está bem-formado porque usa um elemento de T_0^{nc} como tipo de conceito. Donde o grafo no seu todo também não está bem-formado.

Exemplo 39. O conceito $\boxed{\text{PENSAR: \#agnt}}$ tem género indefinido. Logo, o grafo hierárquico

$$\boxed{\text{GATO: \#Garfield}} \leftarrow (\text{AGNT}) \leftarrow \boxed{\text{PENSAR: \#agnt}} \rightarrow (\text{OBJ}) \rightarrow \boxed{\boxed{\text{CÃO: \#Odie}} \rightarrow (\text{ATRIB}) \rightarrow \boxed{\text{ESTÚPIDO}}}$$

está mal-formado apesar de o grafo encaixado estar bem-formado.

Capítulo 5

Canonicidade

Os grafos bem-formados vistos no capítulo anterior são grafos que estão apenas bem tipados, o que não significa necessariamente que façam sentido para uma certa aplicação. Para que isso aconteça, é preciso ter uma ontologia e um mecanismo para derivar a partir dela grafos ontologicamente correctos. Na Teoria das Estruturas Conceptuais a ontologia designa-se por *cânone*, o mecanismo de derivação é dado pelas *regras canónicas de formação* e os grafos ontologicamente correctos chamam-se *grafos canónicos*. A especificação de um cânone divide-se em três partes: hierarquias de tipos e marcadores, *relação de conformidade* e *base canónica*. A relação de conformidade indica para cada marcador todos os tipos com os quais ele é compatível. Deste modo, a relação de conformidade especifica os conceitos ontologicamente correctos, os quais são usados na base canónica, que não é mais do que o conjunto inicial de grafos canónicos a partir dos quais se podem obter todos os restantes por aplicação das regras canónicas de formação.

A primeira secção deste capítulo é dedicada à relação de conformidade. A maneira mais intuitiva de a entender consiste em encarar a conformidade como uma explicitação da denotação: um marcador m é conforme com um tipo t se m <u>for</u> uma instância de t, ou seja, se m pertencer à denotação de t. Como se verá, esta interpretação da noção de conformidade acarreta vários problemas, pelo que seguirei um outro método: m é conforme com t se m <u>puder ser</u> instância de t. Esta pequena diferença entre necessidade e possibilidade leva a uma nova definição formal da relação de conformidade, mais geral que a original.

A segunda secção trata da base canónica, o meio ao dispôr do engenheiro do conhecimento para especificar *restrições de selecção*. De facto, os grafos que formam a base canónica, chamados *grafos base*, limitam-se a indicar quais conceitos podem ser ligados através de quais relações. Para facilitar o trabalho, e tal como acontece na TEC original, os grafos base não têm contextos nem dependências. A secção termina com a definição formal de cânone.

A terceira secção aborda as regras canónicas de formação. As regras originais de [Sowa, 1984] apenas especializam os grafos a que são aplicadas mas em [Sowa, 1995] também os podem generalizar pois são encaradas como um passo intermédio na especificação das regras de inferência. Tendo em conta os níveis de significação adoptados neste trabalho, os grafos verdadeiros (tratados no próximo capítulo) devem ser canónicos, pelo que sigo a mesma abordagem. No entanto, a definição formal das regras sofre bastantes alterações de pormenor. Além disso, permite manipular contextos e dependências.

A última secção define os grafos canónicos de duas formas. A primeira baseia-se na noção de *derivação canónica*: um grafo canónico é um grafo derivável a partir da base canónica usando as regras canónicas de formação. A segunda baseia-se em projecções e permite reconhecer um grafo canónico sem construir explicitamente a derivação. O algoritmo obtido é quase idêntico ao dado por [Mugnier e Chein, 1993b] para a teoria original, e tem a mesma complexidade, que na maioria dos casos é polinomial. O novo tratamento da canonicidade não tem pois um impacto computacional negativo.

5.1 Relação de Conformidade

Os conceitos da TEC original apenas são constituídos por tipos não-relacionais de primeira ordem e marcadores de "ordem zero". Assim, embora todos eles estejam bem-formados, isto é, não são de ordem ou género indefinido (Definição 19 na página 42), há conceitos que fazem menos sentido que outros. Por exemplo, se `#João` designa uma pessoa, o conceito $\boxed{\text{GATO: \#João}}$ não é aceitável. Para o engenheiro do conhecimento poder especificar as combinações tipo/marcador que fazem sentido para uma certa aplicação, Sowa definiu a relação de conformidade, designada pelo símbolo ::. Assim, dado `PESSOA` :: `#João`, um sistema de Estruturas Conceptuais poderá verificar que $\boxed{\text{GATO: \#João}}$ não está conforme e como tal não pode ser usado em grafos canónicos.

Hipótese 3.3.3 (Sowa). A *relação de conformidade ::* relaciona tipos com marcadores individuais: se $t :: i$ for verdadeiro, diz-se que i é *conforme* com o tipo t. A relação de conformidade obedece às seguintes condições:

- O referente dum conceito tem que ser conforme com o seu tipo. Isto é, se c é um conceito, então $tipo(c) :: referente(c)$.
- Se um marcador individual é conforme com o tipo s, então também é conforme com todos os supertipos de s. Ou seja, se $s \leq t$ e $s :: i$, então $t :: i$.
- Se um marcador individual é conforme com os tipos s e t, é conforme com o seu subtipo comum máximo. Isto é, se $s :: i$ e $t :: i$, então $(s \wedge t) :: i$.
- Todo o marcador individual é conforme com o tipo universal $\top$; nenhum marcador individual é conforme com o tipo absurdo $\bot$. Ou seja, para todo o i em I verifica-se $\top :: i$ mas não $\bot :: i$.
- O marcador genérico $*$ é conforme com todos os tipos. Isto é, para todos os tipos t, $t :: *$.

Antes de prosseguir com a análise desta hipótese, convém saber o que representa a relação de conformidade. De acordo com o exemplo anterior, a interpretação mais imediata, e que foi seguida em [Wermelinger e Lopes, 1994], consiste em considerar que um marcador m é conforme com um certo tipo t se a entidade representada por

m pertencer ao conjunto de entidades representado por t. A esse conjunto chama-se *denotação* e escreve-se δt. Como já foi dito no Capítulo 3 na página 39, visto os tipos estarem organizados numa hierarquia, a denotação tem certas propriedades, a saber[1]: se t é subtipo de t' então δt é um subconjunto de $\delta t'$; a denotação do tipo absurdo é o conjunto vazio; a denotação do tipo universal é o universo todo.

A vantagem de encarar a relação de conformidade como a simples explicitação da denotação dos tipos tem a vantagem prática de a Hipótese 3.3.3 na página anterior poder ser reformulada de maneira muito mais compacta e elegante ganhando em generalidade: $t :: m \Leftrightarrow m = * \vee m \in \delta t$. Repare-se como, partindo desta definição, se podem deduzir todas as condições pretendidas:

- se $t < t'$ então $\delta t \subseteq \delta t'$ donde $t :: m \Rightarrow t' :: m$;
- como $\delta \top_c$ é todo o universo, $\top_c :: m$ para qualquer m;
- como $\delta \bot_c = \emptyset$, $\bot_c :: m$ é falso para todo o $m \neq *$;
- para qualquer t, é imediato que $t :: *$.

Intuitivamente, vê-se que se "`#João` é conforme com `PESSOA`" equivale a "João é uma pessoa", então João também é um mamífero, um animal, um objecto físico, etc., razão pela qual `#João` tem que ser conforme com todos os supertipos de `PESSOA`.

No entanto, esta abordagem levanta vários problemas, a começar pela terceira restrição da Hipótese 3.3.3 na página oposta: um marcador conforme com dois tipos tem que ser conforme com o seu subtipo comum máximo, o qual é único visto os tipos estarem organizados num reticulado. Essa condição implica que nenhum marcador individual pode ser conforme com dois ou mais tipos incompatíveis, como foi observado em [Chein e Mugnier, 1992]. De facto se $t :: m$ e $t' :: m$ mas $t \wedge t' = \bot_c$, obter-se-ia $\bot_c :: m$ o que não é permitido. Poder-se-á perguntar: onde é que está o problema? Se dois tipos são incompatíveis claro que não podem ter nenhum indivíduo em comum. Ninguém duvida desta afirmação. A questão reside no entanto em se saber quando é que dois tipos são incompatíveis. Mais precisamente, será que se $t \wedge t' = \bot_c$ na hierarquia de tipos, então t e t' são incompatíveis? A resposta de Sowa é mista: sim e não. O sim obtém-se desta hipótese, o não vem do Teorema 3.2.6 [Sowa, 1984, p. 83], no qual se diz que se $t \wedge t' = t''$ então o conjunto $\delta t''$ das entidades denotadas por t'' está contido em $\delta t \cap \delta t'$. Ou seja, t'' denota um subconjunto das entidades comuns a t e t'. No caso a tratar, t e t' têm um indivíduo m em comum, e $t'' = \bot_c$. Portanto deve-se ter $\delta \bot_c \subseteq \delta t \cap \delta t'$, isto é $\delta \bot_c \subseteq \{m\}$, o que é perfeitamente possível, porque o tipo absurdo representa o conjunto vazio: $\delta \bot_c = \emptyset \subseteq \{m\}$. Resumindo, segundo a Hipótese 3.3.3 na página anterior é impossível dois tipos, cujo subtipo comum máximo é o tipo absurdo, terem pelo menos um indivíduo em comum, enquanto o Teorema 3.2.6 permite essa situação.

O fulcro da questão é pois: qual a interpretação da hierarquia de tipos? Segundo [Beierle *et al.*, 1990] há pelo menos duas hipóteses. Na chamada interpretação segundo o reticulado[2] (dos conjuntos denotados pelos tipos), o subtipo comum máximo de

[1] Para mais detalhes ver a Secção 6.2 na página 103.

[2] No original, *lattice-theoretic interpretation*.

dois tipos denota exactamente a intersecção das denotações desses dois tipos, isto é, $\delta(t \wedge t') = \delta t \cap \delta t'$. Já na interpretação segundo a ordem parcial[3] de conjuntos, o subtipo comum máximo pode representar apenas um subconjunto da intersecção: $\delta(t \wedge t') \subseteq \delta t \cap \delta t'$. Ou seja, segundo esta interpretação, a hierarquia de tipos é apenas uma representação parcial do reticulado dos subconjuntos do domínio da denotação. Voltando à pergunta, e pelo que ficou atrás exposto, a resposta dada pela TEC tal como se encontra formulada em [Sowa, 1984] é a seguinte: a hierarquia de tipos deve ser interpretada com base no reticulado para as entidades nomeáveis (isto é, para as quais existem marcadores) e com base na ordem parcial para todas as outras. Como os seguintes exemplos mostram, a interpretação segundo o reticulado é pouco natural, donde não muito aconselhável em IA, e leva à explosão de tipos, isto é, à existência de muitos tipos sem relevância ontológica.

Exemplo 40 (adaptado de [Beierle et al.*, 1990]).* Assuma que os edifícios históricos são o subtipo comum máximo das igrejas e edifícios antigos:

$$\begin{tikzcd} \texttt{IGREJA} \arrow[dr, dash] & & \texttt{EDIFÍCIO-ANTIGO} \arrow[dl, dash] \\ & \texttt{EDIFÍCIO-HISTÓRICO} & \end{tikzcd}$$

Isto significa que qualquer edifício histórico é uma igreja antiga mas assumir o inverso (isto é, todas as igrejas antigas são edifícios históricos) não é conveniente. Por exemplo, é possível ter $\delta(\texttt{IGREJA}) = \{$S. Pedro, Sta. Maria, S. Paulo$\}$, $\delta(\texttt{EDIFÍCIO-ANTIGO}) = \{$S. Pedro, Sta. Maria, Câmara Municipal$\}$, mas apenas $\delta(\texttt{EDIFÍCIO-HISTÓRICO}) = \{$S. Pedro$\}$. Por outras palavras, embora `IGREJA` :: `#sta-maria` e `EDIFÍCIO-ANTIGO` :: `#sta-maria`, *não* se tem `EDIFÍCIO-HISTÓRICO` :: `#sta-maria`.

Exemplo 41. Imagine uma base de conhecimentos sobre pessoas, contendo profissões (`PROFESSOR`, `ADVOGADO`, etc.), relações de parentesco (`MÃE`, `IRMÃO`, etc.) e estados civis (`CASADO`, `DIVORCIADO`, etc.). Cada um destes tipos denota um conjunto de pessoas, nomeadamente todas aquelas que têm em comum a característica dada pelo tipo, e cada indivíduo pode ser instância de vários tipos. Por exemplo, imaginemos que João é professor e casado, e Maria é mãe divorciada exercendo advocacia. A interpretação segundo o reticulado força a criação de um tipo para cada combinação possível tal como `PROFESSOR-CASADO` e `ADVOGADO-DIVORCIADO-MÃE`. E para tratar casos de duplo emprego seriam necessários tipos como `PROFESSOR-ADVOGADO-CASADO-IRMÃO`. Esta explosão de tipos não acontece com a interpretação segundo a ordem parcial que permite ter `PROFESSOR` :: `#João` e `CASADO` :: `#João` sem a adição de um novo tipo: $\texttt{PROFESSOR} \wedge \texttt{CASADO} = \bot_c$.

Mesmo adoptando a interpretação segundo a ordem parcial, se a relação de conformidade apenas for a indicação dos tipos de que cada marcador é instância, então ela torna-se desnecessária, pois o mesmo efeito pode ser obtido através de axiomas na base de conhecimentos: a relação $t :: m$ equivale ao grafo $\boxed{t : m}$. Uma indicação clara desta equivalência é o facto de as regras de inferência de primeira ordem (Hipótese 4.3.5 na página 111) conseguirem derivar o mesmo que é exigido para a relação de conformidade.

[3] No original, *order-theoretic interpretation*.

Como $\boxed{t : m}$ se encontra no contexto exterior, pode ser generalizado, assim obtendo-se $\boxed{t' : m}$ (donde $t' :: m$) para qualquer supertipo t' de t, incluindo o caso $t' = \top_c$. Para obter $t :: *$, qualquer que seja o tipo t, basta ter o grafo $\boxed{\bot_c : *}$; o processo de generalização faz o resto. Finalmente, visto as regras de inferência de primeira ordem serem consistentes, o grafo $\boxed{\bot_c : m}$, com $m \neq *$, jamais pode ser obtido.

Neste texto darei pois outra interpretação à relação de conformidade, consistente com a vontade de separar claramente os vários níveis de significação, e talvez mais próxima das intenções originais de Sowa[4]. A ideia é que um marcador m é conforme com um tipo t se quisermos que um dos grafos $\boxed{t\colon m}$ e $\neg\boxed{\boxed{t\colon m}}$ seja verdadeiro. Ou seja, conformidade não vai ser o mesmo que verdade: um grafo pode obedecer à relação de conformidade e ser falso. Inversamente, se os conceitos de um grafo não forem conformes, não faz sentido dizer que esse grafo é verdadeiro ou falso. Por exemplo, tendo apenas `GATO` :: `#Garfield` a veracidade dos grafos $\boxed{\text{PESSOA: \#Garfield}}$ e $\neg\boxed{\boxed{\text{PESSOA: \#Garfield}}}$ fica indefinida. Esta nova noção de conformidade leva quase automaticamente à nova definição formal.

A primeira observação a fazer é a seguinte. Sendo :: um conjunto de pares ordenados tipo/marcador, terá que ser $:: \subseteq \mathcal{C}$. Mas como não faz sentido dizer que um marcador é conforme com um certo tipo se forem de ordens ou géneros diferentes, chega-se à conclusão que a conformidade só se aplica a conceitos bem-formados. A segunda observação que nos conduz à nova definição tem a ver com os tipos universal e absurdo e com os marcadores genérico e absurdo. Repare-se que Sowa permite (1) $\bot_c :: *$ mas não (2) $\bot_c :: m$ sendo m um marcador individual. Qual a razão? Vistas bem as coisas, interpretando :: como explicitação de δ, as duas afirmações são falsas porque correspondem a (1) $\exists x \in \delta\bot_c$ e (2) $m \in \delta\bot_c$, sendo $\delta\bot_c = \emptyset$. Não há pois nenhuma razão intuitiva para permitir uma delas mas não a outra. Das duas uma: ou se proibe ou se permite $\bot_c :: m$ para qualquer m. Segundo a minha interpretação da relação ::, seguirei o segundo caminho, porque o grafo $\neg\boxed{\boxed{\bot_c\colon m}}$ é verdadeiro. Também é preciso ter em conta o marcador absurdo, o qual não existia na teoria original. Visto ele representar o "anti-indivíduo", que não é instância de nenhum tipo, o grafo $\neg\boxed{\boxed{t\colon \bar{*}}}$ é verdadeiro para qualquer tipo t, pelo que permitirei ao marcador absurdo ser conforme com qualquer tipo.

A última observação a fazer diz respeito à segunda restrição da Hipótese 3.3.3 na página 68: se um marcador é conforme com um certo tipo t também o é com todos os supertipos de t. Na nova abordagem, se $t :: m$ então ou $\boxed{t\colon m}$ ou $\neg\boxed{\boxed{t\colon m}}$ é verdadeiro. Mas se o grafo $\boxed{t\colon m}$ for verdadeiro, o grafo $\boxed{t'\colon m}$, em que t' é supertipo de t, também o é, pelo que $t' :: m$. Por outro lado, se t' é um subtipo de t e se $\neg\boxed{\boxed{t\colon m}}$ é verdadeiro, $\neg\boxed{\boxed{t'\colon m}}$ também se verifica, donde $t' :: m$. Resumindo, se m for conforme com um dado tipo t, também é conforme com todos os supertipos e subtipos de t. No entanto, esta regra implica a separação da relação de conformidade em duas partes: a que é especificada pelo engenheiro do conhecimento e a que pode ser fornecida pelo sistema. A razão principal é que a noção de conformidade se reduz à de boa-formação se as duas partes ficarem juntas. De facto, como qualquer marcador m é conforme com o tipo absurdo (e com o universal), por aplicação da regra anterior ele será conforme

[4]O seu novo livro não inclui a noção de conformidade.

com qualquer outro tipo. Ou seja, sem a divisão de :: em duas subrelações, todos os conceitos bem-formados seriam conformes, e assim a própria noção de conformidade deixaria de ser necessária.

Formalizando toda esta discussão, a hipótese com a qual começou esta secção transforma-se na Hipótese 53, onde se impede o engenheiro do conhecimento de especificar os pares tipo/marcador que podem ser automaticamente adicionados por um sistema de Estruturas Conceptuais (Hipótese 54).

Hipótese 53. A *relação de conformidade ::* é um subconjunto de $\mathcal{C}_f - (\{\top_c, \bot_c\} \times \mathcal{M}) - (\mathcal{T_C} \times \{*, \bar{*}\})$.

Hipótese 54. O *fecho* da relação de conformidade ::, designado por $::^*$, obtém-se do seguinte modo:

- $\forall m \in \mathcal{M} \;\; \top_c ::^* m \wedge \bot_c ::^* m$
- $\forall t \in \mathcal{T_C} \;\; t ::^* * \wedge t ::^* \bar{*}$
- $\forall t, t' \in \mathcal{T_C} \;\; \forall m \in \mathcal{M} \;\; t :: m \wedge (t \leq t' \vee t' \leq t) \Rightarrow t' ::^* m$

O relaxamento das condições impostas à relação de conformidade, para além de me parecer teoricamente mais elegante, tem uma vantagem prática: permitir indicar onde a junção de dois conceitos falhou. Imaginemos que pretendemos juntar o conceito $\boxed{\text{PESSOA: \#João}}$ com $\boxed{\text{GATO: \#João}}$. Na TEC original as regras canónicas de formação não o permitem pois obter-se-ia um conceito não conforme: $\boxed{\bot_c\text{: \#João}}$. Se esta junção fizesse parte de um processo de reconhecimento de frases em Português, e se o utilizador quisesse saber quais as interpretações que foram rejeitadas e porquê, o sistema teria que providenciar código especial para produzir uma explicação para estes casos. Com a nova formulação, basta ao sistema ter uma variável booleana indicando se grafos com tipos e/ou marcadores absurdos devem ser considerados. No processo normal de análise de um texto, a variável estará a 'falso', mas se o utilizador quiser conhecer todas as interpretações possíveis, a variável passa a 'verdade' e o mesmo código de processamento de grafos permitirá obter o conceito acima dado, indicando assim que a interpretação falha no tipo do conceito.

A definição que se segue será útil na próxima secção e permite eliminar a distinção entre a relação de conformidade e o fecho da mesma. Assim, no resto do texto o termo "relação de conformidade" designará $::^*$, excepto nas definições formais e salvo indicação em contrário.

Definição 55. Dada uma relação de conformidade ::, um *conceito conforme* é um conceito que obedece a $::^*$ e um *grafo conceptual conforme* é um grafo conceptual bem-formado em que todos os conceitos são conformes.

5.2 Cânone

Como se viu na secção anterior, a relação de conformidade apenas permite ao engenheiro do conhecimento impedir o utilizador da base de conhecimentos de usar conceitos que, embora bem-formados, não fazem sentido para a aplicação em causa. Por exemplo, para considerar o conceito $\boxed{\text{VERMELHO: \#João}}$ não conforme basta ter o cuidado de não indicar t :: `#João` para algum subtipo ou supertipo de `VERMELHO`. Visto a relação de conformidade só actuar no nível dos conceitos, é preciso um mecanismo para impedir a má utilização de relações. Por essa razão, a teoria tem a noção de *grafo canónico*. Dito de modo informal, os grafos canónicos são os grafos que fazem sentido segundo uma certa visão do modelo que se pretende representar em Estruturas Conceptuais. Sowa apresenta[5] três processos para a obtenção de grafos canónicos:

Percepção Representação em Estruturas Conceptuais de observações directas da realidade.

Formação Aplicação de regras especiais a grafos canónicos obtidos previamente.

Introspecção Estipulação de grafos arbitrários como sendo canónicos, baseada em intuição e diversas formas de raciocínio.

Visto as capacidades de percepção e introspecção dos actuais sistemas computacionais serem bastante limitadas ou mesmo inexistentes, cabe ao engenheiro do conhecimento definir o conjunto mínimo de grafos que descreve com suficiente detalhe parte da ontologia necessária para as bases de conhecimentos em vista. Esse conjunto chama-se *base canónica* e a ontologia designa-se por *cânone*. O processo de formação é simulado através da definição de um conjunto de regras concretas, as chamadas *regras canónicas de formação*. Assim, a base canónica não é mais do que o conjunto inicial de grafos conceptuais aos quais se podem aplicar as regras. Os grafos obtidos deste modo designam-se por *grafos canónicos*.

Definição 3.4.4 (Sowa). Seja A um conjunto qualquer de grafos conceptuais. Diz-se que um grafo w é *canonicamente derivável* de A se uma das seguintes condições for verdadeira:

- w é um membro de A.
- w pode ser derivado aplicando uma das regras canónicas de formação aos grafos u e v que são eles próprios canonicamente deriváveis de A.

Hipótese 3.4.5 (Sowa). O *cânone* contém a informação necessária para derivar um conjunto de grafos conceptuais. Tem quatro componentes:

- uma hierarquia de tipos T,
- um conjunto de marcadores individuais I,
- uma relação de conformidade :: que relaciona etiquetas em T com marcadores em I,
- um conjunto finito de grafos conceptuais B, chamado *base canónica*, com todos os tipos em T e todos os referentes ou $*$ ou marcadores em I.

[5] Trata-se da Hipótese 3.4.1 que decidi não apresentar textualmente por envolver termos que não são usados neste documento.

Os *grafos canónicos* são o fecho de B sob as regras canónicas de formação. Se um novo grafo, que não pode ser canonicamente derivado de B, é considerado canónico, então tem que ser adicionado a B.

No entanto, ao contrário desta definição de cânone, o Catálogo Conceptual [Sowa, 1984, Apêndice B] associa um grafo canónico a cada tipo de conceito ou relação, de modo a indicar as restrições que esse tipo deve observar. Esta associação, além de não estar formalmente definida em [Sowa, 1984][6], levou a comunidade das Estruturas Conceptuais a usar o termo "grafo canónico" em dois sentidos diferentes: (1) um grafo que é derivável da base canónica e (2) *o* grafo da base canónica que está associado a um dado tipo. Claro que estes dois sentidos não são incompatíveis, porque (2) implica (1), isto é, cada grafo da base canónica é canónico (por definição). Convém no entanto distinguir estes dois sentidos, pelo que utilizarei o termo *grafo base* para designar um grafo da base canónica.

A questão que se põe pois é se a cada tipo deve estar associado um grafo base ou não. Mark Willems, em comunicação enviada para a lista de distribuição de mensagens, notou que esta questão se resume ao significado intuitivo da base canónica. Se não houver associações, os grafos base apenas representam restrições de selecção sobre as ligações possíveis entre vértices. No entanto, se cada tipo t estiver associado a um grafo base g, isso significa que qualquer grafo que use t deve conter g como subgrafo. Assim, os grafos base passam a especificar os "argumentos" obrigatórios dos tipos. Eis o exemplo apresentado por Willems, ligeiramente modificado.

Exemplo 42. Considere-se os seguintes três grafos, sendo os dois primeiros grafos base:

1. [ACTO] → (AGNT) → [ANIMADO]
2. [PESSOA] ← (AGNT) ← [DAR] → (OBJ) → [OBJECTO]
   [DAR] ↓ (RCPT) → [PESSOA]
3. [DAR] → (AGNT) → [PESSOA: `#João`]

Se não houver associações de grafos a tipos então o grafo 3 é canónico porque pode ser derivado do primeiro grafo base. No entanto, se o grafo 1 estiver associado ao tipo de relação `AGNT` e o segundo grafo estiver associado ao tipo de conceito `DAR`, então o grafo 3 já não é canónico porque lhe faltam os argumentos sobre o recipiente da acção e o objecto dado.

Como se vê, a noção de grafo canónico depende da decisão em associar os grafos base a tipos ou não. Neste trabalho optei pela segunda via, porque a existência de associações traz vários problemas. Em primeiro lugar, deixa de ser válida a ideia intuitiva de que um grafo canónico representa uma situação possível segundo a ontologia dada. No caso concreto do exemplo, o grafo 1 obriga a que os agentes de acções sejam seres animados, pelo que João ser dador é perfeitamente concebível. No entanto, o

[6] Uma formalização muito simples é dada em [Wermelinger e Lopes, 1994]. Trata-se apenas de uma função sobrejectiva entre tipos e grafos.

terceiro grafo não é considerado canónico apenas porque não contém a informação completa. Ao facto da noção de canonicidade ficar pouco intuitiva acresce portanto a dificuldade em especificar conhecimento parcial. Por último, parece-me que a base canónica não é o local apropriado para enumerar os argumentos obrigatórios de verbos. Uma aplicação que utilize língua natural deverá pôr esse tipo de informação no léxico, que será pois um conjunto de pares palavra/grafo canónico, podendo haver vários pares com a mesma palavra no caso de ela ter vários sentidos.

A definição formal de cânone mantém-se pois equivalente à de Sowa. As alterações introduzidas na Hipótese 3.4.5 na página 73 são mínimas e óbvias. Para começar, a hierarquia de tipos T tem que dar lugar a duas hierarquias $\mathcal{T}_\mathcal{C}$ e $\mathcal{T}_\mathcal{R}$ que obedeçam à Hipótese 1 na página 30, à Hipótese 5 na página 34 e à Hipótese 10 na página 35. Além disso, o conjunto de marcadores individuais I tem que ser substituído pela nova definição de marcadores $\mathcal{M}$ da Hipótese 11 na página 37. A inclusão da relação de conformidade no cânone mantém-se, embora a definição usada seja naturalmente a deste trabalho (Hipótese 53 na página 72). Sendo a noção de conformidade parte integrante da noção de canonicidade, e sendo esta última mais forte que a noção de boa-formação, os grafos canónicos terão que ser bem-formados e os conceitos que neles ocorrem devem ser conformes. Ou seja, os grafos canónicos têm que ser conformes (Definição 55 na página 72). Tendo em conta que a base canónica é o conjunto gerador de todos os grafos canónicos, os grafos base também terão que ser conformes.

Hipótese 56. Um *cânone* é um tuplo $\langle \mathcal{T}_\mathcal{C}, \mathcal{T}_\mathcal{R}, \mathcal{M}, ::, \mathcal{B} \rangle$ tal que

- $\mathcal{T}_\mathcal{C}$ é uma hierarquia de tipos de conceito;
- $\mathcal{T}_\mathcal{R}$ é uma hierarquia de tipos de relação;
- $\mathcal{M}$ é um conjunto de marcadores;
- $::$ é uma relação de conformidade;
- a *base canónica* $\mathcal{B}$ é um conjunto finito de grafos conformes, rasos e sem dependências, chamados *grafos base*.

Exemplo 43. Os grafos base [ACTO]→(AGNT)→[ANIMADO] e [OBJECTO]→(CARAC)→[COR] indicam que os agentes de acções são seres animados e que uma das características dos objectos é a sua cor. Qualquer um dos grafos base impede que o grafo conforme [DORMIR]→(AGNT)→[IDEIA]→(CARAC)→[COR: `#verde`], uma representação parcial da famosa frase de Chomsky "Ideias verdes sem cor dormem furiosamente", seja considerado canónico. De facto, como `IDEIA` não é subtipo nem supertipo de `ANIMADO` ou `OBJECTO`, as regras canónicas de formação não poderão derivar este grafo (ver a Secção 5.4 na página 85).

Repare-se que, tal como acontecia na definição original, a base canónica pode ser redundante, ou seja, pode ser possível derivar exactamente os mesmo grafos canónicos a partir de um subconjunto próprio da base canónica. Um sistema de Estruturas Conceptuais poderá facilmente detectar esse caso, bastando verificar para cada grafo base se é derivável dos restantes utilizando o Teorema 63 na página 87.

5.3 Regras Canónicas de Formação

Ao contrário dos processos de percepção e introspecção, que são delegados numa entidade externa ao sistema (o engenheiro do conhecimento), limitando-se a TEC a providenciar um meio de representar os resultados dessas actividades (a base canónica), o processo de formação já faz parte integrante da teoria, nomeadamente através das regras canónicas de formação. Assim, a base canónica é o conjunto inicial de grafos canónicos a partir do quais se poderá gerar todos os outros. Antes de apresentar a definição original de Sowa é preciso nomenclatura adicional.

Definição 3.4.2 (Sowa). Duas relações do mesmo tipo são *duplicadas* se para cada i, o i-ésimo arco de uma está ligado ao mesmo conceito como o i-ésimo arco da outra.

Devido a tentar usar a terminologia da forma o mais rigorosa possível, distingo entre conceito e vértice conceptual: dois vértices distintos podem representar — isto é, estar etiquetados com — o mesmo conceito (Hipótese 33 na página 50). Nesse sentido, nas definições de Sowa "conceito" deve ser entendido como "vértice conceptual" e "relação" significa "vértice relacional".

Posto isto, vejamos as regras canónicas de formação propostas em [Sowa, 1984]. Entre muitas formulações possíveis — de regras equivalentes ou diferentes — a que se segue tem as vantagens de ser extremamente simples e de não conter redundâncias, ou seja, as regras são independentes entre si.

Hipótese 3.4.3 (Sowa). Há quatro *regras canónicas de formação* para derivar um grafo conceptual w a partir dos grafos conceptuais u e v (em que u e v podem ser o mesmo grafo):

Copiar w é uma cópia exacta de u.

Restringir Para qualquer conceito c de u, $tipo(c)$ pode ser substituído por um subtipo; se c for genérico, o seu referente pode ser substituido por um marcador individual. Estas alterações são permitidas apenas se $referente(c)$ é conforme com $tipo(c)$ antes e depois da alteração.

Juntar Se um conceito c de u é idêntico a um conceito d de v, então seja w o grafo obtido apagando d e ligando a c todos os arcos de relações que estavam ligadas a d.

Simplificar Se as relações r e s no grafo u são duplicadas, então uma delas pode ser apagada de u juntamente com todos os seus arcos.

Embora independentes entre si, as regras têm todas um ponto em comum, o qual é a razão fundamental para a escolha destas regras e não de outras: o grafo w é uma especialização de u e v (Teorema 3.5.4 na página 86) pelo que de w se pode inferir u e v (Teorema 3.5.3 na página 107). Uma derivação canónica estabelece assim uma relação entre os grafos iniciais e o grafo resultante que pode ser analisada do ponto de vista da lógica (implicação) ou da teoria dos grafos (projecção).

A pergunta que se coloca agora é: poderemos adoptar as regras de Sowa para este trabalho? A resposta é: sim, mas não chegam. Segundo a metodologia seguida, os grafos verdadeiros devem ser um subconjunto dos grafos canónicos, porque não faz sentido um grafo ontologicamente incorrecto ser considerado verdadeiro. Mas isto implica que as regras de inferência sejam um caso particular das regras canónicas de formação. Dito de outro modo, dado um conjunto de grafos verdadeiros, os grafos derivados a partir deles usando as regras de inferência têm que ser canónicos pelo que também devem poder ser obtidos por aplicação das regras canónicas de formação ao mesmo conjunto de grafos. O inverso é que não é necessariamente verdade: as regras canónicas de formação podem gerar grafos falsos que portanto não devem ser obtidos pelas regras de inferência sob pena de elas não serem coerentes.

É precisamente este o caminho seguido em [Sowa, 1995]. A partir da constatação das semelhanças entre as regras de inferência de primeira ordem (Hipótese 4.3.5 na página 111) e as regras canónicas de formação (Hipótese 3.4.3 na página anterior) — em vários casos uma regra de inferência inclui uma regra canónica ou o seu inverso — Sowa generalizou as regras canónicas de formação passando as regras de inferência a limitar a aplicabilidade das regras de formação. As alterações por ele feitas foram de duas ordens. Por um lado, algumas regras já não se aplicam a vértices individuais ou a um grafo completo (como acontecia na regra de cópia) mas a subgrafos. Por outro lado, as regras não só especializam os grafos sobre os quais se aplicam, mas também passam a poder generalizá-los.

Antes de apresentar a longa definição das novas regras canónicas de formação, convém esclarecer alguns dos termos e convenções nela usados. No seu novo livro, Sowa alterou bastante a noção de referente, distinguindo entre *referentes existenciais* e *quantificadores generalizados*. Estes últimos representam a habitual quantificação universal mas também plurais tais como "muito" e conjuntos tais como "18 pessoas". Por oposição, um referente existencial representa uma só entidade, a qual pode ser um indivíduo ou uma situação complexa. Ou seja, um referente existencial pode ser um marcador individual, o marcador genérico, ou um grafo conceptual, entre outras espécies de referentes que não são relevantes neste trabalho. É também de referir que Sowa já não usa um símbolo especial para o marcador genérico. Sendo o referente por defeito, é representado por espaço em branco.

Definição (Sowa). Seja $\mathcal{C}$ um contexto que contém um grafo conceptual ou uma colecção de grafos conceptuais u. Assuma que todos os conceitos de u imediatamente contidos em $\mathcal{C}$ têm referentes existenciais. Pode haver elos de correferência de conceitos em u para conceitos fora de $\mathcal{C}$ ou para conceitos em contextos encaixados dentro de contextos de u. Os conceitos encaixados dentro de qualquer contexto de u podem ter referentes de qualquer espécie, incluindo quantificadores generalizados.

1. *Regras de equivalência.* As duas primeiras regras transformam u num grafo ou colecção w que é logicamente equivalente a u. Seja v um subgrafo de u que não está encaixado dentro de qualquer contexto de u (v pode ser vazio ou todo o u). O subgrafo v pode ter alguns arcos relacionais ou elos de correferência ligados a conceitos que não estão incluidos em v.

   Copiar A regra da cópia faz uma cópia exacta do subgrafo v e adiciona-o a u para formar w. Se i é um arco ou elo de correferência de v ligado a um conceito c

que não está em v, então w tem que conter uma cópia de i com uma ponta ligada a c e a outra ponta ligada à cópia do conceito ou relação correspondente em v.

Simplificar A regra da simplificação é o inverso da cópia: permite apagar qualquer subgrafo que possa ter sido derivado por cópia. Um subgrafo v_2 de u diz-se ser um *duplicado* de v se v e v_2 forem idênticos e para cada i de v que estiver ligado a um conceito c que não está em v, o correspondente i_2 de v_2 também está ligado a c. Se v_2 é um duplicado de v então pela regra da simplificação v_2 pode ser apagado de u para formar w.

2. *Regras de especialização.* As duas regras seguintes transformam u num grafo ou colecção w que é mais especializada que u.

Restringir Seja c um conceito de u que não está encaixado dentro de um contexto de u (o conceito c tem que ter um referente existencial). Então w pode ser derivado de u restringindo c por tipo ou por referente: a restrição por tipo substitui o tipo de c por um subtipo; e restrição por referente substitui o espaço em branco por outro referente existencial.

Juntar Sejam c e d dois conceitos distintos de u. Nem c nem d podem estar dentro de um contexto de u e tanto c como d têm que ter tipo e referente idênticos. Então w é o grafo obtido apagando d e juntando a c todos os elos de correferência e arcos de relações que estavam ligadas a d.

3. *Regras de generalização.* As duas últimas regras são o inverso das regras de especialização. Elas transformam u num grafo ou colecção w que é uma generalização de u.

Relaxar Seja c um conceito de u que não está dentro de nenhum contexto de u e que tem um referente existencial. Então w pode ser derivado de u relaxando c por tipo ou por referente: o relaxamento por tipo substitui o tipo de c por um supertipo; e relaxamento por referente apaga um referente existencial deixando um espaço em branco.

Separar Seja v um subgrafo de u que não está dentro de nenhum contexto de u. O subgrafo v pode ter alguns arcos relacionais e elos de correferência ligados a conceitos que não estão em v. Então w é o resultado de apagar v de u. Se c não é um conceito de v e estava ligado por um arco a uma relação de v ou por um elo de correferência a um conceito de v, então c fica inalterado, mas o arco ou elo é apagado.

Como o próximo exemplo mostra, as regras já não são independentes entre si, podendo-se chegar ao mesmo resultado a partir do(s) mesmo(s) grafo(s) por aplicação de regras diferentes. Além disso, e como a própria formulação dá a entender, por subgrafo entende-se um subconjunto dos vértices e arcos do grafo original. Um subgrafo não tem pois que ser um grafo conceptual. Em particular, um subgrafo pode ser apenas um vértice relacional. Aliás, só assim se consegue garantir que esta nova versão inclua a antiga regra de simplificação.

Exemplo 44. A partir do grafo $\boxed{\text{ACTO}} \rightarrow (\text{AGNT}) \rightarrow \boxed{\text{ANIMADO}}$ pode-se gerar o grafo

$$\begin{tikzcd} \boxed{\text{ACTO}} \arrow[r] \arrow[dr] & (\text{AGNT}) \arrow[r] & \boxed{\text{ANIMADO}} \\ & (\text{AGNT}) \arrow[ur] & \end{tikzcd}$$

de duas maneiras distintas. A primeira consiste em fazer uma cópia do subgrafo (AGNT), o que mostra que os dois grafos são equivalentes. A segunda envolve começar pela cópia do grafo todo e depois juntar dois dos conceitos idênticos (por exemplo $\boxed{\text{ACTO}}$) seguindo-se a junção dos outros dois conceitos. Inversamente, pode-se voltar ao grafo original pela regra da simplificação ou da separação.

A noção de canonicidade não existe em [Sowa, 1995], pelo que as regras canónicas de formação não são usadas isoladamente mas sim pelas regras de inferência (Definição na página 112). Isto garante em particular que as regras de especialização e generalização nunca são "misturadas", pois se o forem permitem gerar grafos não canónicos como mostra o Exemplo 47 na página 82. Daqui resultam duas opções: ou se adopta o método de Sowa, assumindo que o utilizador de uma base de conhecimentos apenas está interessado nas verdades que o sistema consegue inferir, ou alteram-se as regras canónicas de formação de modo a podê-las utilizar directamente e não apenas através das regras de inferência. Seguirei esta segunda via por várias razões:

- em muitas situações é útil poder obter grafos cuja veracidade é desconhecida à partida[7];
- é desejável ter definições "declarativas" e "operacionais" dos grafos canónicos, assim como aconteceu para os bem-formados e irá acontecer para os verdadeiros;
- as regras canónicas de formação podem formar um ponto de partida para a definição de outras operações que se julguem necessárias.

A utilização directa das regras canónicas de formação implica que poderão ser aplicadas regras de especialização *e* generalização ao mesmo grafo. Torna-se assim necessário estender a noção de projecção (Definição 38 na página 51) de modo a incluir a generalização de vértices. Por sua vez, isto leva a uma nova noção de instância (Definição 39 na página 52).

Definição 57. Sejam $g = \langle V_C, V_R, E, etiqueta\rangle$ e $g' = \langle V'_C, V'_R, E', etiqueta'\rangle$ dois grafos conceptuais. Uma *semi-projecção* $\pi : g \to g'$ é uma função que associa a g um subgrafo de g' chamado *semi-projecção de* g *em* g' tal que

- $\forall c \in V_C \;\; \pi(c) \in V'_C \wedge (etiqueta'(\pi(c)) \leq etiqueta(c) \vee etiqueta'(\pi(c)) \geq etiqueta(c))$
- $\forall r \in V_R \;\; \pi(r) \in V'_R \wedge (etiqueta'(\pi(r)) \leq etiqueta(r) \vee etiqueta'(\pi(r)) \geq etiqueta(r))$
- $\forall e = \langle r, i, c\rangle \in E \;\; \pi(e) = \langle \pi(r), i, \pi(c)\rangle \in E'$

Definição 58. Um grafo conceptual g' é uma *semi-especialização* (ou *semi-restrição*) de um grafo conceptual g e inversamente g é uma *semi-generalização* de g', se existir uma semi-projecção de g em g'. Diz-se que g' é uma *semi-instância* de g se a projecção for uma função bijectiva.

[7] As regras canónicas de formação são por exemplo muito usadas em interpretadores semânticos que fazem parte de sistemas de reconhecimento de língua natural [Sowa e Way, 1986].

De seguida apresentam-se pois as novas regras canónicas de formação. A abordagem adoptada mantém a mesma divisão em três grupos introduzida por Sowa e as regras também são bastante parecidas. As modificações introduzidas surgem da necessidade de gerar grafos ontologicamente correctos, donde conformes. Assim, a aplicabilidade de algumas regras teve que ser restringida. Por outro lado, foi necessário acrescentar regras, não só para tratar grafos hierárquicos e/ou com dependências, mas também para garantir que as regras de inferência sejam de facto um caso particular das regras canónicas de formação, o que não acontece nas abordagens de Sowa. Felizmente não são precisas regras distintas para alcançar estes dois objectivos. Os contextos e as dependências não influem na noção de canonicidade, pelo que bastam regras para os acrescentar e apagar à vontade. Em relação às regras de inferência (Definição na página 112), comparando-as com as regras canónicas de formação (Definição na página 77) verifica-se que também basta acrescentar regras de inserção e remoção.

Hipótese 59. Dados um cânone e zero ou mais grafos conceptuais, as *regras canónicas de formação* geram novos grafos. Em parte, as regras estão definidas à custa das operações de duplicação, remoção e substituição de subgrafos. A noção de subgrafo depende da regra que se quer aplicar, mas seja ela qual for assume-se que as operações têm em conta os arcos e dependências entre vértices desses subgrafos e vértices externos. Além disso, se os grafos conceptuais a que são aplicadas as regras forem conformes, o resultado também tem que ser conforme. No que se segue, c é um contexto vazio ou contendo pelo menos os grafos conceptuais g_1 e g_2, que podem ser o mesmo grafo, e g'_1 e g'_2 são subgrafos de g_1 e g_2, respectivamente.

- *Regras de Equivalência.* Nestas regras, se x for um vértice de um dado subgrafo mas y não, e se existir um arco entre x e y, então y é um vértice conceptual e x um vértice relacional.

  Cópia Pode-se fazer uma cópia de g'_1.

  Simplificação Pode-se apagar g'_1 se g'_1 e g'_2 forem duplicados (isto é, idênticos e ligados aos mesmos vértices externos) mas não tiverem vértices em comum.

- *Regras de Especialização.* Nestas regras, se x for um vértice de um dado subgrafo mas y não, e se existir um arco entre x e y, então y é um vértice relacional e x um vértice conceptual.

  Junção Se g'_1 e g'_2 forem idênticos podem ser sobrepostos.

  Restrição A etiqueta de um vértice v de g_1 pode ser substituída por uma sua especialização desde que v não tenha sido relaxado anteriormente.

  Inserção Em c pode-se inserir um grafo base ou um contexto vazio, seja ele positivo ou negativo. Pode-se acrescentar uma dependência entre um vértice de g_1 e qualquer vértice de qualquer grafo em qualquer contexto.

- *Regras de Generalização.* Nestas regras, se x for um vértice de um dado subgrafo mas y não, e se existir um arco entre x e y, então y é um vértice relacional e x um vértice conceptual.

  Separação Pode-se substituir g_1 por g'_1 se g'_1 for semi-instância de algum grafo base.

Relaxamento A etiqueta de um vértice v de g_1 pode ser substituída por uma sua generalização desde que v não tenha sido restrito anteriormente.

Remoção Pode-se apagar g_1. Pode-se remover qualquer dependência que envolva um vértice de g_1.

Note-se que as regras canónicas de formação devem manter a base de conhecimentos conforme (donde bem-formada) apenas se ela o for originalmente. O utilizador tem assim maior margem de manobra porque pode aplicar as regras a grafos que não sejam conformes, e nesse caso não se exige sequer que o resultado esteja bem-formado. Mas se os grafos originais estiverem conformes, a aplicação de uma regra exige no pior caso a verificação de toda a base de conhecimentos porque uma alteração local pode ter efeitos globais devido às dependências. Se o resultado não for conforme será preciso desfazer a regra. À primeira vista o processo de derivação canónica parece pois ter custos incomportáveis, mas como adiante se verá, na maioria das vezes tal verificação é relativamente eficiente podendo até ser desnecessária.

É também de salientar que a definição das regras ficou muito mais concisa pois assume-se logo desde o início que qualquer operação sobre subgrafos também tem que envolver os arcos e as dependências que os ligam ao "exterior". Fica igualmente bastante claro que condições os subgrafos devem satisfazer. Por último, repare-se que as regras não são independentes entre si na globalidade, mas são-no dentro de cada grupo. As secções seguintes dão uma explicação detalhada de cada regra.

5.3.1 Cópia e Simplificação

A regra da cópia manteve-se tal como na versão de Sowa. Apenas se tornou mais clara a condição a que o subgrafo deve obedecer, pois sem ela é possível derivar grafos mal-formados. Compare-se o exemplo que se segue com o Exemplo 44 na página 78: ambos fazem cópias a partir do mesmo grafo, mas num caso obtém-se um resultado aceitável, no outro não. A diferença reside no subgrafo escolhido.

Exemplo 45. Copiando o subgrafo $\boxed{\text{ACTO}}$ do grafo $\boxed{\text{ACTO}} \rightarrow (\text{AGNT}) \rightarrow \boxed{\text{ANIMADO}}$ obtém-se

$$\begin{array}{ccccc} \boxed{\text{ACTO}} & \rightarrow & (\text{AGNT}) & \rightarrow & \boxed{\text{ANIMADO}} \\ & & \uparrow & & \\ & & \boxed{\text{ACTO}} & & \end{array}$$

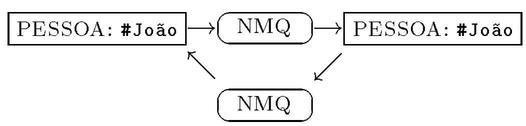
que não está bem-formado porque `AGNT` não é um tipo de relação ternária.

Não há pois problema em copiar um vértice relacional mas o mesmo não se pode dizer de um vértice conceptual, porque enquanto uma relação só pode ter um determinado número de conceitos como argumentos, cada conceito faz parte de tantas relações quantas se queira. Assim, ao copiar um vértice relacional r ligado aos vértices conceptuais $c_1, \ldots, c_n$, apenas se está a aumentar o número de arcos ligados aos c_i, mas não o número de arcos de r. Ao invés, copiar um c_i que está ligado a r implica aumentar o número de arcos de r o que não é permitido. Resumindo, para copiar um vértice conceptual c é preciso copiar também todas os vértices relacionais $r_1, \ldots, r_m$ a que está ligado por arcos. Por outras palavras, se c pertence a um subgrafo que vai ser duplicado, todos os r_j também têm que estar nesse subgrafo. Esta condição é equivalente à dada na Hipótese 59 na página 80.

Repare-se que não é preciso fazer nenhuma verificação em relação à conformidade ou às dependências do grafo resultante. No primeiro caso, se o original obedecer à relação de conformidade, os conceitos da cópia também serão conformes. No segundo caso, a operação de cópia limita-se a duplicar parte das dependências existentes, pelo que estando o original bem-formado nesse aspecto, a cópia também o está.

Passando à regra de simplificação, manteve-se igualmente a ideia base de Sowa. Visto dois subgrafos g'_1 e g'_2 serem duplicados apenas se tiverem as mesmas ligações para o exterior, há duas hipóteses. No primeiro caso não existem arcos para o exterior e portanto $g'_1 = g_1$ e $g'_2 = g_2$ o que significa que se está perante duas cópias de um grafo completo, pelo que não há qualquer problema em eliminar uma delas. No segundo caso $g_1 = g_2$, isto é, os dois subgrafos fazem parte do mesmo grafo, pelo que há problemas quando os subgrafos se sobrepõem.

Exemplo 46. Assuma-se que a relação de altura entre pessoas é designada por `NMQ` ("não maior que"). Então o subgrafo (NMQ)→[PESSOA: `#João`]→(NMQ) ocorre duas vezes no grafo



Eliminando uma das cópias obtém-se apenas [PESSOA: `#João`] que não é equivalente ao original.

Por esta razão exige-se que os subgrafos não tenham vértices em comum. De novo não é preciso fazer a verificação do resultado obtido pois a regra da simplificação limita-se a eliminar dependências e subgrafos redundantes, pelo que se o original estiver bem-formado, o novo grafo também está.

5.3.2 Restrição e Relaxamento

As regras de restrição e relaxamento tiveram que ser alteradas. Em primeiro lugar, porque com a hierarquia de tipos de relação já faz sentido restringir ou relaxar vértices relacionais. Em segundo lugar, porque se aplicadas na sua forma original, isto é, sem impôr quaisquer condições adicionais, as regras tornam a base canónica dispensável além de permitirem substituir um tipo de conceito t por qualquer outro tipo t', mesmo que t e t' sejam incompatíveis.

Exemplo 47. A partir do grafo [ACTO]→(AGNT)→[ANIMADO] pode-se obter por restrição [ACTO]→(AGNT)→[$\perp_c$] e de seguida por relaxamento [ACTO]→(AGNT)→[IDEIA] ou [ACTO]→(AGNT)→[RELAÇÃO]. O passo intermédio também poderia ser um relaxamento para [ACTO]→(AGNT)→[$\top_c$] seguido de uma restrição para o grafo final.

Como se vê, usando as regras originais na presente abordagem pode até levar à geração de grafos mal-formados, além de que as condições impostas pelos grafos base se tornam completamente inúteis. O exemplo mostra que não se pode misturar a restrição com o relaxamento. Na abordagem de Sowa, esta separação é imposta pelas regras de inferência. Passando-a para as regras canónicas de formação há duas possibilidades: ou impede-se a mistura ao nível do grafos — isto é, um grafo que tenha sido restrito não pode ser relaxado e vice-versa — ou ao nível dos vértices, isto é, pode-se fazer restrições e relaxamentos sobre o mesmo grafo mas não sobre o mesmo vértice. A segunda opção é obviamente a mais flexível pelo que foi a adoptada na Hipótese 59 na página 80.

Há no entanto duas desvantagens. Uma delas é que certos vértices não podem mais ser alterados, nomeadamente quando são obtidos por junção de dois vértices tais que um deles foi restrito e o outro relaxado.

Exemplo 48. Seja `PESSOA` < `ANIMAL` < `ANIMADO` e seja `NMQ` a relação de altura entre pessoas como no Exemplo 46 na página anterior. Então a partir dos grafos conceptuais

[ACTO]→(AGNT)→[ANIMADO]   [PESSOA]→(NMQ)→[PESSOA]

pode-se obter [ACTO]→(AGNT)→[ANIMAL]→(NMQ)→[PESSOA] por restrição de [ANIMADO] e relaxamento de [PESSOA] seguido de junção de ambos. O conceito [ANIMAL] já não pode ser livremente restrito (ou relaxado). Caso contrário obter-se-ia por exemplo [ACTO]→(AGNT)→[CÃO]→(NMQ)→[PESSOA] que viola a assinatura de `NMQ`.

A outra desvantagem das novas regras é obrigar um sistema de Estruturas Conceptuais a lembrar-se da "história" de cada vértice. Uma implementação possível consiste em ter duas variáveis booleanas para cada vértice, uma indicando se ele já foi restrito e a outra se já foi relaxado. Assim, antes de iniciar uma operação de restrição, por exemplo, o sistema testa a variável correspondente ao relaxamento. Se tiver o valor 'falso' a regra pode ser aplicada e no fim a variável da restrição é posta a 'verdade'. Sempre que dois vértices são sobrepostos devido à junção, basta fazer a conjunção das respectivas variáveis.

Por fim, note-se que a especialização ou generalização de um grafo bem-formado pode estar mal-formada, como o Exemplo 14 na página 44 e o Exemplo 19 na página 48 indicam. Portanto, e ao contrário do que acontecia com a cópia e a simplificação, as regras de restrição e relaxamento exigem que no caso geral se faça a verificação do resultado quanto à conformidade e boa-formação.

5.3.3 Junção e Separação

A regra de junção também foi generalizada: em vez de se juntar um vértice conceptual de cada vez, juntam-se subgrafos iguais mas possivelmente fazendo parte de grafos diferentes. Nesse caso tem-se uma *junção externa* (segundo a nomenclatura de [Mugnier e Chein, 1993b]). Como a junção de dois vértice relacionais envolve automaticamente a junção dos seus argumentos, a condição a que os subgrafos devem obedecer é exactamente a simétrica da usada para as regras de cópia e simplificação.

Exemplo 49. Juntando os grafos [LASANHA]←(OBJ)←[COMER]→(AGNT)→[GATO: `#Garfield`] e [LASANHA]←(OBJ)←[COMER]→(MANR)→[DEPRESSA] através do subgrafo comum [LASANHA]←(OBJ)←[COMER] obtém-se

```
[LASANHA]←(OBJ)←[COMER]→(AGNT)→[GATO: #Garfield]
                   ↓
                 (MANR)→[DEPRESSA]
```

É também de salientar que a simplificação pode ser simulada à custa de uma *junção interna*, isto é, quando os dois subgrafos fazem parte do mesmo grafo. De facto, a sobreposição de dois subgrafos equivale a eliminar um deles mas mantendo as suas ligações com o exterior. Como no caso da simplificação os dois subgrafos têm exactamente as mesmas ligações para o exterior, obtém-se o efeito da sobreposição. Vejamos um exemplo usando o grafo do Exemplo 46 na página 82.

Exemplo 50. O subgrafo [PESSOA: `#João`]→(NMQ)→[PESSOA: `#João`] ocorre duas vezes no grafo

```
[PESSOA: #João]→(NMQ)→[PESSOA: #João]
             ↖           ↙
                (NMQ)
```

Sobrepondo as duas ocorrências tem-se [PESSOA: `#João`]→(NMQ)→[PESSOA: `#João`]. O mesmo resultado pode ser obtido pela simplificação aplicada ao subgrafo (NMQ).

Quanto à regra da separação, se se pretende que ela seja o inverso da junção, então a sua formulação deveria ser: “pode-se separar um grafo g em dois grafos g_1 e g_2 se a sua junção der g”. Infelizmente esta definição permite obter grafos não canónicos.

Exemplo 51. Considere-se os dois grafos base [ACTO]→(AGNT)→[ANIMADO] e [INSTRUMENTO]←(INST)←[ACTO]→(OBJ)→[OBJECTO]. Ao escrever os grafos desta maneira e não de outra, pretende-se impôr a condição de que um acto tem necessariamente um agente, ou um instrumento e um objecto. Fazendo a junção obtém-se

```
[INSTRUMENTO]←(INST)←[ACTO]→(OBJ)→[OBJECTO]
                        ↓
                     (AGNT)→[ANIMADO]
```

e separando de novo pode-se ter [INSTRUMENTO]←(INST)←[ACTO] e [OBJECTO]←(OBJ)←[ACTO]→(AGNT)→[ANIMADO], dois grafos que violam as restrições ontológicas.

O problema parece pois ser o facto de se fazer a separação no sítio errado. Uma solução imediata é exigir que os grafos g_1 e g_2 obtidos pela separação sejam canónicos. No entanto a noção de grafo canónico ainda não foi formalmente introduzida. Antecipando um pouco a Secção 5.4 na próxima página, a definição de grafo canónico vai basear-se justamente na das regras canónicas de formação, tal como em [Sowa, 1984]. Assim a inclusão do termo "grafo canónico" na formulação da regra da separação levaria a uma definição circular.

A regra de Sowa (Definição na página 77) também não serve porque equivale a separar o grafo arbitrariamente e depois eliminar uma das partes. O que é preciso é garantir que a parte que sobra está ontologicamente correcta. Nesta altura do trabalho, os únicos grafos para os quais há essa a certeza são os grafos base. Deste modo obtém-se a regra da Hipótese 59 na página 80: pode-se eliminar parte de um grafo desde que se garanta que o resto é um grafo base (a menos de algumas restrições e/ou relaxamentos). Note-se que por aplicação repetida desta regra é possível separar um grafo nos seus componentes base.

5.3.4 Inserção e Remoção

A regra da remoção permite apagar grafos (incluindo contextos positivos ou negativos) e dependências de um forma arbitrária pois se o original estava bem-formado, o resultado também o está. Quanto à inserção, de novo a formulação mais geral seria "pode-se inserir um grafo canónico no contexto c". Pelas razões acima indicadas, os únicos grafos que se podem inserir (sem ter que fazer qualquer género de verificação) são os grafos base. Em relação aos contextos, repare-se que as regras dadas nas secções anteriores operam dentro do mesmo contexto. É preciso pois uma regra para criar novos contextos onde as restantes operações possam ser aplicadas. Essa regra é igualmente a da inserção. Para garantir que o resultado está bem formado, apenas se permite inserir contextos vazios. Os novos contextos podem ser "populados" com grafos base através da própria regra de inserção. Finalmente, em relação às dependências, é preciso fazer a verificação do resultado, como acontece com as outras regras de especialização. No entanto, o custo de tal processo é bastante atenuado se for usada uma versão incremental do algoritmo que resolve o sistema de condições associado às dependências [Menezes, 1995].

5.4 Grafos Canónicos

Munidos de um conjunto gerador (a base canónica) e de regras de geração (as regras canónicas de formação), podemos finalmente definir a noção de grafo canónico, que será equivalente à da Hipótese 3.4.5 na página 73[8]. Apenas a formulação difere. Enquanto Sowa apenas utiliza a relação de conformidade na definição das regras, pelo que é preciso dizer explicitamente qual o conjunto inicial de grafos ao qual as regras devem ser aplicadas, na presente abordagem esse conjunto (a base canónica) já faz parte da definição das regras canónicas de formação (Hipótese 59 na página 80). Assim, um grafo canónico é um grafo que pode ser criado a partir de um cânone e da "folha em branco".

[8] Para uma formulação alternativa e mais detalhada ver [Mugnier e Chein, 1993b].

Definição 60. Um grafo conceptual diz-se *canónico* em relação a um dado cânone C se for possível deriva-lo a partir do conjunto vazio de grafos no contexto exterior por aplicação de uma ou mais regras canónicas de formação usando a base canónica de C.

Note-se que um mesmo grafo pode ser considerado canónico ou não, consoante o cânone que se está a usar. Da definição anterior é óbvio que, tal como se esperava,

Proposição 61. *Um grafo base é canónico (em relação ao cânone a que pertence).*

Demonstração. A regra da inserção permite colocar qualquer grafo base em qualquer contexto, em particular no contexto exterior. □

Mas para além de saber formar grafos canónicos convém também conseguir reconhece-los sem ter que construir explicitamente a derivação (isto é, a sequência de regras) que leva à sua formação. Na definição original de Sowa, as regras canónicas de formação limitam-se a especializar um grafo pelo que o processo de derivação corresponde a uma projecção.

Definição 3.5.1 (Sowa). Se um grafo conceptual u for canonicamente derivável de um grafo conceptual v (possivelmente com a junção de outros grafos conceptuais $w_1, \ldots, w_n$), então u é chamado uma *especialização* de v, e escreve-se $u \leq v$, e v é chamado uma *generalização* de u.

Teorema 3.5.4 (Sowa). [9] *Para quaisquer grafos conceptuais u e v em que $u \leq v$, tem que existir uma função $\pi : v \rightarrow u$ em que $\pi(v)$ é um subgrafo de u chamado* projecção *de v em u. O operador de projecção tem as seguintes propriedades:*

- *Para todo o conceito c de v, $\pi(c)$ é um conceito de $\pi(v)$ em que tipo$(\pi(c)) \leq$ tipo(c). Se c é individual, então referente(c) = referente$(\pi(c))$.*
- *Para toda a relação r de v, $\pi(r)$ é uma relação de $\pi(v)$ com tipo$(\pi(r))$ = tipo(r). Se o i-ésimo arco de r está ligado a um conceito c de v, o i-ésimo arco de $\pi(r)$ tem que estar ligado a $\pi(c)$ em $\pi(v)$.*

A função π não é necessariamente injectiva: se x_1 e x_2 forem dois conceitos ou relações tais que $x_1 \neq x_2$, pode acontecer que $\pi(x_1) = \pi(x_2)$. A função π não é necessariamente única: o grafo v pode ter outra projecção $\pi'(v)$ em u tal que $\pi'(v) \neq \pi(v)$.

Repare-se que estas definições são menos gerais que as dadas na Secção 4.1 na página 50 porque se baseiam na noção de derivação canónica. No entanto, a noção de projecção é idêntica à da Definição 38 na página 51, à parte do facto óbvio de não permitir a restrição de marcadores individuais e de vértices relacionais. A interdependência entre derivação canónica e projecção é ainda mais forte do que o teorema anterior faz supor. Mugnier e Chein [1993b] demonstraram que se houver uma projecção entre dois grafos canónicos então também há uma derivação. Daqui, e de outros resultados intermédios, Mugnier e Chein obtiveram um resultado que permite caracterizar os grafos

[9] As demonstrações deste e outros teoremas podem ser consultadas em [Sowa, 1984].

canónicos: um grafo conceptual g é canónico se houver projecções de grafos base em g que cubram todo o g. Como as novas regras canónicas de formação também permitem a generalização de vértices, a projecção é substituida pela semi-projecção, mas a ideia principal mantém-se.

Definição 62. Uma *cobertura* de um grafo conceptual g é um conjunto finito de grafos conceptuais $\{g_1, \ldots, g_n\}$ tais que cada g_i é um subgrafo de g e cada vértice e arco de g ocorre em pelo menos um dos grafos da cobertura.

Teorema 63. *Um grafo conceptual raso e sem dependências é canónico em relação a um cânone C se e só se for conforme e tiver uma cobertura G tal que cada grafo $g \in G$ é semi-instância de um grafo base de C.*

Demonstração. Para provar que a canonicidade de um grafo implica a existência de uma cobertura basta mostrar que existe uma derivação canónica do grafo tal que o teorema é satisfeito (1) pelos grafos iniciais e (2) pela aplicação de cada regra. Para simplificar a demonstração, repare-se que cada derivação canónica pode ser posta numa "forma normal". Como já foi dito em secções anteriores, a regra da simplificação corresponde a uma junção interna, enquanto que a separação pode ser substituída pela inserção de um grafo base seguida de zero ou mais restrições e/ou relaxamentos. Por fim, a cópia de um subgrafo próprio g' do grafo g pode ser obtida duplicando g e fazendo a junção de todos os vértices conceptuais de g que não estão em g' mas que têm arcos para vértices relacionais de g'. Por outras palavras, cada grafo canónico pode ser derivado usando apenas as regras de inserção, restrição, relaxamento, junção e cópia (de grafos completos). Como um grafo canónico é obtido a partir do nada, o processo tem que forçosamente começar pela inserção de todos os grafos base necessários no resto da derivação. A parte (1) fica assim trivialmente demonstrada.

Em relação à parte (2) é preciso mostrar que se o grafo g é obtido por cópia, restrição ou relaxamento do grafo g_1 ou por junção dos grafos g_1 e g_2, ambos conformes e com coberturas G_1 e G_2 nas condições do teorema, então (a) g é conforme e (b) g tem uma cobertura G nas condições do teorema. A afirmação (a) é trivial pois é consequência imediata da própria definição das regras canónicas de formação. Para as regras de cópia e junção (b) é óbvio: G é igual a G_1 ou à união de G_1 e G_2. Quanto à restrição (ou relaxamento), g é idêntico a g_1 a menos de um vértice v_1 que passou a v. Então G obtém-se de G_1 substituindo v_1 por v em todos os grafos de G_1 em que ocorre. Falta provar que os grafos de G assim obtidos também são semi-instâncias de grafos base.

Vai-se utilizar o seguinte lema trivial: se $\pi : g \to g'$ e $\pi' : g' \to g''$ forem duas semi-projecções e se π' não relaxar (respectivamente restringir) nenhum vértice restrito (respectivamente relaxado) por π, então a composição de funções $\pi'\pi$ também é uma semi-projecção. Seja b um grafo base, g'_1 um elemento de G_1 contendo v_1 e seja $g' \in G$ contendo v. Então, por hipótese, existe $\pi : b \to g'_1$. Além disto, como g' é obtido de g'_1 por restrição ou relaxamento, $\pi' : g'_1 \to g'$ também é semi-projecção. Visto as regras canónicas de formação obedecerem à condição do lema, a composição $\pi'\pi : b \to g'$ é uma semi-projecção.

Provar a implicação inversa do teorema é muito mais fácil: basta construir uma derivação canónica para cada grafo g nas condições do teorema. Seja então g um grafo conceptual conforme com cobertura $\{g_1, \ldots, g_n\}$ tal que cada g_i é semi-instância de um grafo base b_i. Como g é conexo, cada g_i tem que ter pelo menos um vértice conceptual em comum com pelo menos um g_j, sendo $j \neq i$. Além disso, como g é conforme, cada g_i também o é, pelo que pode ser derivado pelas regras canónicas de formação a partir do respectivo b_i. A derivação canónica de g consiste então em inserir os b_i, aplicar restrições e relaxamentos até obter os g_i e finalmente juntar os vértices idênticos. $\square$

O reconhecimento de grafos canónicos hierárquicos ou dependentes é bastante fácil porque a regra de inserção permite adicionar contextos e dependências de modo arbitrário, desde que o grafos resultantes estejam bem-formados. A Proposição seguinte é pois imediata.

Proposição 64. *Um grafo conceptual hierárquico e/ou dependente é canónico se e só se for bem-formado e o(s) grafo(s) subjacente(s) for(em) canónico(s).*

Os dois resultados anteriores merecem alguns comentários. Em primeiro lugar, a demonstração do teorema não implica de modo algum que as regras de simplificação e separação sejam inúteis, pois elas são necessárias quando a derivação não é feita a partir do "nada", mas sim a partir de grafos que já se sabem ser canónicos. Além disso, a divisão das regras canónicas de formação em três grupos (necessários para as regras de inferência) obriga a ter uma certa redundância. Por exemplo, a simplificação é uma junção que mantém o resultado equivalente ao original.

Em segundo lugar é de realçar a importância do teorema (e da Proposição) que advém do facto de permitir obter um algoritmo para verificar se um grafo conceptual g (dado pelo utilizador, por exemplo) é canónico ou não. O algoritmo consiste basicamente em encontrar grafos base cujas semi-projecções em g cubram g totalmente. Mas como a semi-projecção de um vértice relacional também tem que incluir os vértices conceptuais a que está ligado, o algoritmo resume-se ao seguinte. Escolhe-se um vértice relacional v de g e percorre-se a base canónica até encontrar um grafo base cuja semi-projecção em g contenha v. Todos os vértices relacionais cobertos pela semi-projecção são marcados e o processo continua com um dos vértices relacionais de g ainda não marcados. Este algoritmo é idêntico ao apresentado em [Mugnier e Chein, 1993b] exceptuando a substituição da projecção pela semi-projecção. Assim, mantém-se a propriedade da complexidade do algoritmo ser polinomial em relação à da semi-projecção. Na maioria das vezes os grafos base são árvores, pelo que a semi-projecção é de complexidade polinomial. Daqui resulta que, nesses casos, reconhecer um grafo canónico demora tempo proporcional ao tamanho do grafo e da base canónica.

Embora a nova noção de canonicidade use a semi-projecção, a operação de projecção não deixa de ter importância, nomeadamente para a organização dos grafos conceptuais numa hierarquia. A pesquisa e a inserção de grafos na base de conhecimentos tornam-se assim muito mais eficientes [Ellis, 1992; Ellis, 1993].

Teorema 3.5.2 (Sowa). *A generalização define uma ordenação parcial dos grafos conceptuais chamada* hierarquia de generalização. *Para quaisquer grafos conceptuais u, v e w, as seguintes propriedades são verdade:*

Reflexividade $u \leq u$.

Transitividade *Se $u \leq v$ e $v \leq w$, então $u \leq w$.*

Anti-simetria *Se $u \leq v$ e $v \leq u$, então $u = v$.*

Subgrafo *Se v é um subgrafo de u, então $u \leq v$.*

Subtipos *Se u é idêntico a v excepto que um ou mais tipos de v estão restritos a subtipos em u, então $u \leq v$.*

Indivíduos *Se u é idêntico a v excepto que um ou mais conceitos genéricos de v estão restritos a conceitos individuais em u, então $u \leq v$.*

Topo *O grafo* `[⊤]` *é a generalização de todos os outros grafos conceptuais.*

Embora Sowa esteja a usar a noção de $\leq$ (generalização) dada pela sua Definição 3.5.1 na página 86, o teorema também é aplicável usando a minha (Definição 39 na página 52), porque ambas se baseiam na projecção entre grafos. Infelizmente, tanto num caso como no outro o teorema tem um ligeiro erro: $\leq$ não é uma ordem parcial porque não é anti-transitiva, como já foi notado por Gerard Ellis e muitos outros [Chein e Mugnier, 1992]. O Exemplo 44 na página 78 mostra dois grafos que, sendo diferentes, são no entanto projectáveis um no outro. Será pois mais correcto dizer que a hierarquia de generalização é um conjunto parcialmente ordenado das classes de equivalência de grafos induzidas pela operação de projecção.

Capítulo 6

Lógica

O cânone permite ao engenheiro do conhecimento especificar uma ontologia que serve de base para a construção de uma base de conhecimentos que represente uma teoria sobre um determinado problema. Assim, várias bases de conhecimentos (de um ou mais utilizadores) podem partilhar o mesmo cânone. Basta ao utilizador acrescentar à base canónica um conjunto de grafos conceptuais assumidos como verdadeiros, a partir dos quais pretende inferir nova informação. As regras canónicas de formação apenas garantem que os grafos por elas gerados estão ontologicamente correctos, mas nada se pode dizer sobre a sua veracidade. São precisas pois regras de inferência para os grafos conceptuais.

Este capítulo mostra como a teoria descrita até ao momento cobre a lógica de primeira ordem. A próxima secção mostra como se pode traduzir grafos em fórmulas lógicas. Apenas estaremos interessados em fórmulas fechadas porque uma fórmula com variáveis livres é sempre verdadeira (isto é, independente da avaliação das variáveis) se e só se o seu fecho universal o for. Apresentarei primeiro a tradução dada por Sowa para a TEC original e mostrarei que a sua definição está incompleta porque não permite obter todas as fórmulas lógicas de primeira ordem. A nova tradução segue as linhas gerais da abordagem de Sowa: os tipos correspondem a predicados, os marcadores a constantes e os elos de correferência (as únicas dependências usadas neste capítulo) representam a igualdade dos referentes ligados.

A segunda secção define uma interpretação para os grafos conceptuais que dá significado às condições de boa-formação e à hierarquia de tipos. Como já foi dito anteriormente, um tipo de conceito não-relacional de ordem $n+1$ denota um conjunto de tipos de ordem n, um tipo relacional de ordem n denota um conjunto de relações de ordem não superior a n, e uma relação de aridade n e ordem i denota um conjunto de tuplos n-ários em que cada elemento é de ordem inferior a i. Além disso, se t for subtipo de t', o conjunto denotado por t é um subconjunto do conjunto denotado por t'.

Finalmente, a terceira e última secção deste capítulo apresenta as regras de inferência, distinguindo-se o caso proposicional do geral para facilitar a leitura. As regras de inferência são um caso particular das regras canónicas de formação, pois devido à classificação dos grafos em quatro níveis, um grafo verdadeiro tem que ser canónico. Neste capítulo assume-se portanto que os grafos a traduzir e inferir são todos bem-formados e obedecem à base canónica.

6.1 Tradução

O primeiro passo na exposição dos fundamentos lógicos dos grafos conceptuais consiste em encontrar uma correspondência entre grafos e fórmulas de uma linguagem de primeira ordem. Essa correspondência é dada por um algoritmo de tradução. A subsecção seguinte apresenta a definição original de Sowa e mostra as suas limitações. Na segunda subsecção é dada uma nova versão juntamente com uma definição rigorosa da linguagem de primeira ordem usada. Na TEC original, essa linguagem encontra-se implicitamente definida pelo processo de tradução.

6.1.1 A Definição de Sowa

A transformação de grafos conceptuais em fórmulas da lógica de primeira ordem está a cargo de um operador específico, denominado ϕ. A sua definição encontra-se espalhada por várias hipóteses em [Sowa, 1984], consoante a complexidade do grafo a que se aplica. No caso mais simples, sem contextos nem elos de correferência, a tradução é imediata. Um tipo de conceito corresponde a um predicado monádico, um tipo de relação a um predicado com a mesma aridade que a relação, e os argumentos desses predicados são dados pelos referentes dos conceitos. Um marcador individual corresponde a uma constante e o marcador genérico traduz-se numa variável quantificada existencialmente, ou não fossem os grafos conceptuais "descendentes" dos grafos existenciais de Peirce. A fórmula denotada por um grafo completo é então a conjunção destas sub-fórmulas.

Hipótese 3.3.2 (Sowa). O operador ϕ transforma grafos conceptuais em fórmulas do cálculo de predicados de primeira ordem. Se u é um grafo conceptual qualquer, então ϕu é a fórmula determinada pela seguinte construção:

1. Se u contém k conceitos genéricos, afecte um símbolo de variável distinto $x_1, \ldots, x_k$ a cada um.
2. Para cada conceito c de u, seja $identificador(c)$ a variável afecta a c se c for genérico, ou $referente(c)$ se c for individual.
3. Represente cada conceito c como um predicado monádico cujo nome é o mesmo que $tipo(c)$ e cujo argumento é $identificador(c)$.
4. Represente cada relação n-ádica r de u como um predicado n-ádico cujo nome é o mesmo que $tipo(r)$. Para cada i de 1 até n, o i-ésimo argumento do predicado é o identificador do conceito ligado ao i-ésimo arco de r.
5. Então ϕu tem o prefixo de quantificação $\exists x_1 \exists x_2 \ldots \exists x_k$ e o corpo é a conjunção de todos os predicados para os conceitos e relações de u.

Exemplo 52. O grafo $g = \boxed{\text{GATO: \texttt{\#Garfield}}} \leftarrow (\text{AGNT}) \leftarrow \boxed{\text{PERSEGUIR}} \rightarrow (\text{OBJ}) \rightarrow \boxed{\text{CÃO}}$ indica que existem um cão e uma perseguição particular que estão relacionados com o indivíduo Garfield. Seguindo a convenção habitual de escrever predicados com maiúscula inicial, obtém-se

$$\phi(g) = \exists x \exists y \; Gato(garfield) \wedge Perseguir(x) \wedge Cão(y) \wedge Agnt(x, garfield) \wedge Obj(x, y)$$

Embora a hipótese anterior pareça trivial, tem um problema: não dá nenhum tratamento especial aos tipos pré-definidos $\top$ e $\bot$ (Hipótese 3.2.5 na página 28). Aplicando a definição de ϕ, estes dois tipos são traduzidos em dois predicados que em nada se distinguem dos outros. Isto é coerente com as regras de inferência (Hipótese 4.3.5 na página 111) que também não tratam $\top$ e $\bot$ de maneira diferente. No entanto, não corresponde à ideia intuitiva que se tem do topo e da base de um reticulado, nomeadamente que o topo representa o domínio completo e a base representa o conjunto vazio de indivíduos. Em termos lógicos, isto equivale a dizer que $\top$ deve ser traduzido pela constante 'verdade' e $\bot$ deve corresponder à falsidade. O facto destas noções informais não se encontrarem reflectidas nas definições formais de Sowa é uma lacuna grave pois, como veremos adiante, sem elas não é possível aos grafos conceptuais representarem todas as fórmulas de primeira ordem.

Continuando com a definição de ϕ, o próximo passo é cobrir o caso dos grafos hierárquicos. De novo, a tradução é bastante simples: um contexto representa a conjunção dos grafos que nele se encontrem. Se o contexto for negativo, a fórmula obtida deverá ser negada.

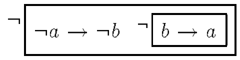
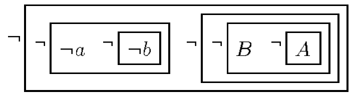
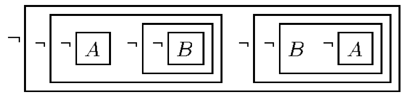
Hipótese 4.2.3 (Sowa). Se p é uma proposição asserindo os grafos $\{u_1, \ldots, u_n\}$, então ϕp é a fórmula $(\phi u_1 \wedge \ldots \wedge \phi u_n)$. Se c é um contexto negativo formado pela relação (NEG) ligada a uma proposição p, então ϕc é $\neg\phi p$. O operador de fórmula ϕ associa símbolos de variável distintos a todos os conceitos genéricos que ocorram em p ou em qualquer contexto dentro de p.

Exemplo 53. Sejam A e B grafos arbitrários representando as proposições a e b, isto é, $\phi(A) = a$ e $\phi(B) = b$. Então a implicação $a \to b$ é representada pelo grafo $g = \neg\boxed{A\ \neg\boxed{B}}$. De facto, aplicando a definição de ϕ obtém-se

$$\phi(g) = \neg(\phi(A) \wedge \neg\phi(B)) = \neg(a \wedge \neg b) = \neg a \vee b = a \to b$$

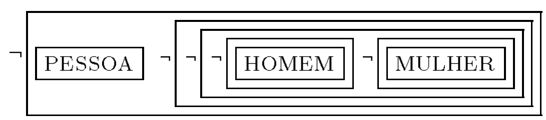
Para traduzir fórmulas com várias implicações, basta aplicar o padrão sistematicamente a cada sub-fórmula. Eis três exemplos que serão utilizados na demonstração da completude das regras de inferência proposicionais (Teorema 69 na página 109).

1.
   - $a \to (b \to a)$
   - $\neg\boxed{A\ \neg\boxed{b \to a}}$
   - $\neg\boxed{A\ \neg\boxed{\neg\boxed{B\ \neg\boxed{A}}}}$
2.
   - $(a \to (b \to c)) \to ((a \to b) \to (a \to c))$
   - $\neg\boxed{a \to (b \to c)\ \neg\boxed{(a \to b) \to (a \to c)}}$
   - $\neg\boxed{\neg\boxed{A\ \neg\boxed{b \to c}}\ \neg\boxed{\neg\boxed{a \to b\ \neg\boxed{a \to c}}}}$
   - $\neg\boxed{\neg\boxed{A\ \neg\boxed{\neg\boxed{B\ \neg\boxed{C}}}}\ \neg\boxed{\neg\boxed{\neg\boxed{A\ \neg\boxed{B}}\ \neg\boxed{\neg\boxed{A\ \neg\boxed{C}}}}}}$

3. • $(\neg a \rightarrow \neg b) \rightarrow (b \rightarrow a)$
   • 
   • 
   • 

O exemplo que se segue ilustra dois pontos importantes que podem não ser facilmente perceptíveis numa leitura menos atenta da Hipótese 4.2.3 na página anterior. Em primeiro lugar, uma negação nunca aparece imediatamente antes de um predicado. Há sempre pelo menos um quantificador existencial pelo meio. Em segundo lugar, um contexto corresponde no fundo a um par de parêntesis que servem para delimitar o escopo das variáveis introduzidas nesse contexto. Assim, se c for um contexto contendo o contexto c', na tradução de c' pode-se reutilizar as variáveis de c[1].

Exemplo 54. O grafo $\neg\boxed{\boxed{\text{CÃO}}}$ corresponde à fórmula $\neg(\exists x\ Cão(x))$. A afirmação "se existir uma pessoa, então existe um homem ou existe uma mulher" pode ser representada por $(\exists x\ Pessoa(x) \rightarrow (\exists x\ Homem(x)) \vee (\exists x\ Mulher(x)))$. Sendo $a \vee b$ equivalente a $\neg(\neg a \wedge \neg b)$ obtém-se



Assim como aconteceu com a Hipótese 3.3.2 na página 92, a segunda parte da definição de ϕ também está incompleta. Neste caso já não se trata de como traduzir os tipos $\top$ e $\bot$, mas sim de como traduzir um contexto vazio. De acordo com a Hipótese 4.2.3 na página anterior, a aplicação de ϕ ao grafo $\boxed{}$, que é a abreviatura de $\boxed{\text{PROPOSIÇÃO: \{\}}}$, resulta na fórmula () que não obedece à sintaxe habitual de uma linguagem de primeira ordem porque não contém nenhum literal. Poder-se-á perguntar se o contexto vazio é relevante ou se é algum caso patológico que não necessita de tradução. Segundo [Sowa, 1984, p. 151],

> "um conjunto vazio de grafos não faz nenhuma asserção. Por convenção, assume-se que é verdadeiro. A negação do conjunto vazio, chamada *cláusula vazia*, tem pois que ser falsa; é escrita como `¬[ ]`."

Esta convenção explica porque é que o conjunto vazio de grafos é um axioma lógico (Hipótese 4.3.1 na página 107), o que prova a necessidade de um tratamento especial para o contexto vazio, donde para a cláusula vazia. Além disso, os exemplos da Secção 6.3 na página 107 mostram que a aplicação das regras de inferência podem gerar contextos vazios. Por todas estas razões, e ao contrário do que sucede de momento, a aplicação $\phi(\boxed{})$ tem que estar definida e o resultado deve reflectir a convenção mencionada.

[1]Como se verá adiante, isto só funciona se não houver elos de correferência de c para c'. Por isso Sowa requer desde já que se usem variáveis distintas em todos os contextos.

A terceira e última parte da definição formal de ϕ diz respeito a grafos com linhas de identidade. Neste caso a tradução resulta numa fórmula da lógica de primeira ordem com igualdade. Um elo de correferência indica que dois conceitos representam o mesmo indivíduo ou indivíduos diferentes (se o elo de correferência atravessar um contexto negativo). Por isso, a tradução destes casos envolve copiar o identificador do conceito dominante para o dominado, mas não vice-versa por causa da negação.

Hipótese 4.2.6 (Sowa). Se u for um grafo conceptual contendo uma ou mais linhas de identidade, calcule a fórmula ϕu transformando primeiro o grafo u segundo o seguinte algoritmo:

- associe um nome de variável único a cada conceito genérico de u;
- para todo o a no conjunto dos conceitos dominantes de u fazer
  - $x := \mathit{identificador}(a)$;
  - acrescente "$=x$" ao campo do referente de cada conceito dominado por a;
- apague todos os elos de correferência de u.

A fórmula ϕu é o resultado de aplicar ϕ à versão transformada de u com a seguinte regra para tratar conceitos com múltiplos referentes: se b for um conceito de u da forma $[t : x_1 = x_2 = \ldots = x_n]$, então ϕb tem a forma $t(x_1) \wedge x_1 = x_2 \wedge \ldots \wedge x_1 = x_n$.

Exemplo 55. O grafo do Exemplo 54 na página anterior não representa a afirmação "todas as pessoas são homens ou mulheres". Para tal é necessário introduzir uma linha de identidade mostrando que se existir uma pessoa então *essa* pessoa tem que ser homem ou mulher. Transformando os elos de correferência segundo o algoritmo acima dado obtém-se o grafo

$$\neg\boxed{\boxed{\text{PESSOA: } x} \ \neg\boxed{\neg\boxed{\neg\boxed{\boxed{\text{HOMEM: } y = x}} \ \neg\boxed{\boxed{\text{MULHER: } z = x}}}}}$$

A aplicação de ϕ resulta na fórmula

$$\neg(\exists x\ \mathit{Pessoa}(x) \wedge \neg\neg(\neg(\exists y\ \mathit{Homem}(y) \wedge y = x) \wedge \neg(\exists z\ \mathit{Mulher}(z) \wedge z = x)))$$

que é equivalente à pretendida: $\forall x\ \mathit{Pessoa}(x) \rightarrow \mathit{Homem}(x) \vee \mathit{Mulher}(x)$.

Este exemplo mostra várias coisas. Em primeiro lugar, quando um elo de correferência liga dois conceitos genéricos, pode-se utilizar a mesma variável para ambos. A cópia dos identificadores só é necessária para os marcadores individuais, como acontece no exemplo "Rosalie não é Rosann" dado na Secção 1.2 na página 16. Em segundo lugar, repare-se na diferença entre o "esqueleto" do grafo do exemplo anterior e o do Exemplo 54 na página anterior. O padrão $\neg\boxed{A \ \neg\boxed{B}}$ é traduzido para $\exists x\ A(x) \rightarrow \exists y\ B(y)$ enquanto $\neg\boxed{A \cdot \neg\boxed{\cdots B}}$ representa $\forall x\ A(x) \rightarrow B(x)$. Em terceiro lugar, um elo de correferência de um contexto dominante c para um contexto dominado c' indica a presença de uma variável livre em c'. Como as linhas de identidade não podem ter "pontas soltas" (isto é, tem que haver conceitos em ambos os extremos de cada elo de correferência), significa isto que qualquer variável livre tem que ser introduzida anteriormente por um predicado.

Chegamos assim finalmente ao fulcro da questão: os grafos conceptuais, tais como definidos formalmente em [Sowa, 1984] através de ϕ, não conseguem representar todas as fórmulas fechadas da lógica de primeira ordem. Considere-se como exemplo $\forall x\ P(x)$ em que P é um predicado qualquer. É óbvio que esta fórmula jamais pode ser obtida através de ϕ por causa do quantificador universal. Transformemo-la então em $\neg\exists x\ \neg P(x)$. Continuamos com uma fórmula impossível de obter devido ao literal negativo. Introduzindo então mais um quantificador existencial vem $\neg\exists x\ \neg\exists y\ P(y) \wedge y = x$. Esta última versão da fórmula inicial corresponde ao grafo incompleto $\neg\boxed{\neg\boxed{\boxed{\text{P}: y = x}}}$ ou na notação puramente gráfica $\neg\boxed{\cdot\cdot\neg\boxed{\cdots\boxed{\text{P}}}}$. A pergunta coloca-se imediatamente: mas qual é o conceito que introduz a variável x e que devia estar no outro extremo do elo de correferência? Eis a resposta de Sowa.

Definição 4.2.8 (Sowa). Se t é um tipo para um conceito, o *tipo negado* $\neg t$ é definido por uma definição de tipo da forma **tipo** `¬t(x) é [T: *x] ¬[[t:  *x]]`.

A fórmula $\neg\exists x\ \neg P(x)$ corresponde pois ao grafo $\neg\boxed{\boxed{\neg\text{P}}}$ que deve ser expandido em $g = \neg\boxed{\boxed{\text{T}}\cdot\cdot\neg\boxed{\cdots\boxed{\text{P}}}}$. Mas esta definição não é consistente com a de ϕ porque dá um significado especial ao tipo universal que não é correspondido na tradução para fórmulas lógicas nem no processo de inferência. De facto, a única maneira de $\phi(g) = \neg(\exists x\ \ T(x) \wedge \neg(\exists y\ \ P(y) \wedge y = x))$ corresponder a $\forall x\ P(x)$ é assumindo que $T(x)$ é sempre verdade para qualquer x.

Para finalizar a apresentação do esquema de tradução na TEC original, gostaria de chamar a atenção para outro pequeno problema na definição de ϕ. Repare-se que não existe nenhuma ordem pré-estabelecida entre os grafos que se encontrem num mesmo contexto. Assim, podem-se obter fórmulas diferentes consoante a ordem pela qual se escrevem as conjunções. No entanto, se não houver linhas de identidade não há problema porque embora diferentes, as fórmulas obtidas são equivalentes. O caso muda de figura com a presença de elos de correferência. A presença de variáveis livres implica uma certa ordem que, se não for obedecida, leva à geração de uma fórmula não equivalente à que se pretende. Vejamos um exemplo.

Exemplo 56. O grafo $\neg\boxed{\boxed{\text{GATO}}\cdots}\cdots\boxed{\text{CÃO}}$ é suposto representar a frase "existe um cão que não é gato". Fazendo a transformação dada na Hipótese 4.2.6 na página anterior obtém-se $\neg\boxed{\boxed{\text{GATO}: y = x}}\ \boxed{\text{CÃO}: x}$. Como a Hipótese 4.2.3 na página 93 não diz qual a ordem pela qual devem ser traduzidos os dois grafos que se encontram no contexto exterior, pode-se obter $\neg(\exists y\ \ Gato(y) \wedge y = x) \wedge \exists x\ \ C\tilde{a}o(x)$ ou então $\exists x\ \ C\tilde{a}o(x) \wedge \neg(\exists y\ \ Gato(y) \wedge y = x)$. Estas duas fórmulas não são equivalentes porque há uma ocorrência livre de x na primeira fórmula. Assim, apenas a segunda corresponde ao significado pretendido.

6.1.2 A Nova Definição

A nova definição de ϕ que apresentarei no resto desta secção é basicamente igual à de Sowa. As diferenças devem-se à resolução dos problemas já mencionados e ao facto de os marcadores individuais estarem organizados numa hierarquia. Quanto ao problema de nem todas as fórmulas de primeira ordem poderem ser representadas, a solução é a que foi indicada: basta interpretar os tipos universais e absurdos como verdade e falsidade, respectivamente. Em relação aos contextos vazios, basta traduzi-los por tautologias. Finalmente, pode-se resolver o problema do mesmo grafo poder originar fórmulas incompatíveis garantindo que todos os quantificadores existenciais são colocados antes da conjunção das sub-fórmulas respeitantes aos grafos do mesmo contexto.

A diferença mais significativa deve-se no entanto ao facto de haver uma ordem parcial sobre os referentes, ao contrário do que sucede na TEC original. Por essa razão, um elo de correferência entre referentes não representa apenas a igualdade. Como já foi dito na Secção 4.4 na página 58, um grafo com dependências (o que inclui os elos de correferência) representa todos as suas instâncias que satisfazem as equações associadas. No caso dos referentes não formarem uma hierarquia, um grafo é a sua própria instância e portanto a ordem parcial entre referentes resume-se à igualdade. Mas na presente abordagem o teste de identidade tem que ser substituído por um teste de subsunção.

Consideremos por exemplo o grafo $\boxed{\text{FORMA: }\texttt{\#losango}}\cdots\boxed{\text{FORMA: }\texttt{\#rectângulo}}$. Na abordagem de Sowa todos os tipos são de primeira ordem e os referentes representam indivíduos incomparáveis entre si. Deste modo, o grafo afirma que “um losango é a mesma forma que um rectângulo”. Aplicando a definição original de ϕ a este grafo composto, faz-se uma cópia mútua dos referentes, porque ambos os conceitos são dominantes, e obtém-se

$$Forma(losango) \wedge losango = rectângulo \wedge Forma(rectângulo) \wedge rectângulo = losango$$

que equivale de facto ao significado intuitivo. No entanto, neste texto `RECTÂNGULO` e `LOSANGO` são tipos de primeira ordem e `FORMA` é um tipo de segunda ordem (Exemplo 1 na página 31). Assim, a leitura do grafo será “existe uma forma que é simultaneamente um losango e um rectângulo”, pelo que a tradução deve ser

$$\exists x\ Forma(x) \wedge x \sqsubseteq losango \wedge Forma(x) \wedge x \sqsubseteq rectângulo$$

em que $\sqsubseteq$ é um predicado especial (escrito sob a forma de operador) denotando a ordem parcial entre os tipos. Note-se que a primeira fórmula é falsa, mas a segunda é verdadeira porque x pode ser substituído por `QUADRADO` (Exemplo 29 na página 57).

O resto desta secção limita-se a definir primeiro uma linguagem de primeira ordem L e depois a função ϕ que transforma grafos conceptuais em fórmulas fechadas de L. Embora a tradução siga no geral a abordagem de Sowa com as diferenças mencionadas nos parágrafos anteriores, as fórmulas obtidas terão um aspecto bastante diferente. Vejamos porquê. Os tipos de relação e os tipos de conceito não-relacionais podem aparecer tanto à esquerda como à direita de ‘:’ num conceito. Por outras palavras, a maior parte dos tipos pode ser usada como marcadores, o que significa que nas fórmulas lógicas podem dar origem a predicados ou a constantes. Por exemplo $\boxed{\text{TIGRE: }\texttt{\#Hobbes}}$

traduz-se em $Tigre(hobbes)$ enquanto $\boxed{\text{ESPÉCIE: \#tigre}}$ equivale a $Espécie(tigre)$, em que $Tigre$ e $tigre$ são símbolos diferentes de L. Com o objectivo de manter a linguagem tão simples quanto possível, escrever-se-á $Holds(tigre, hobbes)$ em vez de $Tigre(hobbes)$. Assim, cada tipo gera apenas um único símbolo em L e não dois. O "meta-predicado" $Holds$ (parecido com o usado na linguagem KIF [Genesereth e Fikes, 1992]) pode ser igualmente aplicado a relações. Assim por exemplo, $Agnt(x, y)$ passa a ser escrito na forma $Holds(agnt, x, y)$. No entanto, como cada predicado tem uma aridade fixa, não existe um único $Holds$ mas sim um conjunto $\{Holds_i | i > 0\}$. Uma relação n-ária será traduzida numa fórmula atómica com o predicado $Holds_n$, pelo que $Holds_n$ tem $n + 1$ argumentos sendo o primeiro a constante correspondente à relação.

A utilização de $Holds$, para além de tornar L mais compacta, torna o predicado especial $\sqsubseteq$ desnecessário, desde que se tenha a igualdade. A fórmula atómica $m \sqsubseteq m'$ pode ser reescrita em $\forall x\ Holds(m, x) \rightarrow Holds(m', x)$ se m e m' forem tipos. Caso contrário, ou seja, se m e m' forem indivíduos, $m \sqsubseteq m'$ é simplesmente o mesmo que $m = m'$. Por sua vez, a igualdade também serve para representar os tipos universais e absurdos: $\top_c$ e $\top_r$ são traduzidos para $x = x$, em que x é uma variável qualquer, e $\perp_c$ e $\perp_r$ correspondem pois ambos a $\neg(x = x)$. Torna-se assim desnecessário ter um predicado especial para representar 'verdade' ou 'falso'[2].

Estamos pois em condições de passar à definição formal de L. Como qualquer linguagem de primeira ordem, L contém símbolos lógicos que compõem a parte fixa da linguagem, e símbolos não lógicos que são dependentes da aplicação. Os símbolos lógicos dividem-se em conectivos, quantificadores, variáveis, e símbolos auxiliares de pontuação. Os conectivos necessários são de novo a negação e a conjunção, pelo que basta ter o quantificador existencial. Quanto aos símbolos não lógicos, dividem-se em constantes, funções e predicados. Como acontecia na teoria original, não há funções. Quanto às constantes, a cada marcador individual tem que corresponder uma constante distinta de todas as outras. Em relação aos predicados, é preciso antes de mais a igualdade, que representarei com o símbolo $\doteq$ para a diferenciar da igualdade usada nas definições e hipóteses deste texto. Em segundo lugar é preciso um conjunto de predicados $Holds_i$ para as diferentes aridades dos tipos de relação (um tipo de conceito pode ser encarado como uma relação unária). Finalmente, é necessário um predicado distinto para cada tipo de conceito relacional, pois esses são os únicos tipos que não podem ser usados como marcadores. Como tal, não podem ser representados por constantes de L donde não podem ser traduzidos usando $Holds_i$.

Hipótese 65. A *linguagem de primeira ordem* L associada a um dado cânone (com hierarquias de tipos $\mathcal{T_C}$ e $\mathcal{T_R}$ e conjunto de marcadores $\mathcal{M}$) é composta de:

- um conjunto não vazio de variáveis;
- um conjunto de constantes $\{c_m | m \in \mathcal{M} - \{*, \bar{*}\}\}$ tal que $m \neq m' \Rightarrow c_m \neq c_{m'}$;
- um conjunto de predicados $\{\doteq\} \cup \{Holds_i | i > 0\} \cup \{p_t | t \in T^{rc}_{\leq or}\}$ tal que $t \neq t' \Rightarrow p_t \neq p_{t'}$;
- os símbolos de pontuação $(), ;$

[2]Ter uma constante em vez de um predicado não basta, como mostrou a discussão da Definição 4.2.8 na página 96.

- os conectivos $\neg$ e $\wedge$;
- o quantificador $\exists$.

A definição da função de tradução ϕ é fácil de entender nas suas linhas gerais pois correspondem às da definição original. No entanto os detalhes técnicos são um pouco complexos pelo que requerem uma explicação alargada. Pela mesma ordem em que serão dadas na definição formal, eis as regras para a construção da fórmula de primeira ordem correspondente a um dado grafo.

1. O primeiro passo consiste em tratar os contextos vazios. Por convenção, eles representam 'verdade', tal como o conceito $\boxed{\top_c\colon *}$. Assim, introduzindo um conceito desses em cada contexto vazio, o esquema habitual de tradução encarregar-se-á do resto.

2. De seguida associa-se uma variável única, chamada identificador, a *cada* vértice conceptual. Ou seja, ao contrário do que fazia Sowa, até os conceitos individuais vão ter a sua variável, porque como se viu anteriormente a existência de um elo de correferência obriga à introdução de uma variável para representar a instância comum dos marcadores que satisfaz a equação.

3. A tradução de um conceito tem que ter em conta se ele domina (ou é dominado por) outros conceitos. Assim, é preciso saber para cada conceito se ele faz parte de linhas de identidade ou não.

4. Os identificadores dos conceitos dominantes são copiados para os conceitos dominados.

5. Cada conceito está quantificado existencialmente.

6. Um conceito $c = \boxed{t\colon m}$ com identificador x corresponde basicamente à fórmula atómica $Holds_1(c_t, x)$. No entanto, se $t = \top_c$ ou $t = \bot_c$ então a constante c_t não está definida. Nesses casos a fórmula fica simplesmente $\doteq (x, x)$ ou $\neg \doteq (x, x)$, respectivamente. Quando $m = \bar{*}$ o conceito também é falso, pelo que é preciso de novo incluir $\neg \doteq (x, x)$. Para além deste núcleo, a tradução de c tem que conter a fórmula atómica $x \doteq y$ para cada identificador y de um vértice conceptual que domine c. Por último, deve-se ter em conta m se ele for um marcador individual. Como já foi dito, há dois casos possíveis. Se o conceito não fizer parte de nenhum elo de correferência ou se m denotar um indivíduo, então basta ter $x \doteq m$. Se m denotar um tipo e fizer parte de um elo de correferência, então $x \sqsubseteq m$, o que equivale a dizer $\forall y\ Holds_1(x, y) \rightarrow Holds_1(c_m, y)$. Segundo o vocabulário adoptado, a fórmula resultante deve ser $\neg\exists y\ Holds_1(x, y) \wedge \neg Holds_1(c_m, y)$.

7. Um vértice relacional de tipo t e ligado a n vértices conceptuais com identificadores $x_1, \ldots, x_n$ é traduzido para $Holds_n(c_t, x_1, \ldots, x_n)$. De novo, se $t = \top_r$ ou $t = \bot_r$ a constante c_t não está definida. Mas como $\top_r$ denota a relação universal (isto é, a que contém todos as combinações de indivíduos possíveis) e $\bot_r$ denota a relação absurda (isto é, a que não contém nenhum tuplo de indivíduos), as traduções destes casos podem ser simplesmente $\doteq (x_1, x_1)$ e $\neg \doteq (x_1, x_1)$, respectivamente.

8. A fórmula de primeira ordem respeitante a um grafo raso é obtida como na abordagem de Sowa, isto é, após a quantificação existencial de todos os vértices conceptuais segue-se a conjunção das sub-fórmulas obtidas nos passos anteriores.

9. A tradução de um contexto também é feita como na Hipótese 4.2.3 na página 93, mas com o cuidado de colocar todos os quantificadores existenciais no início da fórmula.

10. Se o contexto for negativo, nega-se a fórmula obtida pela regra anterior.

No resto do texto optei por escrever o predicado $\doteq$ como operador infixo para maior legibilidade. Assim, $\neg x \doteq x$ representa $\neg \doteq (x, x)$ e não $\doteq (\neg x, x)$. De facto, esta última não é uma fórmula lógica bem formada porque os conectivos apenas podem ser aplicado a fórmulas (atómicas ou compostas) e não directamente a variáveis ou constantes. Eis pois a definição formal das regras acima descritas[3].

Hipótese 66. Seja L a linguagem de primeira ordem associada a um dado cânone. As funções ϕ, ϕ_p, ϕ_c devolvem para cada grafo conceptual uma sequência de símbolos de L. A sequência $\phi(g)$ é a *fórmula de primeira ordem* associada ao grafo g, composta pelo prefixo de quantificação $\phi_p(g)$ e pelo corpo $\phi_c(g)$. A tradução é feita aplicando as regras que se seguem. As funções auxiliares id, li e dom devolvem respectivamente para cada vértice conceptual c uma variável que identifica univocamente c, um booleano que indica se c faz parte de alguma linha de identidade, e o conjunto de identificadores dos vértices conceptuais que dominam c. Quando necessário, o operador $\odot$ representa explicitamente a operação de concatenação de símbolos de L.

1. Em cada contexto vazio de g insere-se um conceito $\boxed{\top_c\texttt{: *}}$.

2. Para cada vértice conceptual c define-se $id(c) = x$, em que x é uma variável de L não usada anteriormente.

3. Para cada vértice conceptual c define-se $li(c) = \texttt{verdade}$ se c fizer parte de alguma linha de identidade, caso contrário $li(c) = \texttt{falso}$.

4. Para cada vértice conceptual c define-se $dom(c) = \{id(c')|c'\text{domina } c\}$.

5. Para cada vértice conceptual c, $\phi_p(c) = \exists id(c)$.

6. Para cada vértice conceptual c com $tp(c) = t$ e $rf(c) = m$, a fórmula $\phi_c(c)$ é obtida pela conjunção de todas as sub-fórmulas aplicáveis das seguidamente indicadas:

   - $Holds_1(c_t, id(c))$;
   - $\neg id(c) \doteq id(c)$ se $t = \bot_c$ ou $m = \bar{*}$;
   - $id(c) \doteq id(c)$ se $t = \top_c$;
   - $p_t(id(c))$ se $t \in T^{rc}_{\leq or}$;
   - $\bigwedge_{x \in dom(c)} id(c) \doteq x$;
   - $c_m \doteq id(c)$ se $m \in T^{nc}_0$ ou $li(c) = \texttt{falso}$;

[3] Relembro que $tp(c)$ é a abreviatura de $tipo(etiqueta(c))$ e $rf(c)$ a de $referente(etiqueta(c))$.

- $(\neg\exists x\ \mathit{Holds}_1(id(c), x) \wedge \neg\mathit{Holds}_1(c_m, x))$ em que $x \neq id(c)$ se $m \notin T_0^{nc}$ e $li(c) = \texttt{verdade}$.

7. Seja r um vértice relacional com $tp(r) = t$ e ligado aos vértices conceptuais $c_1, \ldots, c_n$. Se $t = \top_r$ ou $t = \bot_r$ então $\phi_c(r) = id(c_1) \doteq id(c_1)$ ou $\phi_c(r) = \neg id(c_1) \doteq id(c_1)$, respectivamente. Caso contrário $\phi_c(r) = \mathit{Holds}_n(c_t, id(c_1), \ldots, id(c_n))$.

8. Se g for um grafo conceptual raso com vértices conceptuais V_C e vértices relacionais V_R, então $\phi(g) = \phi_p(g)\phi_c(g)$ em que

$$\phi_p(g) = \bigodot_{c \in V_C} \phi_p(c) \qquad\qquad \phi_c(g) = \bigwedge_{c \in V_C} \phi_c(c) \wedge \bigwedge_{r \in V_R} \phi_c(r)$$

9. Se p for uma proposição contendo o conjunto de grafos G então $\phi_p(p)$ é a sequência vazia e $\phi(p) = \phi_c(p) = (\bigodot_{g \in G} \phi_p(g) \bigwedge_{g \in G} \phi_c(g))$.

10. Se c for um contexto formado pela negação da proposição p, então $\phi_p(c)$ é a sequência vazia e $\phi(c) = \phi_c(c) = \neg\phi(p)$.

Seguem-se vários exemplos, sendo os primeiros meras repetições de grafos apresentados anteriormente. Esses exemplos não só mostram as diferenças entre a antiga e a nova definição de ϕ como também ilustram o tratamento de alguns casos que antes eram problemáticos. Os restantes exemplos serão úteis no resto do capítulo. Exemplos adicionais podem ser encontrados na Tabela 6.2 na página 115. Para simplificar a escrita, a constante c_m associada ao marcador m terá o mesmo nome que m e será grafada em minúsculas, enquanto os predicados p_t, embora também tenham o mesmo nome que t, serão escritos com maiúscula inicial.

Exemplo 57. O grafo $\boxed{\text{GATO: \#Garfield}} \leftarrow (\text{AGNT}) \leftarrow \boxed{\text{PERSEGUIR}} \rightarrow (\text{OBJ}) \rightarrow \boxed{\text{CÃO}}$ do Exemplo 52 na página 92 traduz-se em

$$\begin{aligned}\exists x \exists y \exists z\quad & \mathit{Holds}_1(\mathit{gato}, x) \wedge x \doteq \mathit{garfield} \wedge \mathit{Holds}_1(\mathit{perseguir}, y) \wedge \mathit{Holds}_1(\mathit{cão}, z) \wedge \\ & \mathit{Holds}_2(\mathit{agnt}, y, x) \wedge \mathit{Holds}_2(\mathit{obj}, y, z)\end{aligned}$$

A tradução dos grafos do Exemplo 53 na página 93 e do Exemplo 54 na página 94 mantém-se basicamente igual (excepto o uso do predicado Holds_1) porque esses grafos não têm elos de correferência ou, quando os têm, eles não envolvem marcadores individuais. O exemplo seguinte mostra que a nova definição de ϕ já permite representar a fórmula $\forall x\ P(x)$ da maneira intuitiva que se espera. Nomeadamente, a afirmação "para todo o x, $P(x)$ é verdade" corresponde a "se x for uma entidade qualquer, então $P(x)$".

Exemplo 58. Seja P um tipo de conceito relacional qualquer[4]. Então

$$\phi(\neg\boxed{\boxed{\top_c}\ \neg\boxed{\cdots\boxed{\text{P}}}}) = \neg(\exists x\ x \doteq x \wedge \neg(\exists y\ P(y) \wedge y \doteq x))$$

Ora isto não é mais do que $\forall x\ x \doteq x \rightarrow \exists y\ \mathit{Holds}_1(p, y) \wedge y \doteq x$. Devido às propriedades da igualdade, $x \doteq x$ é sempre verdade e $\mathit{Holds}_1(p, y) \wedge y \doteq x$ corresponde a $\mathit{Holds}_1(p, x)$ pelo que se obtém $\forall x\ \mathit{Holds}_1(p, x)$ como se esperava.

[4] Se P for não-relacional basta substituir $P(y)$ por $\mathit{Holds}_1(p, y)$ no que se segue.

Exemplo 59. Devido à formulação da regra 9, o grafo $\neg\boxed{\boxed{\text{GATO}}\cdots}\cdots\boxed{\text{CÃO}}$ do Exemplo 56 na página 96 já é traduzido correctamente para

$$\exists x\ \neg(\exists y\ Holds_1(gato, y) \wedge y \doteq x) \wedge Holds_1(cão, x)$$

Exemplo 60. O grafo $\boxed{\text{FORMA: \texttt{\#rectângulo}}}\cdots\boxed{\text{FORMA: \texttt{\#losango}}}$ tem um elo de correferência entre tipos, pelo que a sua tradução será

$$\begin{aligned}\exists x \exists y \quad & Holds_1(forma, x) \wedge x \doteq y \wedge (\neg\exists z\ Holds_1(x, z) \wedge \neg Holds_1(rectângulo, z)) \wedge \\ & Holds_1(forma, y) \wedge y \doteq x \wedge (\neg\exists z\ Holds_1(y, z) \wedge \neg Holds_1(losango, z))\end{aligned}$$

Exemplo 61. A linha de identidade em $\boxed{\text{PESSOA: Rosalie}}\cdot\cdot\neg\boxed{\cdots\boxed{\text{PESSOA: Rosann}}}$ liga dois indivíduos. Neste caso, a aplicação da regra 6 resulta em

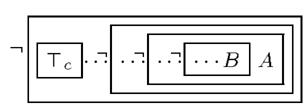
$$\exists x\ Holds_1(person, x) \wedge x \doteq rosalie \wedge \neg(\exists y\ Holds_1(person, y) \wedge y \doteq rosann \wedge y \doteq x)$$

Exemplo 62. A fórmula $\forall x\ x \doteq x$, ou equivalentemente $\neg\exists x\ \neg x \doteq x$, é representada pelo grafo $\neg\boxed{\boxed{\bot_c}}$.

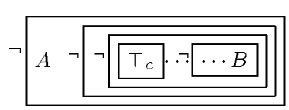
Exemplo 63. Seja A uma fórmula da lógica de primeira ordem em que a variável x não ocorre livre. Então a fórmula $\forall x\ A$ é representada pelo grafo $\neg\boxed{\boxed{\top_c}\ \neg\boxed{A}}$ em que a ausência de elos de correferência mostra que a variável introduzida pelo conceito universal não está livre no grafo A. De facto, pela aplicação da definição de ϕ, obtém-se $\neg(\exists x\ x \doteq x \wedge \neg A)$ ou seja $\forall x\ x \doteq x \rightarrow A$ o que vem dar ao mesmo que $\forall x\ A$.

O Exemplo 58 na página anterior e o Exemplo 63 ilustram de novo que os elos de correferência mostram onde existem variáveis livres e onde é que elas são introduzidas. O exemplo seguinte combina implicações, quantificação universal e variáveis livres e ligadas.

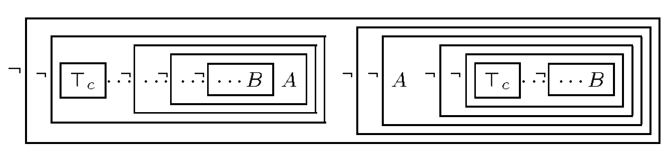
Exemplo 64. Sejam A e B duas fórmulas tais que x ocorre livre em B mas não em A. Os grafos que representam essas duas fórmulas são igualmente designados por A e B. Para enfatizar o facto de B conter um conceito com um elo de correferência escreverei $\cdots B$. Então a fórmula $A \rightarrow B$ é representada pelo grafo $\neg\boxed{A\ \neg\boxed{\cdots B}}$ que não está completo porque o elo de correferência tem uma "ponta solta". Como a quantificação universal corresponde a uma implicação, a fórmula $\forall x\ (A \rightarrow B)$ é a tradução de



Visto x não ser livre em A, não há elo de correferência entre o conceito universal e o grafo A. De modo semelhante obtém-se para $A \rightarrow \forall x\ B$ o grafo



Então o grafo para a fórmula $(\forall x\ (A \rightarrow B)) \rightarrow (A \rightarrow \forall x\ B)$ é



6.2 Interpretação

Após a definição de uma linguagem e de como os grafos conceptuais são representados nessa linguagem, o passo seguinte é definir uma interpretação. Basicamente, trata-se de indicar um domínio e de como as constantes e predicados da linguagem são interpretados. Uma constante corresponde a um elemento do domínio, enquanto um predicado denota um conjunto de tuplos de elementos. Um predicado unário será pois um subconjunto do domínio, um predicado binário um conjunto de pares ordenados, e assim por diante.

Mas enquanto a definição normal de interpretação é bastante flexível, no caso dos grafos conceptuais há bastantes restrições que devem ser satisfeitas. Até ao momento, de toda a informação contida num cânone apenas foi usada a que diz respeito aos tipos e marcadores existentes. É com base neles que se define o vocabulário da linguagem de primeira ordem. Mas os tipos e marcadores não são apenas nomes de entidades arbitrárias; há uma relação entre elas, a qual é dada pelas hierarquias de tipos e marcadores. Em lógica de primeira ordem, essa ordem taxonómica pode ser representada por implicações quantificadas universalmente, como já foram usadas para tratar os elos de correferência entre marcadores individuais.

Hipótese 67. Seja C um cânone com hierarquias de tipos $\mathcal{T_C}$ e $\mathcal{T_R}$ e seja L uma linguagem de primeira ordem associada a C. Então os *axiomas próprios* associados a C e L são

- para todos $t, t' \in T_i^{nc}$ com $t < t'$: $\neg\exists x\ Holds_1(c_t, x) \wedge \neg Holds_1(c_{t'}, x)$
- para todos $t, t' \in T_i^{rc}$ com $t < t'$: $\neg\exists x\ p_t(x) \wedge \neg p_{t'}(x)$
- para todos $t, t' \in T_{\langle i \rangle}^{r}$ com $t < t'$:
  $\neg\exists x_1, \ldots, x_i\ Holds_i(c_t, x_1, \ldots, x_i) \wedge \neg Holds_i(c_{t'}, x_1, \ldots, x_i)$

Qualquer interpretação para L terá pois que satisfazer estes axiomas. No entanto só isto não basta. Os tipos, além de organizados por inclusão numa ordem parcial, encontram-se igualmente classificados segundo a sua ordem e género. É por exemplo impossível um tipo de segunda ordem denotar indivíduos de terceira ordem, ou um tipo não-relacional denotar relações. No fundo, estamos de novo perante os critérios de boa-formação dos conceitos e relações. Um conceito $\boxed{t\colon m}$, e portanto a fórmula atómica $Holds_1(c_t, c_m)$, representa a afirmação "m é instância de t". Por isso é que quando $t = \top_c$ a afirmação é necessariamente verdadeira e quando $t = \bot_c$ ou $m = \bar{*}$ a afirmação torna-se falsa. Do mesmo modo, a relação representada pela fórmula $Holds_i(c_t, x_1, \ldots, x_i)$ corresponde à afirmação "o tuplo $\langle x_1, \ldots, x_i \rangle$ é um elemento da relação denotada por t". Assim torna-se óbvio que para estas afirmações serem verdadeiras ou falsas têm que estar bem-formadas.

A maneira mais fácil de garantir que uma interpretação inclui os critérios de boa-formação dados no Capítulo 3 na página 39 é subdividir o domínio da mesma forma que os tipos e marcadores e depois repetir as condições da Proposição 20 na página 43 e da Proposição 30 na página 46 para os predicados $Holds_i$. Vejamos com mais algum detalhe como obter uma interpretação apropriada para uma base de conhecimentos de grafos conceptuais.

1. O domínio de interpretação tem que ser dividido em sub-domínios que correspondam aos vários subconjuntos de tipos. Esses sub-domínios serão obviamente disjuntos, porque cada tipo só pode ter uma única ordem e um único género ou aridade.

2. A constante associada a um tipo de conceito não-relacional de ordem i será interpretada como um elemento do sub-domínio correspondente.

3. O mesmo acontece para as constantes associadas a tipos de relação de ordem i e aridade j. Como os sub-domínios são disjuntos, garante-se assim que um tipo de conceito não é interpretado como uma relação, que uma relação de segunda ordem não é interpretada como uma relação de primeira ordem, que uma relação binária não é considerada ternária, etc.

4. Quanto aos tipos de conceito relacionais, os símbolos de L que lhes estão associados representam predicados e não constantes, pelo que a denotação terá que ser um subconjunto do domínio e não apenas um elemento. Um tipo relacional de ordem i denota um conjunto de relações de aridade arbitrária mas de ordem não superior a i. Portanto, a interpretação do predicado tem que ser um subconjunto da união dos sub-domínios respeitantes a essas relações.

5. Claro que a interpretação dos predicados para os tipos relacionais tem que obedecer aos axiomas próprios acima apresentados.

6. Continuando com a denotação dos predicados, é a vez dos vários *Holds*. Como se viu na Hipótese 66 na página 100, o predicado $Holds_i$ tem aridade $i+1$ porque o primeiro elemento representa o tipo da relação ou conceito e os restantes i elementos indicam os argumentos dessa relação ou conceito. Assim sendo, a interpretação de $Holds_i$ tem que ser um subconjunto do produto cartesiano do domínio consigo próprio $i+1$ vezes.

7. Tal como aconteceu no ponto 4 para os predicados associados aos conceitos relacionais, os predicados $Holds_i$ têm que satisfazer as condições de boa-formação. O caso de $Holds_1$ é diferente dos restantes, pois representa tanto conceitos não-relacionais como relações unárias. Há pois duas possibilidades para $Holds_1(c_t, c_m)$ estar bem-formado. Na primeira, t é um tipo não-relacional de ordem n e m é de ordem $n-1$. Na segunda hipótese, t é um tipo de relação unário de ordem j e m é um tipo de conceito de qualquer ordem ou então m é um tipo de relação de aridade arbitrária mas de ordem inferior a j. A denotação de $Holds_1$ é a união dos produtos cartesianos dos sub-domínios correspondentes a estas duas possibilidades.

8. A denotação de $Holds_i$ é a generalização do caso de $Holds_1$ para as relações unárias. O primeiro argumento de $Holds_i$ tem que representar uma relação i-ária de ordem j e cada um dos restantes i argumentos tem que denotar um tipo de conceito arbitrário ou uma relação de aridade qualquer mas de ordem inferior a j.

9. O denotação de $Holds_1$ tem que obviamente obedecer aos axiomas próprios para os tipos de conceito não-relacionais.

10. O mesmo acontece para os predicados $Holds_n$ no que respeita à ordem parcial sobre as relações n-árias.

11. A denotação do predicado que falta, a igualdade, é trivial.

Antes de apresentar a definição formal convém relembrar certas convenções notacionais. Se S for um conjunto qualquer, S^i é uma forma abreviada de escrever o produto cartesiano $S \times \ldots \times S$, com i ocorrências de S. Se S_i for uma família de conjuntos indexados dentro de certos limites, então $S_{\leq j} = \bigcup_{i \leq j} S_i$. De forma análoga, se $S_{\langle i,j \rangle}$ for uma familia indexada por pares ordenados, então $S_{\langle \leq k, >m \rangle}$ não é mais do que a abreviatura de $\bigcup_{i \leq k} \bigcup_{j > m} S_{\langle i,j \rangle}$.

Hipótese 68. Uma *interpretação* de uma linguagem L, associada a um dado cânone, é um par ordenado $\langle D, \delta \rangle$ em que D é um conjunto não vazio chamado *domínio* ou *universo* de L e δ é uma função total que associa a cada constante e predicado de L a sua *denotação*. A interpretação deve obedecer às seguintes restrições:

1. D é decomponível em subconjuntos $D^c_0, D^c_1, \ldots, D^r_{\langle 1,1 \rangle}, D^r_{\langle 1,2 \rangle}, \ldots, D^r_{\langle 2,1 \rangle}, \ldots$ mutuamente disjuntos
2. $\forall t \in T^{nc}_i \;\; \delta(c_t) \in D^c_i$
3. $\forall t \in T^r_i \cap T^r_{\langle j \rangle} \;\; \delta(c_t) \in D^r_{\langle i,j \rangle}$
4. $\forall t \in T^{rc}_i \;\; \delta(p_t) \subseteq D^r_{\langle \leq i, >0 \rangle}$
5. $\forall t, t' \in T^{rc}_{\leq or} \;\; t \leq t' \Rightarrow \delta(p_t) \subseteq \delta(p_{t'})$
6. $\forall i > 0 \;\; \delta(Holds_i) \subseteq D^{i+1}$
7. $\delta(Holds_1) \subseteq \bigcup_{i>0} D^c_i \times D^c_{i-1} \cup \bigcup_{j>0} D^r_{\langle j,1 \rangle} \times (D^c_{\geq 0} \cup D^r_{\langle <j, >0 \rangle})$
8. $\forall i > 1 \;\; \delta(Holds_i) \subseteq \bigcup_{j>0} D^r_{\langle j,i \rangle} \times (D^c_{\geq 0} \cup D^r_{\langle <j, >0 \rangle})^i$
9. $\forall t, t' \in T^{nc}_{\leq on} \;\; \forall d \in D \;\; t \leq t' \wedge Holds_1(c_t, d) \Rightarrow Holds_1(c_{t'}, d)$
10. $\forall i > 0 \;\; \forall t, t' \in T^r_{\langle i \rangle} \;\; \forall d_1, \ldots, d_i \in D$
    $t \leq t' \wedge Holds_i(c_t, d_1, \ldots, d_i) \Rightarrow Holds_1(c_{t'}, d_1, \ldots, d_i)$
11. $\delta(\doteq) = \{\langle d, d \rangle | d \in D\}$

Exemplo 65. Seja o domínio D o próprio conjunto de constantes de L, ou seja, para cada marcador m tem-se $\delta(c_m) = c_m$. Eis a interpretação de alguns tipos e marcadores que ocorrem em exemplos anteriores, sobretudo no Capítulo 2 na página 27 e no Capítulo 3 na página 39.

- $D^c_0 = \{garfield, odie, joão, \ldots\}$
- $D^c_1 = \{gato, felino, animal, vermelho, quadrado, \ldots\}$
- $D^c_2 = \{tipo, espécie, família, reino, forma, cor, \ldots\}$
- $D^c_3 = \{tipo', categoria, característica \ldots\}$
- $D^r_{\langle 1,2 \rangle} = \{agnt, obj, classe, carac, \ldots\}$
- $D^r_{\langle 1,3 \rangle} = \{entre, \ldots\}$

- $D^r_{\langle or+1,2\rangle} = \{elo, inversa\text{-}de, \ldots\}$
- $\delta(Relação) = D^r_{\langle \leq or, >0\rangle} = \{agnt, obj, classe, carac, entre, \ldots\}$
- $\delta(Binária) = D^r_{\langle \leq or, 2\rangle} \subset \delta(Relação)$, em particular $entre \notin \delta(Binária)$
- $\delta(Holds_1)$ inclui os seguintes pares ordenados
  - $\forall d \in D^c_1 \;\; \langle tipo, d\rangle$
  - $\forall d \in D^c_2 \;\; \langle tipo', d\rangle$
  - $\{\langle categoria, espécie\rangle, \langle categoria, família\rangle, \langle categoria, reino\rangle\} \subset D^c_3 \times D^c_2$
  - $\{\langle característica, forma\rangle, \langle característica, cor\rangle\} \subset D^c_3 \times D^c_2$

  e obedece entre outras às seguintes restrições
  - $\forall d \in D^c_2 \;\; Holds_1(categoria, d) \Rightarrow Holds_1(tipo', d)$
  - $\forall d \in D^c_2 \;\; Holds_1(categoria, d) \Rightarrow Holds_1(característica, d)$
- $\delta(Holds_2)$ contém os seguintes triplos
  - $\{\langle classe, joão, pessoa\rangle, \langle classe, joão, animal\rangle, \langle classe, garfield, gato\rangle, \langle classe, garfield, felino\rangle\} \subset D^r_{\langle 1,2\rangle} \times D^c_{\geq 0} \times D^c_{\geq 0}$
  - $\{\langle inversa\text{-}de, antepassado, descendente\rangle\} \subset D^r_{\langle or+1,2\rangle} \times D^r_{\langle \leq or, >0\rangle} \times D^r_{\langle \leq or, >0\rangle}$

  e obedece entre outras às seguintes restrições
  - $\forall d_1, d_2 \in D \;\; Holds_2(classe, d_1, d_2) \Rightarrow Holds_2(elo, d_1, d_2)$
  - $\forall d_1, d_2 \in D \;\; Holds_2(carac, d_1, d_2) \Rightarrow Holds_2(elo, d_1, d_2)$

Com vimos, uma interpretação para grafos conceptuais tem que incluir a semântica dos critérios de boa-formação e às hierarquias de tipos. No entanto, não se pode forçar a interpretação a estabelecer relações que não existem entre as constantes da linguagem. Por exemplo, à primeira vista a restrição $t_1 < t_2 \wedge Holds_1(c_{t_3}, c_{t_1}) \Rightarrow Holds_1(c_{t_3}, c_{t_2})$ pode parecer verdadeira: se um certo tipo t_1 é instância de um certo tipo (de ordem superior) t_3, então todos os supertipos de t_1 também são instâncias de t_3. Infelizmente, nem esta regra nem a complementar (os *subtipos* de t_1 são instâncias de t_3) devem ser assumidas. Sowa [1992] apresenta um contra-exemplo que na notação desta dissertação se escreve do seguinte modo: `GATO` < `FELINO` e $\langle espécie, gato\rangle \in \delta(Holds_1)$ mas $\langle espécie, felino\rangle \notin \delta(Holds_1)$ porque felino é um género (ver Exemplo 13 na página 43). Aplicando a $\langle género, felino\rangle \in \delta(Holds_1)$ a regra complementar dever-se-ia obter $\langle género, gato\rangle \in \delta(Holds_1)$ que também não é verdade.

A lição a tirar deste exemplo é que os tipos de ordem superior servem para reagrupar os tipos da ordem imediatamente inferior segundo quaisquer critérios que pareçam ontologicamente válidos. Assim como os marcadores de T_0^{nc} são mutuamente incomparáveis porque representam indivíduos, os tipos de primeira ordem também são indivíduos em relação aos tipos de segunda ordem, e assim por diante. Ou seja, a hierarquia formada pelos tipos da mesma ordem não faz necessariamente sentido quando vista da ordem acima. Claro que há excepções, como o caso de `QUADRADO` em que os seus supertipos imediatos (`LOSANGO` e `RECTÂNGULO`) são instâncias do mesmo tipo de segunda ordem (`FORMA`). Mas regra geral, um tipo de ordem superior pode ter como instâncias tipos de ordem inferior que não estão relacionados entre si. Só assim se consegue ter a expressividade necessária para fazer um bom aproveitamento dos tipos de ordem superior através da criação de ontologias ricas.

A observação feita nos dois parágrafos anteriores tem consequências práticas nas regras de inferência, o tema da próxima secção.

6.3 Inferência

Munidos de uma linguagem e da sua interpretação podemos passar às regras de inferência. No seu livro original, Sowa apenas tinha regras canónicas de formação que especializavam os grafos a que eram aplicadas (ver Hipótese 3.4.3 na página 76). Desse modo tinha-se uma simples relação entre as regras canónicas de formação e as regras de inferência, nomeadamente se um grafo u for canonicamente derivável de um grafo v então v implica u.

Teorema 3.5.3 (Sowa). *Para quaisquer grafos conceptuais u e v, se $u \leq v$, então $\phi u \Rightarrow \phi v$.*

Mas em [Sowa, 1995] as regras canónicas de formação já podem generalizar grafos (Definição na página 77) pelo que o teorema anterior deixa de ser válido. Como a abordagem deste trabalho é nesse aspecto igual à de Sowa, também não existe nenhuma relação simples entre as regras canónicas de formação e as regras de inferência. Quanto muito poderá dizer-se que cada regra de inferência é a aplicação de uma ou mais regras canónicas de formação, desde que certas condições se verifiquem. Como se verá no resto desta secção, a condição mais usada é o encaixe do grafo a ser manipulado (Definição 4.2.4 na página 54).

6.3.1 O Caso Proposicional

O processo de inferência é o mais simples possível quando os grafos representam proposições e como tal não podem ser decompostos, analisados ou interligados. Deste modo, as regras proposicionais de inferência têm que manipular grafos completos.

Hipótese 4.3.1 (Sowa). Contenha o contexto exterior um conjunto S de grafos conceptuais. Qualquer grafo derivado de S pelas seguintes *regras proposicionais de inferência* é dito *demonstrável* a partir de S.

Remoção Qualquer grafo com encaixe par pode ser apagado.

Inserção Qualquer grafo pode ser inserido num contexto com encaixe ímpar.

Iteração Um cópia de qualquer grafo u pode ser inserida no mesmo contexto em que u ocorre ou num contexto dominado por u.

Deiteração Qualquer grafo cuja ocorrência pudesse ser o resultado de uma iteração pode ser apagado—isto é, se for idêntico a outro grafo no mesmo contexto ou num contexto dominante.

Dupla negação Uma dupla negação pode ser removida ou desenhada à volta de qualquer grafo ou conjunto de grafos em qualquer contexto.

O conjunto vazio de grafos é o único axioma lógico; é escrito como {} ou simplesmente como espaço em branco. Qualquer grafo que é demonstrável a partir de {} por estas regras é um *teorema*.

Nos vários exemplos que se seguem, A e B são grafos conceptuais arbitrários representando respectivamente as proposições a e b. As três primeiras derivações mostram que os grafos dados no Exemplo 53 na página 93 são teoremas.

Exemplo 66. O teorema $a \to (b \to a)$ deriva-se do seguinte modo.

1. Dupla negação: $\neg\boxed{\ \neg\boxed{}}$
2. 2 Inserções: $\neg\boxed{A\ B\ \neg\boxed{}}$
3. Iteração: $\neg\boxed{A\ B\ \neg\boxed{A}}$
4. Dupla negação: $\neg\boxed{A\ \neg\boxed{\neg\boxed{B\ \neg\boxed{A}}}}$

Exemplo 67. Eis a derivação de $(a \to (b \to c)) \to ((a \to b) \to (a \to c))$:

1. Dupla negação: $\neg\boxed{\ \neg\boxed{}}$
2. 3 Inserções: $\neg\boxed{B\ A\ \neg\boxed{C}\ \neg\boxed{}}$
3. 3 Iterações: $\neg\boxed{B\ A\ \neg\boxed{C}\ \neg\boxed{A\ B\ \neg\boxed{C}}}$
4. Dupla negação: $\neg\boxed{\neg\boxed{\ \neg\boxed{B}}\ A\ \neg\boxed{C}\ \neg\boxed{A\ B\ \neg\boxed{C}}}$
5. Iteração: $\neg\boxed{\neg\boxed{A\ \neg\boxed{B}}\ A\ \neg\boxed{C}\ \neg\boxed{A\ B\ \neg\boxed{C}}}$
6. 3 Duplas negações: $\neg\boxed{\neg\boxed{\neg\boxed{\neg\boxed{A\ \neg\boxed{B}}\ \neg\boxed{\neg\boxed{A\ \neg\boxed{C}}}}}\ \neg\boxed{A\ \neg\boxed{\neg\boxed{B\ \neg\boxed{C}}}}}$

Exemplo 68. O teorema $(\neg a \to \neg b) \to (b \to a)$ demonstra-se assim:

1. Dupla negação: $\neg\boxed{\ \neg\boxed{}}$
2. 2 Inserções: $\neg\boxed{A\ B\ \neg\boxed{}}$
3. Iteração: $\neg\boxed{A\ B\ \neg\boxed{A}}$

4. Dupla negação: $\neg\boxed{\neg\boxed{\neg\boxed{A}}\ B\ \neg\boxed{A}}$

5. Iteração: $\neg\boxed{\neg\boxed{\neg\boxed{A}\ B}\ B\ \neg\boxed{A}}$

6. 2 Duplas negações: $\neg\boxed{\neg\boxed{\neg\boxed{A}\ \neg\boxed{\neg\boxed{B}}}\ \neg\boxed{\neg\boxed{B\ \neg\boxed{A}}}}$

Exemplo 69. A regra de *modus ponens* $a \wedge (a \rightarrow b) \vdash b$ pode ser simulada da seguinte maneira.

1. Hipótese: $A\ \neg\boxed{A\ \neg\boxed{B}}$

2. Deiteração: $A\ \neg\boxed{\ \neg\boxed{B}}$

3. Dupla negação: $A\ B$

4. Remoção: B

Exemplo 70. O *teorema da dedução* diz que se Γ for um conjunto (possivelmente vazio) de proposições e se $\Gamma \cup \{a\} \vdash b$ então $\Gamma \vdash a \rightarrow b$. Seja G o conjunto de grafos que representam Γ. Assumindo pois que G são grafos verdadeiros, consegue-se chegar à implicação do seguinte modo:

1. Hipótese: G

2. Dupla negação: $G\ \neg\boxed{\ \neg\boxed{}}$

3. Inserção: $G\ \neg\boxed{A\ \neg\boxed{}}$

4. Iterações: $G\ \neg\boxed{A\ \neg\boxed{G\ A}}$

5. $G, A \vdash B$: $G\ \neg\boxed{A\ \neg\boxed{B}}$

6. Remoção: $\neg\boxed{A\ \neg\boxed{B}}$

Repare-se que o último exemplo tirou partido do facto das regras de inferência serem independentes do nível em que são aplicadas. Assim, embora a hipótese seja que B é derivável a partir de G e A no contexto exterior, pode-se repetir a mesma derivação em qualquer contexto com encaixe par, em particular de nível 2.

Os exemplos acima dados não são "inocentes", pois servem para provar que as regras proposicionais de inferência são completas.

Teorema 69. *As regras proposicionais de inferência e o axioma lógico* $\{\}$ *formam um sistema formal completo e consistente para o cálculo proposicional.*

Demonstração. Hamilton [1988] demonstrou que a regra de inferência de *modus ponens* e os esquemas de axiomas dados no Exemplo 53 na página 93 formam um sistema completo. Como os exemplos acima dados provam, os axiomas são teoremas e o *modus ponens* pode ser simulado usando as regras proposicionais de inferência. Fica assim provada a completude.

Quanto à consistência, basta verificar que cada regra de inferência corresponde a uma dedução válida em lógica proposicional. Para tal é preciso ter em conta duas observações. A primeira é que os os grafos em contextos com encaixe ímpar denotam proposições negadas enquanto os grafos em contextos com encaixe par correspondem a proposições positivas. A segunda observação consiste em notar que as regras de iteração/deiteração só são mesmo necessárias quando complementam as de inserção/remoção, respectivamente; isto é, quando permitem colocar grafos em contextos com encaixe par ou apagá-los de contextos com encaixe ímpar. Posto isto, vejamos quais as derivações correspondentes às diversas regras de inferência.

Remoção Num contexto par, eliminar o grafo A_1 do conjunto de grafos $A_1, A_2, \ldots, A_n$ corresponde a $a_1 \wedge a_2 \wedge \ldots \wedge a_n \vdash a_2 \wedge \ldots \wedge a_n$. Se A_1 for o único grafo, então $a_1 \vdash \mathit{verdade}$.

Inserção Num contexto ímpar, acrescentar o grafo A_1 ao conjunto de grafos $A_2, \ldots, A_n$ corresponde a $\neg(a_2 \wedge \ldots \wedge a_n) \vdash \neg(a_1 \wedge a_2 \wedge \ldots \wedge a_n)$. Se A_1 for o único grafo, então $\mathit{falso} \vdash \neg a_1$.

Iteração Seja c um contexto contendo o grafo A que se pretende iterar para um contexto com encaixe par contido em c. Há dois casos, consoante c tiver profundidade par ou ímpar. As derivações respectivas são $a \vdash a \wedge a$ e $\neg a \vdash \neg(a \wedge \neg a)$.

Deiteração Seja c um contexto contendo o grafo A e um contexto com encaixe ímpar onde se encontra uma cópia de A. A deiteração da cópia de A corresponde à derivação $a \wedge \neg a \vdash a$ se c tiver profundidade par, caso contrário tem-se $\neg(a \wedge a) \vdash \neg a$.

Dupla negação A inserção e remoção de uma dupla negação corresponde a $a \vdash \neg\neg a$ e $\neg\neg a \vdash a$, respectivamente. □

Uma outra maneira de compreender as regras proposicionais de inferência consiste em encarar os grafos nos contextos com encaixe ímpar como antecedentes de uma implicação, sendo os grafos nos contextos com encaixe par os consequentes. Assim, a regra de remoção permite eliminar consequentes, a inserção acrescenta antecedentes, a iteração copia antecedentes para os consequentes e a deiteração elimina consequentes que também sejam antecedentes. Como facilmente se percebe, estas regras mantêm a validade da implicação.

6.3.2 O Caso Geral

Para o cálculo de predicados é preciso manipular as variáveis, os termos e os predicados que compõem as proposições. Em Estruturas Conceptuais isto equivale a dizer que as regras de inferência de primeira ordem devem manipular os tipos, os marcadores, os vértices, os arcos e os elos de correferência que compõem os grafos conceptuais.

Na abordagem de Sowa, só existe uma hierarquia sobre os tipos de conceito, formando os tipos de relação e os marcadores conjuntos rasos. Deste modo, as únicas operações permitidas são: relaxamento e restrição dos tipos de conceito, substituição do marcador genérico por um marcador individual e vice-versa. Quanto aos vértices e arcos, podem-se obviamente apagar relações duplicadas (Definição 3.4.2 na página 76). Quanto aos elos de correferência, as regras são mais complicadas. Para melhor explicá-las irei usar no resto desta secção dois símbolos c_1 e c_2 para representar dois conceitos tais que o contexto de c_1 domina o de c_2. Além disso, os referentes de c_1 e c_2 serão designados por m_1 e m_2 e os seus identificadores por x_1 e x_2 respectivamente. Assim, ligar c_1 com c_2 por um elo de correferência envolve acrescentar a condição $x_1 = x_2$ no contexto de c_2. Ora isso só pode ser feito se c_2 tiver profundidade ímpar. Inversamente, um elo de correferência só pode ser eliminado se c_2 estiver num contexto com encaixe par. A razão é a habitual: pode-se acrescentar novos factos em contextos ímpares e pode-se remove-los de contextos pares.

Hipótese 4.3.5 (Sowa). Contenha o contexto exterior um conjunto S de grafos conceptuais. Qualquer grafo derivado de S pelas seguintes *regras de inferência de primeira ordem* é dito *demonstrável* a partir de S.

Remoção Num contexto com encaixe par, qualquer grafo pode ser apagado, qualquer elo de correferência de um conceito dominante para um conceito com encaixe par pode ser apagado, e qualquer tipo pode ser substituído por um supertipo.

Inserção Num contexto com encaixe ímpar, qualquer grafo pode ser inserido, um elo de correferência pode ser desenhado entre dois conceitos idênticos, e pode-se restringir qualquer conceito.

Iteração Um cópia de qualquer grafo u pode ser inserida no mesmo contexto em que u ocorre ou num contexto dominado por u. Um elo de correferência pode ser desenhado de qualquer conceito de u para o conceito correspondente na cópia de u. Se os conceitos a e b num certo contexto c estiverem ambos dominados por um conceito d numa certa linha de identidade, então um elo de correferência pode ser desenhado de a para b.

Deiteração Qualquer grafo ou elo de correferência cuja ocorrência pudesse ser o resultado de uma iteração pode ser apagado. Pode-se apagar relações duplicadas de qualquer grafo.

Dupla negação Uma dupla negação pode ser removida ou desenhada à volta de qualquer grafo em qualquer contexto.

Junção correferente Dois conceitos idênticos e correferentes no mesmo contexto podem ser juntos, e o elo de correferência entre eles pode ser apagado.

Indivíduos Se um conceito individual a dominar um conceito genérico b em que a e b são correferentes, então *referente*(a) pode ser copiado para b e o elo de correferência pode ser apagado.

O conjunto vazio de grafos $\{\}$ é o único axioma lógico. Qualquer grafo que é demonstrável a partir de $\{\}$ por estas regras é um *teorema*.

Exemplo 71. As proposições "Todas as pessoas são mortais" e "Sócrates é um homem" representam-se com os grafos

$$\neg\boxed{\boxed{\text{PESSOA}} \cdot\cdot \neg\boxed{\cdots \boxed{\text{MORTAL}}}} \qquad \boxed{\text{HOMEM: \texttt{\#Sócrates}}}$$

A seguinte derivação de "Sócrates é mortal" assume que `HOMEM` $<$ `PESSOA`.

1. Inserção (restrição tipo e referente): $\neg\boxed{\boxed{\text{HOMEM: \texttt{\#Sócrates}}} \cdot\cdot \neg\boxed{\cdots \boxed{\text{MORTAL}}}}$
2. Indivíduos: $\neg\boxed{\boxed{\text{HOMEM: \texttt{\#Sócrates}}} \; \neg\boxed{\boxed{\text{MORTAL: \texttt{\#Sócrates}}}}}$
3. Deiteração: $\neg\boxed{\; \neg\boxed{\boxed{\text{MORTAL: \texttt{\#Sócrates}}}}}$
4. Dupla negação: $\boxed{\text{MORTAL: \texttt{\#Sócrates}}}$

Como facilmente se vê, estas regras incluem as proposicionais (Hipótese 4.3.1 na página 107) e também as regras canónicas de formação (Hipótese 3.4.3 na página 76). A cópia é um caso particular da iteração, a simplificação está incluída na deiteração, a restrição de conceitos faz parte da regra de remoção e a junção de dois conceitos obtém-se pela inserção de um elo de correferência seguida de uma junção correferente.

No seu novo livro, Sowa já não considera as regras canónicas de formação um caso particular das regras de inferência, antes pelo contrário. Ao incluirem operações para obtenção de grafos equivalentes, mais gerais ou mais especializados que os grafos a que são aplicadas, as novas regras canónicas de formação (Definição na página 77) formam a base do preocesso dedutivo. Cada regra de inferência não é mais do que uma sequência particular de uma ou mais regras canónicas de formação. Assim, a sua formulação fica muito mais simples. Além disto, Sowa começa por apresentar as regras de inferência de Peirce para os Grafos Existenciais, o que já inclui o tratamento das linhas de identidade, pelo que não repete as regras para os elos de correferência na definição que se segue.

Definição (Sowa). As regras de especialização e generalização permitem formular as regras de inferência para grafos conceptuais de maneira simples, mas geral:

Remoção Num contexto positivo, qualquer grafo u pode ser substituído por uma generalização de u; em particular, u pode ser apagado (isto é, substituído pelo espaço em branco ou grafo vazio, que é uma generalização de todo o grafo).

Inserção Num contexto negativo, qualquer grafo u pode ser substituído por uma especialização de u; em particular, qualquer grafo pode ser inserido (isto é, pode substituir o espaço em branco).

Iteração Se um grafo u ocorre num contexto c, outra cópia de u pode ser desenhada no mesmo contexto c ou em qualquer contexto encaixado em c. Se d é algum conceito de u que não está encaixado dentro de um contexto de u, então a cópia de d é marcada como correferente ao seu original em u.

Deiteração Qualquer grafo u que pudesse ser derivado por iteração pode ser apagado.

Dupla negação Duas negações, sem nada entre a negação interior e a exterior, podem ser desenhadas à volta de ou apagadas de qualquer grafo ou conjunto de grafos em qualquer contexto.

Mesmo considerando a versão corrigida de ϕ e não tendo em conta os tipos de ordem superior, as regras de Sowa estão incompletas. Como há várias maneiras de representar 'verdade', qualquer tautologia que não possa ser obtida à custa das outras tem que ser um axioma lógico. Mas então tem que haver regras para adicionar e remover axiomas em qualquer contexto. De outro modo não será possível simular a regra da instanciação universal com grafos conceptuais. Usando a notação tradicional (isto é, sem *Holds*) para maior clareza, considere a fórmula $\forall x\ Gato(x) \vdash Gato(garfield)$. Partindo da hipótese, $\neg\boxed{\boxed{\top_c}\cdot\cdot\neg\boxed{\boxed{\text{GATO}}}}$, vejamos até onde conseguiremos chegar. Restringindo o referente no contexto com encaixe ímpar, obtém-se $\neg\boxed{\boxed{\top_c\text{: }\texttt{\#Garfield}}\cdot\cdot\neg\boxed{\boxed{\text{GATO}}}}$. Pela regra dos indivíduos, o referente pode ser copiado do conceito individual dominante para o conceito genérico dominado. Daqui resulta $\neg\boxed{\boxed{\top_c\text{: }\texttt{\#Garfield}}\ \neg\boxed{\boxed{\text{GATO: }\texttt{\#Garfield}}}}$ mas não é possível obter apenas $\boxed{\text{GATO: }\texttt{\#Garfield}}$ porque não se pode apagar grafos em contextos ímpares.

No restante desta secção apresento a nova formulação das regras de inferência. Tal como em Sowa, cada regra de inferência corresponde a uma ou mais regras canónicas, usando-se obviamente a Hipótese 59 na página 80. Como tal, e porque os grafos verdadeiros têm que ser canónicos, assume-se que um passo dedutivo só será realizado se o resultado estiver bem-formado e obedecer às restrições impostas pelo cânone. No geral, as regras de inferência são parecidas com as de Sowa. As diferenças mais salientes prendem-se com a hierarquia dos marcadores, com os axiomas lógicos a usar, e com os elos de correferência.

Como os marcadores individuais estão parcialmente ordenados, poder-se-á pensar que as operações de relaxamento e restrição também se aplicam aos referentes e já não apenas aos tipos dos vértices. Mas como se viu no fim da Secção 6.2 na página 103, a hierarquia de tipos deixa de fazer sentido quando os tipos são utilizados como marcadores. Para fins dedutivos, os marcadores individuais continuam pois a ser incomparáveis, caso contrário poder-se-ia inferir grafos falsos a partir de grafos verdadeiros. Como exemplo, considere de novo o caso `GATO` $<$ `FELINO` em que `GATO` é uma espécie e mas `FELINO` não. Então, se $\boxed{\text{ESPÉCIE: }\texttt{\#gato}}$ estiver num contexto com encaixe par, é verdadeiro mas não pode ser generalizado para $\boxed{\text{ESPÉCIE: }\texttt{\#felino}}$, que é falso. De modo análogo, se o segundo conceito estiver num contexto com encaixe ímpar não poderá ser transformado no primeiro. É possível no entanto restringir o referente para o marcador absurdo, porque $\boxed{\text{ESPÉCIE: }\bar{*}}$ é tão falso quanto $\boxed{\text{ESPÉCIE: }\texttt{\#felino}}$.

No entanto, pode-se tirar partido da hierarquia de marcadores se houver um elo de correferência. Seja m o marcador de um conceito dominado ou dominante c cujo identificador é a variável x. Se m for um tipo, a condição $x \sqsubseteq m$ encontra-se no contexto de c. Se esse contexto for par, a condição é verdadeira pelo que $x \sqsubseteq m'$, em que m' é um supertipo de m, também o é. Se o contexto for ímpar, a condição é falsa e

Contexto de c	Elo de correferência?	m	Acção
par	não	$\bar{*}$	generalizar
par	não	$\neq\bar{*}$	generalizar para $*$
par	sim	qualquer	generalizar
ímpar	não	$*$	especializar
ímpar	não	$\neq *$	especializar para $\bar{*}$
ímpar	sim	qualquer	especializar

Tabela 6.1: Condições para alterar o referente m do conceito c

restringindo m a um subtipo não a torna verdadeira. A Tabela 6.1 resume todos estes casos.

Considere um conceito c_1, com marcador m_1 e identificador x_1, dominando um conceito c_2 com marcador m_2 e variável associada x_2. Então, devido ao elo de correferência tem-se $x_2 = x_1$ pelo que se pode iterar qualquer condição sobre x_1 do contexto de c_1 para o contexto de c_2. Na prática, isto resume-se a substituir m_2 por m_{12}, o ínfimo de m_1 e m_2. De facto, assumindo $x_2 \doteq x_1$, tem-se que $x_1 \sqsubseteq m_1$ e $x_2 \sqsubseteq m_2$ implica $x_2 \sqsubseteq m_{12}$ (ver o Exemplo 72 na página 117). É de salientar que esta especialização de m_2 não pode ser feita se o resultado for o marcador absurdo, porque se c_2 tiver com encaixe par o resultado será falso. Se c_2 estiver num contexto ímpar, esta regra, que encontra um "limite superior" para o valor de x_2, é desnecessária pois pode-se aplicar directamente a última regra da Tabela 6.1.

Quanto aos axiomas lógicos, Sowa apenas utilizava o conjunto vazio de grafos. Como se viu, é preciso um predicado $verdade(x)$ para poder representar todas as fórmulas fechadas da lógica de primeira ordem. Esse predicado acabou por ser a igualdade $\doteq$. Assim, os novos axiomas são todos os grafos cuja tradução é uma tautologia $x \doteq x$ ou $\neg\neg x \doteq x$. Olhando para a Hipótese 66 na página 100, obtêm-se as possibilidades dadas na Tabela 6.2 na página oposta, em que m e m' são marcadores quaisquer mas diferentes de $\bar{*}$ e t e t' são tipos de conceito quaisquer (embora as traduções apresentadas assumam que são não-relacionais).

Claro que os axiomas com $\top_r$ e $\bot_r$ podem ter aridade arbitrária. No entanto, enquanto a mera presença do tipo $\bot_r$ torna automaticamente todo o grafo falso, pelo que os conceitos usados como argumentos podem ser arbitrários, o mesmo não acontece com $\top_r$. Os conceitos a que está ligado também têm que ser verdadeiros para que todo o grafo o seja. É também de notar que ϕ traduz um contexto vazio da mesma maneira que $\boxed{\top_c}$. Como as regras de inferência irão permitir inserir e remover axiomas lógicos de qualquer contexto, o conjunto vazio de grafos torna-se assim dispensável porque pode ser obtido a partir de qualquer outro axioma por remoção. No entanto ele será mantido por conveniência. Outros dos grafos acima escritos também são redundantes. Por exemplo, qualquer $\neg\boxed{\boxed{\bot_c : m}}$ pode ser obtido de $\neg\boxed{\boxed{\bot_c : *}}$ por restrição do referente (que não é um marcador individual). Do mesmo modo, $\neg\boxed{(\bot_r) \to \boxed{t : m}}$ pode ser derivado de $\neg\boxed{(\bot_r) \to \boxed{\top_c : *}}$. O conjunto final de axiomas encontra-se na Hipótese 70 na página 116.

$$
\begin{aligned}
\phi(\boxed{}) &= \exists x\ x \doteq x \\
\phi(\boxed{\top_c : m}) &= \exists x\ x \doteq x \land x \doteq c_m \\
\phi(\neg\boxed{\boxed{\bot_c : m}}) &= \neg\exists x\ \neg x \doteq x \land x \doteq c_m \\
\phi(\neg\boxed{\boxed{t\!: \bar{*}}}) &= \neg\exists x\ \mathit{Holds}_1(c_t, x) \land \neg x \doteq x \\
\phi(\neg\boxed{(\bot_r) \to \boxed{t : m}}) &= \neg\exists x\ \neg x \doteq x \land \mathit{Holds}_1(c_t, x) \land x \doteq c_m \\
\phi(\neg\boxed{\boxed{t : m} \to (\bot_r) \to \boxed{t' : m'}}) &= \neg\exists x \exists y\ \neg x \doteq x \land \mathit{Holds}_1(c_t, x) \land x \doteq c_m \land \\
&\quad\ \mathit{Holds}_1(c_{t'}, y) \land y \doteq c_{m'} \\
\phi((\top_r) \to \boxed{\top_c\!: m}) &= \exists x\ x \doteq x \land x \doteq x \land x \doteq c_m \\
\phi(\boxed{\top_c\!: m} \to (\top_r) \to \boxed{\top_c\!: m'}) &= \exists x \exists y\ x \doteq x \land x \doteq x \land x \doteq c_m \land y \doteq y \land y \doteq c_{m'}
\end{aligned}
$$

Tabela 6.2: Possíveis axiomas lógicos

Finalmente, em relação aos elos de correferência, mantém-se no geral as regras de Sowa tendo em atenção os casos em que os marcadores envolvidos são tipos. A explicação que se segue utiliza de novo dois conceitos c_1 e c_2 com marcadores m_1 e m_2 e identificadores x_1 e x_2. Relembro que é pressuposto o contexto de c_1 dominar o de c_2. Além disso, de modo a tornar a exposição mais clara e concisa, voltarei a utilizar $x \sqsubseteq y$ como abreviatura de $\neg\exists z\ \mathit{Holds}_1(x, z) \land \neg \mathit{Holds}_1(y, z)$. Vejamos então o que acontece quando se insere um elo de correferência entre c_1 e c_2:

1. A condição $x_2 \doteq x_1$ é acrescentada ao contexto de c_2.

2. A condição $x_2 = c_{m_2}$ passa a $x_2 \sqsubseteq c_{m_2}$ se $m_2 \notin T_0^{nc} \cup \{*, \bar{*}\}$.

3. A condição $x_1 = c_{m_1}$ passa a $x_1 \sqsubseteq c_{m_1}$ se $m_1 \notin T_0^{nc} \cup \{*, \bar{*}\}$.

A remoção de uma inserção consiste em fazer os passos inversos. Devido ao ponto 1, um elo de correferência não pode ser inserido se c_2 estiver num contexto com encaixe par, nem pode ser removido se c_2 estiver num contexto com encaixe ímpar. Mas imaginemos que m_2 está nas condições do ponto 2. Então ao inserir um elo de correferência está-se a relaxar uma condição pelo que c_2 não pode estar num contexto ímpar. Caso contrário seria eventualmente possível obter uma condição verdadeira a partir de uma falsa. Inversamente, a remoção do elo de correferência fortifica a condição, que passa de $x_2 \sqsubseteq c_{m_2}$ a $x_2 = c_{m_2}$. Neste caso c_2 não pode ter profundidade par de modo a não se obter uma condição falsa a partir de uma verdadeira. Por um raciocínio análogo, se m_1 estiver nas condições do ponto 3, não se pode inserir o elo de correferência se c_1 estiver num contexto com encaixe ímpar, nem se pode remover se estiver num contexto com encaixe par.

Parece pois que se m_2 estiver nas condições do ponto 2, então não se pode mexer em c_2. Felizmente existe uma excepção, nomeadamente quando $m_1 = m_2$. Considere-se o caso em que c_2 está num contexto par e portanto o elo de correferência vai ser removido. Obtém-se então $x_2 \doteq m_2$. Mas como o elo de correferência também foi

Encaixe c_1	Encaixe c_2	$m_1 \in T_0^{nc} \cup \{*, \bar{*}\}$	$m_2 \in T_0^{nc} \cup \{*, \bar{*}\}$	Acção
qualquer	par	sim	sim	remoção
qualquer	ímpar	sim	sim	inserção
ímpar	par	não	sim	remoção
par	ímpar	não	sim	inserção
ímpar	par	não	não	remoção se $m_1 = m_2$
par	ímpar	não	não	inserção se $m_1 = m_2$

Tabela 6.3: Condições para alteração de elos de correferência

removido de c_1, o contexto dominante contém a condição $x_1 \doteq m_1$. Devido à hipótese $m_1 = m_2$ o contexto de c_2 contém pois um facto que já se encontra no antecedente, deste modo correspondendo a remoção do elo de correferência a uma iteração. A outra possibilidade dá-se quando c_2 está num contexto com encaixe ímpar e o elo de correferência é inserido. A nova condição $x_2 \sqsubseteq c_{m_2} \wedge x_2 \doteq x_1$ corresponde a $x_1 \sqsubseteq c_{m_2}$. Mas como o novo elo de correferência gerou a condição $x_1 \sqsubseteq c_{m_1}$ no contexto de c_1, estamos de novo perante a iteração de uma restrição do contexto dominante para o dominado.

A Tabela 6.3 mostra em que condições se pode inserir ou remover um elo de correferência. Nos casos que não se encontram listados não é possível fazer nenhuma das duas acções. Claro que não é preciso verificar quaisquer restrições se o elo de correferência a acrescentar ou apagar for uma cópia exacta de outro existente. Podemos então por fim apresentar as regras de inferência.

Hipótese 70. Seja S um conjunto de grafos canónicos contidos no contexto exterior. Qualquer grafo derivado de S pelas seguintes *regras de inferência de primeira ordem* é dito *demonstrável* a partir de S.

Equivalência Em qualquer contexto pode-se inserir ou remover um axioma, pode-se aplicar as regras canónicas de equivalência e pode-se inserir ou remover uma dupla negação à volta de qualquer conjunto de grafos.

Generalização Num contexto com encaixe par pode-se aplicar qualquer regra canónica de generalização desde que obedeça às condições da Tabela 6.1 na página 114 e da Tabela 6.3.

Especialização Num contexto com encaixe ímpar pode-se inserir qualquer grafo canónico e pode-se aplicar as regras canónicas de especialização desde que obedeçam às condições da Tabela 6.1 na página 114 e da Tabela 6.3.

Iteração Pode-se copiar um grafo de um contexto c para qualquer contexto dominado por c e pode-se inserir elos de correferência entre os conceitos originais e as respectivas cópias.

Deiteração O resultado de uma possível iteração pode ser apagado.

Transitividade Se o conceito c_1 dominar o conceito c_2 que por sua vez domina $c_3 \neq c_1$, então pode-se remover ou inserir um elo de correferência entre c_1 e c_3. Se for inserido, pode-se remover o elo de correferência entre c_2 e c_3.

Individualização Se o conceito c_1 dominar o conceito c_2 então pode-se substituir $referente(c_2)$ por $referente(c_1) \wedge referente(c_2)$ desde que o resultado não seja $\bar{*}$.

Um *teorema* é um grafo demonstrável unicamente a partir dos seguintes *axiomas lógicos*:

- O conjunto vazio de grafos $\{\}$;
- $\boxed{\top_c\colon m}$ para qualquer $m \in \mathcal{M} - \{\bar{*}\}$;
- $\neg\boxed{\boxed{\bot_c\colon *}}$;
- $\neg\boxed{\boxed{\top_c\colon \bar{*}}}$;
- $\neg\boxed{(\bot_r)\rightarrow\boxed{\top_c\colon *}}$ e para todas as outras aridades;
- $(\top_r)\rightarrow\boxed{\top_c\colon m}$ para todo o $m \in \mathcal{M} - \{\bar{*}\}$ e para qualquer aridade.

Exemplo 72. Aplicando a regra da individualização duas vezes, e considerando o Exemplo 13 na página 43 e o Exemplo 29 na página 57, $\boxed{\text{FORMA: \#losango}}\cdots\boxed{\text{FORMA: \#rectângulo}}$ transforma-se primeiro em $\boxed{\text{FORMA: \#losango}}\cdots\boxed{\text{FORMA: \#quadrado}}$ e depois deriva-se $\boxed{\text{FORMA: \#quadrado}}\cdots\boxed{\text{FORMA: \#quadrado}}$

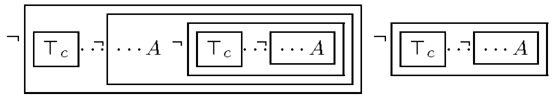
Os restantes exemplos servirão para provar a completude das regras.

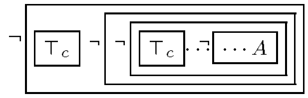
Exemplo 73. Seja A uma fórmula em que a variável x não ocorre livre (ver Exemplo 63 na página 102). A derivação que se segue é $\forall x\ A \vdash A$. Daqui resulta, segundo o teorema da dedução, $\vdash (\forall x\ A) \rightarrow A$.

1. Hipótese: $\neg\boxed{\boxed{\top_c}\ \neg\boxed{A}}$
2. Equivalência (remoção axioma): $\neg\boxed{\neg\boxed{A}}$
3. Equivalência (dupla negação): A

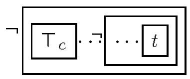
Exemplo 74. A regra da *generalização* $A \vdash \forall x\ A$ tem interesse quando x ocorre livre em A. Os grafos $\cdots A$ e $\neg\boxed{\boxed{\top_c}\cdots\neg\boxed{\cdots A}}$ representam fidedignamente as fórmulas A e $\forall x\ A$, respectivamente. Mas como o primeiro grafo está incompleto, é preciso fazer o seu fecho universal: $\neg\boxed{\boxed{\top_c}\cdots\neg\boxed{\cdots A}}$. Então o resultado da aplicação da regra, que consiste em subsituir o subgrafo $\cdots A$ por $\neg\boxed{\boxed{\top_c}\cdots\neg\boxed{\cdots A}}$, deverá ser $\neg\boxed{\boxed{\top_c}\ \neg\boxed{\neg\boxed{\boxed{\top_c}\cdots\neg\boxed{\cdots A}}}}$. Eis a derivação correspondente:

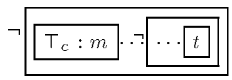
1. Hipótese: $\neg\boxed{\boxed{\top_c}\cdots\neg\boxed{\cdots A}}$
2. Equivalência (cópia): $\neg\boxed{\boxed{\top_c}\cdots\neg\boxed{\cdots A}}\ \neg\boxed{\boxed{\top_c}\cdots\neg\boxed{\cdots A}}$

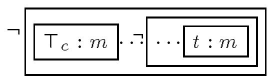
3. Iteração: 

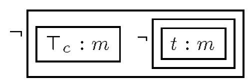
4. Generalização (remoção 2 grafos): 

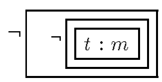
Exemplo 75. Seja t um tipo não-relacional e seja m um marcador individual. A fórmula $(\forall x\ \mathit{Holds}_1(c_t, x)) \rightarrow \mathit{Holds}_1(c_t, c_m)$ obtém-se por aplicação do teorema da dedução à seguinte derivação.

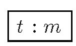
1. Hipótese: 

2. Especialização (restrição): 

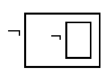
3. Individualização: 

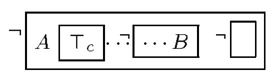
4. Generalização (remoção elo de correferência): 

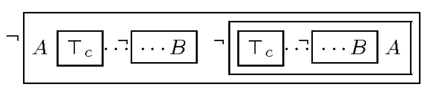
5. Equivalência (remoção axioma): 

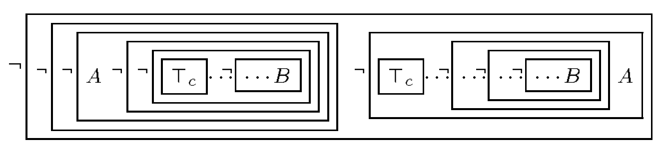
6. Equivalência (dupla negação): 

Exemplo 76. O teorema $(\forall x\ (A \rightarrow B)) \rightarrow (A \rightarrow \forall x\ B)$ do Exemplo 64 na página 102, em que x não ocorre livre em A mas sim em B, deriva-se assim:

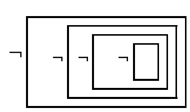
1. Equivalência(dupla negação): 

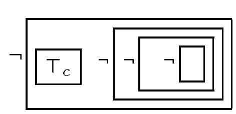
2. Especialização (2 inserções): 

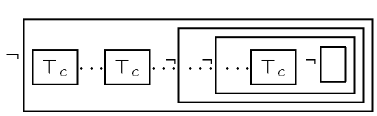
3. 2 Iterações: 

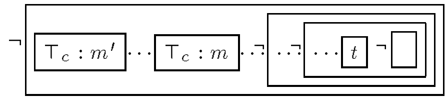
4. Equivalência (3 duplas negações): 

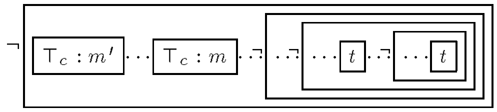
Exemplo 77. A fórmula $c_m \doteq c_{m'} \rightarrow (\mathit{Holds}_1(c_t, c_m) \rightarrow \mathit{Holds}_1(c_t, c_{m'}))$ é um teorema como a seguinte derivação comprova.

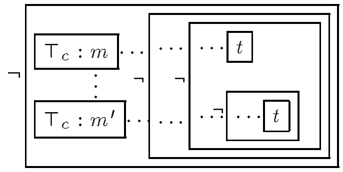
1. Equivalência (2 duplas negações): 

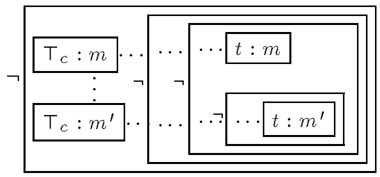
2. Equivalência (inserção axioma): 

3. 2 Iterações: 

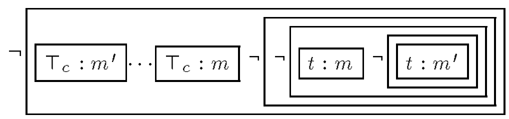
4. Especialização (3 restrições): 

5. Iteração: 

6. 2 Transitividades: 

7. Individualização: 

8. Generalização (remoção elos de correferência):



Teorema 71. *As regras de inferência de primeira ordem e os axiomas lógicos formam um sistema formal completo e consistente para a lógica de primeira ordem com igualdade.*

Demonstração. Hamilton [1988] apresenta o seguinte sistema, que demostra ser completo e consistente, em que A, B e C são fórmulas, x é uma variável, $t, t_1, \ldots$ são termos, f uma função e P um predicado.

K1 $A \rightarrow (B \rightarrow A)$

K2 $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

K3 $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$

K4 $(\forall x\ A) \rightarrow A$ se x não ocorrer livre em A

K5 $(\forall x\ A(x)) \rightarrow A(t)$ se $A(x)$ for uma fórmula e t um termo livre para x em $A(x)$

K6 $(\forall x\ A \rightarrow B) \rightarrow (A \rightarrow \forall x\ B)$ se A não contiver ocorrências livres de x

E7 $x \doteq x$

E8 $t_k \doteq t \rightarrow f(t_1, \ldots, t_k, \ldots, t_n) \doteq f(t_1, \ldots, t, \ldots, t_n)$

E9 $t_k \doteq t \rightarrow (P(t_1, \ldots, t_k, \ldots, t_n) \rightarrow P(t_1, \ldots, t, \ldots, t_n))$

R1 $\{A, A \rightarrow B\} \vdash B$

R2 $A \vdash \forall x\ A$

Os axiomas K1–K3 e a regra de *modus ponens* R1 já foram tratados pelo Teorema 69 na página 109. O axioma K4 e a regra R2 são demonstrados pelo Exemplo 73 na página 117 e pelo Exemplo 74 na página 117. Quanto a K5, o caso geral para fórmulas arbitrárias (com várias ocorrências da variável x e em que o termo t é o marcador genérico ou um marcador individual) pode ser obtido aplicando a derivação do Exemplo 75 na página 118. Para o axioma K6 há duas possibilidades consoante x ocorra livre em B ou não. O segundo caso corresponde ao Exemplo 76 na página 118, o primeiro é idêntico exceptuando o elo de correferência. Quanto ao axioma E7, depois de fechado universalmente, corresponde ao grafo do Exemplo 62 na página 102 que também é um axioma na Hipótese 70 na página 116. O axioma E8 não se aplica porque a linguagem utilizada não tem símbolos de função. O axioma E9 foi demonstrado para o caso de P ser um predicado unário. O caso geral é semelhante.

Em relação à consistência, apresentarei apenas os contornos da demonstração completa para a parte não proposicional. A validade dos axiomas é provada pela Tabela 6.2 na página 115. As regras de equivalência são obviamente consistentes porque limitam-se a manter o valor de verdade de um grafo através da duplicação de fórmulas ou remoção de fórmulas repetidas. As regras de generalização apagam condições, seja através do relaxamento de tipos, que corresponde à aplicação dos axiomas próprios (Hipótese 67 na página 103), ou através da remoção concreta de (sub)grafos. Como um contexto com encaixe par é verdadeiro, as condições que sobram depois de aplicadas as regras de generalização continuam verdadeiras. Nos contextos ímpares pode-se acrescentar condições ou tornar as que existem mais restritivas. É isso que fazem as regras de especialização. A consistência das condições que as regras de especialização e generalização devem observar (Tabela 6.1 na página 114 e Tabela 6.3 na página 116) já foi explicada anteriormente. As regras de iteração e deiteração limitam-se a copiar informação dos antecedentes para os consequentes de uma implicação, ou a remover consequentes que sejam também antecedentes. Os elos de correferência podem ser inseridos ou apagados porque tratam-se dos mesmos conceitos.

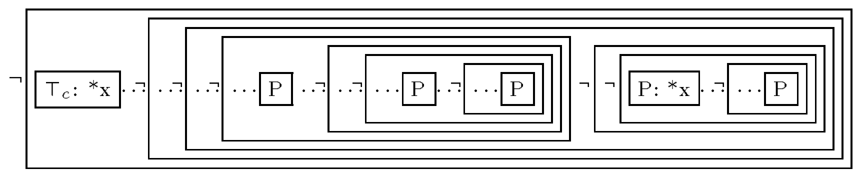
A consistência da regra da transitividade mostra-se facilmente. Sejam c_1, c_2 e c_3 três conceitos com marcadores m_1, m_2, m_3 e identificadores x_1, x_2 e x_3, respectivamente. Por hipótese, existe uma linha de identidade que liga c_1 a c_2 e c_2 a c_3. Ou seja, no contexto de c_2 tem-se $x_2 \doteq x_1$ e no de c_3 tem-se $x_3 \doteq x_2$. Pode-se pois acrescentar $x_3 \doteq x_1$ que é representada por um elo de correferência directo entre c_1 e c_3. Inversamente, tendo as três igualdades, uma deles torna-se redundante e pode ser eliminada. No entanto, devido à negação e ao encaixe dos contextos, apenas se pode copiar condições de contextos dominantes para dominados. Por isso, a igualdade $x_2 \doteq x_1$ não pode ser eliminada. Repare-se que após a aplicação da regra da transitividade, cada conceito continua a ter pelo menos um elo de correferência, pelo que não há o perigo de alguma condição $x_i \sqsubseteq m_i$ desaparecer ou ser acrescentada. Finalmente, a individualização limita-se a iterar uma condição de um contexto dominante para um dominado. A sobreposição das duas condições resulta em achar o ínfimo dos dois referentes. □

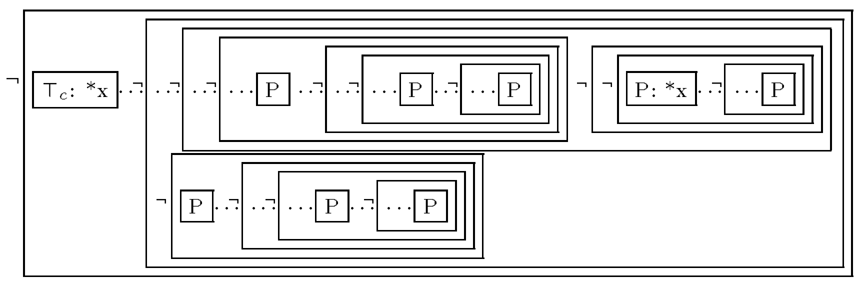
A simulação de uma prova segundo o sistema de Hamilton faz-se num contexto de profundidade 2, contendo o contexto envolvente, de profundidade 1, um conceito $\boxed{\top_c}$ por cada variável livre das fórmulas representadas pelos grafos no contexto interior.

Exemplo 78. A demonstração de $\forall x\ P(x) \rightarrow P(x)$ é feita do seguinte modo:

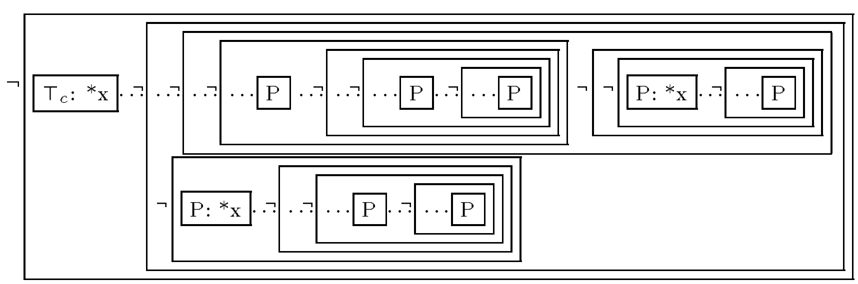
1. K2: $(P(x) \to ((P(x) \to P(x)) \to P(x))) \to ((P(x) \to (P(x) \to P(x))) \to (P(x) \to P(x)))$
2. K1: $P(x) \to ((P(x) \to P(x)) \to P(x))$
3. Modus ponens: $(P(x) \to (P(x) \to P(x))) \to (P(x) \to P(x))$
4. K1: $P(x) \to (P(x) \to P(x))$
5. Modus ponens: $P(x) \to P(x)$
6. Generalização: $\forall x\ P(x) \to P(x)$

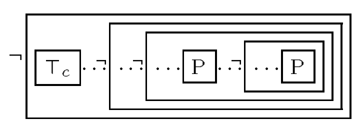
Devido ao tamanho dos grafos correspondentes, apresentarei apenas a derivação a partir do terceiro passo. Além disso, para facilitar a composição do texto os elos de correferência serão representados tanto por linhas pontilhadas como por variáveis.

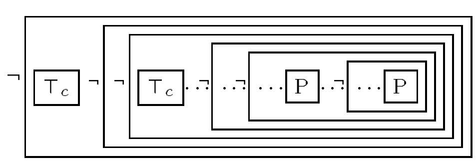
3. Modus ponens: 

4. (a) K1: 

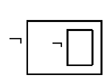
   (b) Especialização (inserção elo de correferência):



5. Modus ponens: 

6. Generalização: 

Usando directamente as regras de inferência dos grafos conceptuais, a demonstração fica muito mais simples:

1. Equivalência (dupla negação): 

2. Especialização (inserção): $\neg\boxed{\boxed{\mathrm{P}}\ \neg\boxed{}}$

3. Iteração: $\neg\boxed{\boxed{\mathrm{P}}\cdot\cdot\neg\boxed{\cdots\boxed{\mathrm{P}}}}$

Note-se que este último grafo representa a fórmula pretendida, $\forall x\ P(x) \rightarrow P(x)$. As duplas negações e os $\boxed{\top_c}$ obtidos adicionalmente na demonstração segundo o método de Hamilton apenas servem para tornar a tradução para grafos conceptuais mais sistemática.

Capítulo 7

Conclusões e Perspectivas

Esta dissertação pretendeu dar um contributo teórico aos actuais esforços de implementação e padronização da Teoria das Estruturas Conceptuais. Penso que a proposta apresentada nos capítulos anteriores atingiu os objectivos principais expostos na introdução: extensão, clarificação e formalização das noções básicas.

As extensões incluem algumas das ideias e tendências recentes na comunidade de Estruturas Conceptuais, como por exemplo o marcador absurdo, a organização hierárquica dos tipos de relação, e sobretudo os tipos de ordem superior. Devido a estes últimos, a teoria proposta contém alguns mecanismos básicos para especificação e manipulação de meta-grafos, isto é, grafos que representam propriedades dos tipos usados noutros grafos. Quanto à especificação de meta-grafos, os mecanismos dados por este trabalho são os tipos de ordem superior e as dependências entre vértices, as quais englobam não só os elos de correferência como também os operadores τ e ρ propostos por Sowa [1993]. Em relação à manipulação dos meta-grafos, as regras de inferência foram alteradas de modo a contemplar tipos de ordem superior e elos de correferência, tendo em conta que um grafo com dependências, sejam elas apenas elos de correferência ou não, representan todas as suas instâncias que satisfazem as dependências.

Assim como as modificações feitas à TEC original foram guiadas sobretudo pelo novo sistema de tipos, a clarificação das noções básicas da teoria também teve uma linha condutora, nomeadamente a divisão dos grafos em quatro níveis de significação. Cada nível é um conjunto de grafos que obedecem a certas restrições, as quais aumentam conforme se desce de nível. Assim, cada nível inclui todos os grafos dos níveis inferiores. O primeiro nível é o dos grafos conceptuais sem restrições semânticas adicionais. O segundo nível é o dos grafos bem-formados, isto é, dos grafos que obedecem às restrições de ordem, género e aridade impostas pelas hierarquias de tipos. No terceiro nível estão os grafos canónicos, os quais obedecem às restrições ontológicas dadas pelo cânone, o qual inclui as hierarquias de tipos. Deste modo, os grafos canónicos têm que estar bem-formados. Finalmente, o último nível é o dos grafos verdadeiros, aqueles que são deriváveis pelas regras de inferência a partir de um conjunto inicial de grafos, assumidos como verdadeiros, dados pela base de conhecimentos. Como cada regra de inferência consiste na aplicação de uma ou mais regras canónicas de formação em determinadas circunstâncias, os grafos verdadeiros são canónicos.

As principais inovações deste trabalho em relação à TEC original foram de facto o sistema de tipos, as dependências, e o tratamento da canonicidade. O sistema de tipos não só inclui os tipos de ordem superior, como também um novo género de tipos de conceito, os relacionais. Além disso, os tipos de relação foram igualmente organizados numa hierarquia, o que permite estender as regras canónicas de restrição e relaxamento aos vértices relacionais. As regras canónicas de formação sofreram ainda outras alterações de pormenor, mas significativas, porque devem garantir que os grafos gerados, os grafos canónicos, estejam ontologicamente correctos. Foi este o significado intuitivo que dei aos grafos canónicos, separando-os desse modo claramente dos grafos bem-formados (isto é, bem tipados) e dos verdadeiros. Em todo o trabalho, as definições formais tentam reflectir a noção informal, o que no caso da canonicidade levou igualmente à alteração profunda da relação de conformidade.

Houve também uma certa preocupação com questões de implementação. Por um lado, a verificação da boa-formação ou da canonicidade de um dado grafo é eficiente em muitos casos. Por outro lado, os tipos encontram-se organizados em reticulados, o que faz com que os marcadores, os conceitos e as relações também o sejam. Daqui resulta, através da noção de projecção, uma ordem parcial sobre os próprios grafos, o que permite o armazenamento e pesquisa eficientes de uma base de grafos. Esta organização hierárquica dos vários componentes da teoria, organização essa que inclui os quatro níveis de significação, não só tem vantagens computacionais como torna o formalismo mais simples, elegante e regular.

Pela própria natureza de um trabalho científico, este encontra-se incompleto, podendo ser melhorado e estendido em muitos aspectos que foram simplificados na teoria agora proposta. Por exemplo, para especificar uma ontologia mais rica e expressiva, convém poder ter grafos com contextos e dependências na base canónica. Claro que isso envolve a criação de regras canónicas de formação específicas para esses casos. Seria igualmente útil se o cânone pudesse incluir condições de subcategorização. Por exemplo, convém poder exprimir a restrição "só os seres humanos falam". Isto significa que sempre que o primeiro argumento da relação `AGNT` for `FALAR` o segundo terá que ser `PESSOA`. De modo a minimizar o esforço dispendido pelo engenheiro do conhecimento na criação de uma ontologia, convém haver bibliotecas de cânones, cada um para um determinado domínio bastante específico. Fazendo a analogia com a noção de "módulo" de certas linguages de programação, um cânone poderá ser encarado como um "pacote de conhecimento" que importa tipos, marcadores e grafos base de outros cânones, exportando por sua vez informação para outros pacotes. Uma formalização simples deste mecanismo para estruturar cânones encontra-se descrita em [Wermelinger e Bejan, 1993]. Para terminar, é de referir que as regras de inferência não tiram completamente proveito dos tipos de ordem superior e das dependências, limitando-se aos elos de correferência e à inferência de primeira ordem. A própria noção de dependência, que neste momento se resume a uma igualdade entre tipos e/ou referentes de vértices, poderá ser generalizada de modo a incluir as relações hierárquicas dos tipos e marcadores. Ou seja, talvez seja útil poder impôr condições como $tp(c) > rf(c')$, em que c e c' são vértices conceptuais.

Agradecimentos

Esta dissertação beneficiou do apoio e da ajuda de várias pessoas, às quais não posso deixar de agradecer:

Ao meu orientador, José Gabriel Lopes, pelo apoio constante e pelas inúmeras sugestões que melhoraram grandemente o trabalho realizado e a sua exposição.

Ao Luís Caires, pelas muitas horas que disponibilizou—mesmo durante a escrita da sua própria dissertação—para discutir a teoria, para ler e comentar uma versão preliminar deste texto, e para me instalar o Linux no computador.

À Margarida Mamede, por me ter poupado muitas horas na preparação de aulas práticas e na avaliação de exames, permitindo-me assim uma maior dedicação a este trabalho.

Aos meus colegas habituais de almoço, pelo agradável convívio que ajudou a suportar o *stress*.

À Claudia, pelo apoio e compreensão durante estes longos meses em que passei muitos serões e fins-de-semana em frente ao computador em vez de lhe dedicar a atenção que merecia.

Bibliografia

[Aït-Kaci *et al.*, 1989] Hassan Aït-Kaci, Robert Boyer, Patrick Lincoln, e Roger Nasr. Efficient implementation of lattice operations. *ACM TOPLAS*, 11(1):115–146, Janeiro 1989.

[Baader *et al.*, 1992] Franz Baader, Bernhard Hollunder, Bernhard Nebel, Hans-Jürgen Profitlich, e Enrico Franconi. An empirical analysis of optimization techniques for terminological representation systems. In Bernhard Nebel, Charles Rich, e William Swartout, editores, *Principles of Knowledge Representation and Reasoning—Proceedings of the 3rd International Conference*. Morgan Kaufmann, Outubro 1992.

[Beierle *et al.*, 1990] C. Beierle, U. Hedstück, U. Pletat, P. H. Schmitt, e J. Siekmann. An order-sorted logic for knowledge representation systems. Relatório Técnico 113, IWBS, Abril 1990.

[Boksenbaum *et al.*, 1993] Claude Boksenbaum, Boris Carboneill, Olivier Haemmerlé, e Thérèse Libourel. Conceptual Graphs for relational databases. In Mineau et al. [1993a], pp. 142–161.

[Buvač e Mason, 1993] Saša Buvač e Ian A. Mason. Propositional logic of context. In *Proceedings of the Eleventh NCAI*, pp. 412–419, Washington, DC, 11–15 Julho 1993.

[Carboneill e Haemmerle, 1994] Boris Carboneill e Ollivier Haemmerle. Standardizing and interfacing relational databases using conceptual graphs. In Tepfenhart et al. [1994a], pp. 311–330.

[Chein e Mugnier, 1992] Michel Chein e Marie-Laure Mugnier. Conceptual graphs: fundamental notions. *Révue d'Intelligence Artificielle*, 6(4):365–406, 1992.

[Chein e Mugnier, 1993] M. Chein e M. L. Mugnier. Specialization: Where do the difficulties occur? In Pfeiffer e Nagle [1993], pp. 229–238.

[Davey e Priestley, 1990] B. A. Davey e H. A. Priestley. *Introduction to Order and Lattices*. Cambridge University Press, 1990.

[Delugach, 1992] Harry S. Delugach. Analysing multiple views of software requirements. In Nagle et al. [1992], pp. 391–410.

[Ellis e Lehmann, 1994] Gerard Ellis e Fritz Lehmann. Exploiting the induced order on type-labeled graphs for fast knowledge retrieval. In Tepfenhart et al. [1994a], pp. 293–310.

[Ellis e Levinson, 1992] Gerard Ellis e Robert A. Levinson, editores. *Proceedings of the First International Workshop on PEIRCE: A Conceptual Graphs Workbench*, Las Cruces, New Mexico, 10 Julho 1992. University of Queensland Technical Report 241.

[Ellis e Levinson, 1993] Gerard Ellis e Robert Levinson. The birth of PEIRCE: A conceptual graphs workbench. In Pfeiffer e Nagle [1993], pp. 219–228.

[Ellis e Levinson, 1994] Gerard Ellis e Robert Levinson, editores. *Proceedings of the Third International Workshop on PEIRCE: A Conceptual Graphs Workbench*, College Park MD, USA, 19 Agosto 1994. University of Maryland.

[Ellis, 1992] Gerard Ellis. Compiled hierarchical retrieval. In Nagle et al. [1992], pp. 271–294.

[Ellis, 1993] Gerard Ellis. Efficient retrieval from hierarchies of objects using lattice operations. In Mineau et al. [1993a], pp. 274–293.

[Esch, 1993a] John W. Esch. Contexts as white box concepts. In Mineau et al. [1993b], pp. 17–29.

[Esch, 1993b] John W. Esch. The scope of coreference in conceptual graphs. In Pfeiffer e Nagle [1993], pp. 3–12.

[Esch, 1994a] John Esch. Contexts and concepts: Abstraction duals. In Tepfenhart et al. [1994a], pp. 175–184.

[Esch, 1994b] John Esch. Contexts, canons and coreferent types. In Tepfenhart et al. [1994a], pp. 185–195.

[Esch, 1994c] John Esch. Similarities of microtheory and conceptual graphs contexts. In Tepfenhart et al. [1994b], pp. 61–69.

[Gardiner *et al.*, 1992] David A. Gardiner, Bosco S. Tjan, e James R. Slagle. Extending conceptual structures: Representation issues and reasoning operations. In Nagle et al. [1992], pp. 67–85.

[Genesereth e Fikes, 1992] Michael R. Genesereth e Richard E. Fikes. Knowledge interchange format version 3.0 reference manual. Relatório Técnico Logic-92-1, Computer Science Department, Stanford University, Junho 1992. “Living document” of the Interlingua Working Group of the DARPA Knowledge Sharing Effort.

[Guha, 1991] R. V. Guha. Contexts: a formalization and some applications. Relatório Técnico ACT-CYC-423-91, MCC, 1991.

[Hamilton, 1988] A. G. Hamilton. *Logic for Mathematicians*. Cambridge University Press, 1988. Edição revista.

[IGES/PDES Organization, 1992] Dictionary/Methodology Committee of the IGES/PDES Organization. Semantic unification meta model—volume 1: Semantic unification of static models. Document ISO TC184/SC4/WG3/N103, International Organisation for Standardization, 19 Outubro 1992.

[JETAI, 1992] *Journal of Experimental and Theoretical Artificial Intelligence*, 4(2), Abril–Junho 1992. Número especial "Conceptual Graphs Workshop" editado por Eileen C. Way.

[JTC1/SC21/WG3, 1992] ISO/IEC JTC1/SC21/WG3. The IRDS conceptual schema. Working paper, International Organisation for Standardization, 1992.

[Levinson e Ellis, 1993] Robert Levinson e Gerard Ellis, editores. *Proceedings of the Second International Workshop on PEIRCE: A Conceptual Graphs Workbench*, Quebec, Canada, 7 Agosto 1993. Laval University.

[Levinson, 1993] Robert Levinson. Towards domain-independent machine intelligence. In Mineau et al. [1993a], pp. 254–273.

[Levinson, 1994] Robert Levinson. USD: A universal data structure. In Tepfenhart et al. [1994a], pp. 230–250.

[Liquière, 1993] Michel Liquière. Graphs and learning. In Mineau et al. [1993b], pp. 45–61.

[Mackworth e Freuder, 1985] Alan K. Mackworth e Eugene C. Freuder. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial Intelligence*, 25:65–74, 1985.

[McCarthy, 1993] John McCarthy. Notes on formalizing context. In *Proceedings of the Thirteenth IJCAI*, pp. 555–560, Chambéry, France, 29 Agosto — 3 Setembro 1993.

[Menezes, 1995] Francisco Menezes. *Defeasible Constraint Solving*. Tese de Doutoramento, Universidade Nova de Lisboa, 1995. Em preparação.

[Mineau *et al.*, 1993a] Guy W. Mineau, Bernard Moulin, e John F. Sowa, editores. *Conceptual Graphs for Knowledge Representation*, número 699 de Lecture Notes in Artificial Intelligence, Québec City, Canada, 4–7 Agosto 1993. Springer-Verlag. Actas da First International Conference on Conceptual Structures.

[Mineau *et al.*, 1993b] Guy W. Mineau, Bernard Moulin, e John F. Sowa, editores. *International Conference on Conceptual Structures: Theory and Applications*, Québec City, Canada, 4–7 Agosto 1993. Actas complementares.

[Mineau, 1992] Guy W. Mineau. Induction on conceptual graphs: Finding common generalizations and compatible projections. In Nagle et al. [1992], pp. 295–310.

[Mineau, 1994] Guy W. Mineau. Views, mappings and functions: Essential definitions to the conceptual graph theory. In Tepfenhart et al. [1994a], pp. 160–174.

[Moulin e Dumas, 1994] Bernard Moulin e Stephanie Dumas. The temporal structure of a discourse and verb tense determination. In Tepfenhart et al. [1994a], pp. 45–68.

[Moulin, 1993] Bernard Moulin. The representation of linguistic information in an approach used for modelling temporal knowledge in discourse. In Mineau et al. [1993a], pp. 182–204.

[Mugnier e Chein, 1993a] M. L. Mugnier e M. Chein. Polynomial algorithms for projection and matching. In Pfeiffer e Nagle [1993], pp. 239–251.

[Mugnier e Chein, 1993b] M.L. Mugnier e M. Chein. Characterization and algorithmic recognition of canonical conceptual graphs. In Mineau et al. [1993a], pp. 294–311.

[Myaeng *et al.*, 1994] Sung H. Myaeng, Christopher Koo, e Ming Li. Linguistic processing of a text for a large-scale conceptual information retrieval system. In Tepfenhart et al. [1994a], pp. 69–83.

[Nagle *et al.*, 1992] Timothy E. Nagle, Janice A. Nagle, Laurie L. Gerholz, e Peter W. Eklund, editores. *Conceptual Structures: Current Research and Practice.* Ellis Horwood Series in Workshops. Ellis Horwood, 1992.

[Nogier e Zock, 1992] Jean-François Nogier e Michael Zock. Lexical choice as pattern matching. In Nagle et al. [1992], pp. 413–435.

[Pagnucco e Foo, 1993] Maurice Pagnucco e Norman Foo. Inverting resolution with conceptual graphs. In Mineau et al. [1993a], pp. 238–253.

[Perez e Sarris, 1993] S. Perez e A. Sarris, editores. *Information Resource Dictionary System Conceptual Schema.* American National Standards Institute and International Organisation for Standardization, 1993. Documento X3H4/92-003 e ISO/IEC JTC1/SC21 N7486.

[Pfeiffer e Hartley, 1992] Heather D. Pfeiffer e Roger T. Hartley. The conceptual programming environment, CP. In Nagle et al. [1992], pp. 87–107.

[Pfeiffer e Nagle, 1993] Heather D. Pfeiffer e Timothy E. Nagle, editores. *Conceptual Structures: Theory and Implementation*, número 754 de Lecture Notes in Artificial Intelligence. Springer-Verlag, 1993. Actas da Seventh Annual Workshop on Conceptual Graphs, Las Cruces, New Mexico, USA, 8–10 Julho 1992.

[Polovina, 1993] Simon Polovina. Bridging accounting and business strategic planning using conceptual graphs. In Pfeiffer e Nagle [1993], pp. 312–321.

[Rassinoux *et al.*, 1994] A.-M. Rassinoux, R. H. Baud, e J.-R. Scherrer. A multilingual analyzer of medical texts. In Tepfenhart et al. [1994a], pp. 84–96.

[Sabah e Vilnat, 1993] Gérard Sabah e Anne Vilnat. Hierarchy of relational types in conceptual graphs to handle natural language parsing. In Mineau et al. [1993b], pp. 198–215.

[Schmidt e Kocura, 1993] Jan Schmidt e Pavel Kocura. Generalized referents: a neat interface for the scruffy work. In Mineau et al. [1993b], pp. 1–16.

[Schröder, 1993] Martin Schröder. Knowledge based analysis of radiology reports using conceptual graphs. In Pfeiffer e Nagle [1993], pp. 293–302.

[Sowa e Way, 1986] John F. Sowa e Eileen C. Way. Implementing a semantic interpreter using conceptual graphs. *IBM Journal Res. Develop.*, 30(1):57–69, Janeiro 1986.

[Sowa, 1984] John F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine.* The System Programming Series. Addison-Wesley Publishing Company, 1984.

[Sowa, 1992] John F. Sowa. Conceptual graph summary. In Nagle et al. [1992], pp. 3–51.

[Sowa, 1993] John F. Sowa. Relating diagrams to logic. In Mineau et al. [1993a], pp. 1–35.

[Sowa, 1995] John F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. PWS Publishing Company, Boston, 1995. A publicar. Versão de 9 de Agosto de 1994.

[Tepfenhart *et al.*, 1994a] William M. Tepfenhart, Judith P. Dick, e John F. Sowa, editores. *Conceptual Structures: Current Practices — Proceedings of the Second International Conference on Conceptual Structures*, número 835 de Lecture Notes in Artificial Intelligence, College Park MD, USA, 16–19 Agosto 1994. University of Maryland, Springer-Verlag.

[Tepfenhart *et al.*, 1994b] William M. Tepfenhart, Judith P. Dick, e John F. Sowa, editores. *Second International Conference on Conceptual Structures — Proceedings Supplement*, College Park MD, USA, 16–19 Agosto 1994. University of Maryland.

[Tjan *et al.*, 1992] Bosco S. Tjan, David A. Gardiner, e James R. Slagle. Representing and reasoning with set referents and numerical quantifiers. In Nagle et al. [1992], pp. 53–66.

[van Hentenryck *et al.*, 1991] P. van Hentenryck, Y. Deville, e C.-M. Teng. A generic arc consistency algorithm and its specializations. Relatório Técnico RR 91-22, K.U. Leuven, F.S.A., Dezembro 1991.

[Velardi *et al.*, 1988] Paola Velardi, Maria Teresa Pazienza, e Mario De' Giovanetti. Conceptual graphs for the analysis and generation of sentences. *IBM Journal of Research and Development*, 32(2):251–267, Março 1988.

[Way, 1992a] Eileen C. Way. Conceptual graph overview. In JETAI [1992], pp. 75–84.

[Way, 1992b] Eileen C. Way. The dynamic type hierarchy theory of metaphor. In Nagle et al. [1992], pp. 543–557.

[Wermelinger e Bejan, 1993] Michel Wermelinger e Alex Bejan. Conceptual structures for modeling in CIM. In Mineau et al. [1993a], pp. 345–360.

[Wermelinger e Lopes, 1994] Michel Wermelinger e José Gabriel Lopes. Basic conceptual structures theory. In Tepfenhart et al. [1994a], pp. 144–159.

[Wuwongse e Manzano, 1993] Vilas Wuwongse e Mario Manzano. Fuzzy conceptual graphs. In Mineau et al. [1993a], pp. 430–449.

Índice Remissivo