Propriedades da Lógica de Defaults e suas Aplicações à Especificação Formal de Programas

José Evande Carvalho Araujo

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovado por :

Sheila Regina Murgel Veloso, D.Sc.
(Presidente)

Paulo Augusto Silva Veloso, Ph.D.

Marco Antônio Casanova, Ph.D.

Gerson Zaverucha, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
JULHO DE 1992

ARAUJO, JOSÉ EVANDE CARVALHO
Propriedades da Lógica de *Defaults* e suas Aplicações à Especificação Formal de Programas [Rio de Janeiro] 1992.
VIII, 79 p., 29,7cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 1992)
Tese – Universidade Federal do Rio de Janeiro, COPPE.
1 - Especificação Formal de Programas 2 – Lógica de *Defaults*
I. COPPE/UFRJ II. Título (série).

A Evande (pai) e Graça,
Adriana e Eduardo,
Cláudia e Davi.

**** ****

Agradecimentos

À professora Sheila Regina Murgel Veloso, pela orientação competente, dedicada e entusiasta.

Ao professor Guilherme Lincoln Aguiar Ellery, por ter me despertado o amar à Lógica c pelo impulso inicial para o começa desse trabalho.

Ao professor Marcelino Pequeno, pelo material e sugestões fundamentais para o início do estudo.

A todos aqueles, em especial aos colega da COPPE, que de alguma forma contribuiram para a elaboração desta tese,

A Deus, pela luz e inspiração necessárias para a consecução desse objetivo.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau da Mestre em Ciências (M.Sc.)

Propriedades da Lógica de Defaults e suas Aplicações à Especificação Formal de Programas

José Evande Carvalho Araujo

Julho, 1992

Orientadora : Sheila Regina Murgel Veloso
Programa : Engenharia de Sistemas e Computação

Nesta tese, são estudadas propriedades da lógica de *defaults* que permitem sua utilização como linguagem de especificação de programas e possibilitam que ela represente o processo de desenvolvimento de programas por refinamentos sucessivos.

Mostra-se que, ao optar-se pela lógica de *defaults*, ao invés da lógica clássica, linguagem de especificação tradicional, aumenta-se o poder de expressão, pois é possível especificar vários programas em uma única especificação e manipular corretamente erros em programas.

Em seguida, desenvolvem-se definições de extensões expansivas e conservativas e de interpretações nessa situação, de forma que esses conceitos possam ser combinados para representar o passo básico do desenvolvimento de programas. Observa-se que, nesse contexto, só se pode enunciar o teorema da modularização, que garante que uma sequência de refinamentos sucessivos dá origem à implementação desejada, sob certas restrições.

Finalmente, desenvolvem-se maneiras de se obter extensões conservativas de e interpretações entre teorias com *defaults*.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

Properties of Default Logic and its Applications ta the Formal Specification of Programs

José Evande Carvalho Araujo

July, 1992

Thesis Supervisor : Sheila Regina Murgel Veloso
Department : Computing and Systems Engineering

This work is mainly concerned with the study of properties of default logic, in a way that it can be used as a language for specifying programs, and that it is possible to use it to represent the process of program development by stepwise refinements.

It is shown that, if default logic replaces classical logic, which is the tradicional specification language, the expressive power is increased, in the sense that several programs can be specified in a unique specification and program errors can be handled properly.

Definitions of expansive and conservative extensions and of interpretations are developed for this logic, in a way that these concepts can be combined to represent the canonical step of the program development process. It is proved that, in this context, one can state the modularization theorem, which guarantees that a sequence of successive refinements originates the desired implementation, under some restrictions.

Finally, forms of obtaining conservative extensions of and interpretations between default theories are proposed.

Índice

Capítulo I

Introdução

Neste trabalho, estudamos conceitos de conservatividade e de interpretação entre teorias no contexto da lógica de *defaults*. Esta pesquisa foi motivada pelo interesse em capacitar tal lógica para trabalhar como linguagem de especificação de programas e para representar o processo de desenvolvimento de programas por refinamentos sucessivos, já que o uso da lógica de *defaults* nessa tarefa, ao invés da lógica de primeira ordem, aumenta o poder de especificação de programas. Estudamos, também, métodos de se obter extensões conservativas de e interpretações entre teorias com *defaults*.

A lógica de primeira ordem tem sido tradicionalmente usada como linguagem de especificação de programas. Isso se deve, primeiro, à sua natureza formal, capaz de expressar, sem ambigüidades e excessos de detalhes, o sentido desejado para um programa, e, segundo, à possibilidade de representar com precisão o processo de desenvolvimento de programas por refinamentos sucessivos.

O processo de desenvolvimento de programas por refinamentos sucessivos divide a implementação de programas em várias etapas menores, fáceis de se manipular, e a posterior composição dessas etapas para formar a implementação desejada. Assim, partindo da especificação abstrata do programa, ruma-se para um programa pronto para rodar em uma máquina existente, passando antes por várias etapas intermediárias, cada vez mais próximas da versão final. Através dos conceitos de extensões conservativas e de interpretações, a lógica clássica reflete a natureza de cada um desses passos de desenvolvimento. Além disso, quando encadeamos vários desses passos, esses conceitos se combinam de tal forma que, de imediato, obtemos uma implementação da especificação inicial na final.

Entretanto, não conseguimos, com a lógica de primeira ordem, lidar apropriadamente com erros em programas.

A motivação de nosso estudo se deu quando percebemos que a lógica de *defaults* permite um tratamento adequado de erros e exceções. Além disso, o poder de especificação de programas aumenta, na medida em que é possível especificar vários programas com uma única especificação.

Para habilitar adequamente a lógica de *defaults* para a tarefa de especificar programas, era preciso capacitá-la para representar o processo de desenvolvimento de programas por refinamentos sucessivos. Para isso, definimos conceitos de extensões conservativas r: de interpretação, nesse contexto, capazes de representar a passo básico de desenvolvimento. Preocupamo-nos em garantir que nossas definições possibilitassem que, dada uma sequência desses passos, de imediato se derivasse um passo Única que expressasse a implementação da especificação inicial na final.

Contudo, nossas definições não são tão simples e não permitem que, dadas duas teorias com *defaults*, se perceba facilmente se uma é extensão conservativa da outra ou se uma pode ser interpretada na outra. Por isso, pesquisamos maneiras de garantir a existência de extensões conservativas de e de interpretações entre teorias com *defaults*.

Dividimos essa tese da seguinte maneira: nos dois próximos capítulos, fazemos uma revisão dos conceitos básicos usados em nosso trabalho; nos dois capítulos seguintes, apresentamos nosso estudo; e, no capítulo final, fazemos uma recapitulação dos resultados obtidos, sugerimos extensões do trabalho e apresentamos nossas conclusões. Finalmente, em um apêndice, mostramos uma prova alternativa para uma proposição de um capítulo anterior.

No capítulo 11, fazemos, inicialmente, uma rápida revisão da lógica de primeira ordem. Citando apenas as definições mais conhecidas, detemo-nos em certos conceitos não muito difundidas, como o de extensões conservativas e o de interpretações. Em relação às extensões conservativas, mostramos maneiras de, por construção, garantir que uma teoria é extensão conservativa de outra. Depois, discutimos como esses conceitos se combinam para representar a passo básico da processo de desenvolvimento de programas, o passo canônico, e mencionamos o teorema que garante que uma sequência desses passos dá origem a um único passo, o teorema da modularização. Encerramos o capitulo discorrendo, rapidamente, sobre as peculiaridades da especificação de tipos abstratos de dados com a lógica clássica.

No capítulo III, revisamos tópicos relacionados com a lógica de *defaults*. Diferentemente do capítulo anterior, não abordamos somente temas que usaremos na discussão posterior. Longe de pretendermos elaborar uma revisão total do assunto, apresentamos uma série de itens referentes a diversos aspectos dessa lógica, com o intuito de proporcionar uma visão geral, apesar de incompleta, da bibliografia existente. Entre esses itens, incluímos os principais teoremas referentes às teorias com *defaults* arbitrários fechados e às teorias com *defaults* normais fechados, uma condição suficiente para a coerência de teorias com *defaults* semi-normais, um procedimento que constrói extensões para teorias com *defaults* finitas, e o estudo de uma semântica para a lógica de *defaults*.

No capítulo IV, inicialmente, discorremos sobre a motivação do nosso estudo. Para isso, apresentamos situações em que a utilização da lógica de *defaults* amplia o poder de especificação em relação à lógica clássica. Em seguida, introduzimos os conceitos de extensões monotônicas restritas, de extensões conservativas restritas (e da sua contrapartida semântica, as extensões expansivas restritas), e de interpretações *default*. Logo após, mostramos como eles se unem para representar o passo básico do processo de desenvolvimento de programas por refinamentos sucessivos. Concluímos o capítulo discutindo uma versão para o teorema da modularização nesse contexto.

No capítulo V, estudamos uma maneira de estender uma teoria com *defaults* de forma que o resultado seja uma extensão conservativa restrita desta, e uma maneira de traduzir uma teoria com *defaults* em outra de modo que exista uma interpretação *default* da primeira na segunda. Nessas condições específicas, provamos o teorema da modularização. Encerramos o capítulo com um exemplo que ilustra as técnicas desenvolvidas.

Capitulo II

Especificação de Programas com a Lógica Clássica

Nesse capítulo estudaremos como a lógica de primeira ordem é usada para especificar formalmente programas, e quais conceitos lógicos estão por trás do processo de desenvolvimento de programas por refinamentos sucessivos.

II.1 Conceitos de lógica de primeira ordem

Nessa seção apresentaremos alguns conceitos básicos da lógica de primeira ordem, especialmente o de extensão de teorias, com ênfase em extensões conservativas, e o de interpretação entre teorias. Maiores detalhes podem ser encontrados na literatura [1,2,3,4].

Usamos as definições usuais de teoria de primeira ordem, de estrutura para uma linguagem, de modelos de uma teoria e de consequência lógica ($Th(S)$ denotará o conjunto de consequências lógicas do conjunto de fórmulas S).

Consideramos uma linguagem L como as fórmulas bem formadas (wffs) usuais construídas sobre um alfabeto A que consiste de símbolos lógicos ($\neg$, h, $\vee$, $\rightarrow$, $\leftrightarrow$), dos quantificadores ($\forall$ e $\exists$), do símbolo de igualdade, de variáveis e de símbolos de predicado e de função. Dizemos que uma linguagem L é uma sublinguagem de L', denotado por L $\subseteq$ L', se e somente se o alfabeto de L está contido no alfabeto de L'. Estender uma linguagem E significa introduzir novos símbolos no seu alfabeto, o que será representado por L + f, onde f é o novo símbolo do alfabeto.

Chamaremos as fórmulas bem formadas de wffs ou apenas de fórmulas. Se S é um conjunto de axiomas da teoria T sobre linguagem L, i.e., se $Th(S) = T$, o par $<$ L, $S >$ é chamado de apresentação da teoria T. Quando a linguagem for conhecida?chamaremos apenas S de apresentação da teoria,

Consideramos que uma estrutura $\mathcal{A}'$ para uma linguagem L' é uma expansão de uma estrutura $\mathcal{A}$ para uma linguagem L, se $L \subseteq L'$ e $\mathcal{A}$ e $\mathcal{A}'$ concordam nas atribuições dadas aos símbolos do alfabeto de L.

Dizemos que a teoria T' sobre a linguagem L' com um conjunto de axiomas S' é uma *extensão clássica* [1] da teoria T sobre linguagem L com um conjunto de axiomas S, representado por $T \subseteq T'$, se e somente se $L \subseteq L'$ e $Th(S) \subseteq Th(S')$. Dizemos que $T \subseteq T'$ [2] *é conservativa clássica*, representado por $T \preceq T'$, se e somente se para toda fórmula F de L, quando $T' \models F$, então $T \models F$. $T \subseteq T'$ é *expansiva clássica* se e somente se todo modelo de T puder ser expandido para um modelo de T'. Usaremos a notação $T' = T + A$ quando a teoria T' for obtida através da adição da fórmula A à teoria T, i.e., $T' = Th(T \cup \{A\})$.

A proposição a seguir de [1, p. 65, exercício 3.b], dá uma condição suficiente para a conservatividade.

Proposição II.1 *Se a extensão $T \subseteq T'$ é expansiva clássica, então ela é conservativa clássica.*

O fato da expansividade não ser uma condição necessária para a conservatividade é discutido por VELOSO e VELOSO [5].

Apesar de o conceito de extensão conservativa clássica ser bastante importante no nosso estudo, não é evidente, apenas pela apresentação de duas teorias de primeira ordem, se uma é extensão conservativa clássica da outra. Por isso, é interessante conhecermos métodos para estender teorias clássicas, de forma que o resultado dessa operação seja, garantidamente, uma extensão conservativa clássica da teoria original. Um caso particular de bastante importância é o de se estender uma teoria por adição de um novo símbolo de função conservativamente.

Usaremos $\underline{x}$ para denotar uma lista de variáveis. $F(\underline{x}, y)$ diz que as variáveis livres da fórmula F são $\underline{x}$ e y.

Dada uma fórmula F de uma linguagem L com $n+1$ variáveis livres, F denota uma relação (n+1)-ária em uma estrutura para L. O que desejamos fazer é "extrair" uma função n-ária dessa relação (n+1)-ária. Mais precisamente, temos uma teoria T com linguagem L e desejamos estendê-la para uma teoria $T' = T + A$ sobre uma linguagem $L' = L + f$, onde A é um novo axioma obtido de F e f é um novo símbolo de função n-ária. Estamos interessados em saber quando esta extensão é conservativa clássica.

Quando o novo axioma A define explicitamente o novo símbolo em termos dos antigos, temos uma extensão por definição clássica de símbolo funcional.

[1] O adjetivo 'clássico' será aplicado aos conceitos tradicionais da lógica de primeira ordem durante todo o texto para diferenciá-los das conceitos equivalentes que definiremos para a lógica de *defaults*.

[2] $\subseteq$ representará extensões clássicas e denotará que uma linguagem é sublinguagem de outra, além da sua função tradicional de representar que um conjunto é subconjunto de outro, o sentido ficando claro pelo contexto.

Chamamos $T' = T + A$ uma *extensão por definição clássica de símbolo funcional* se e somente se o axioma A é da forma

$$A : (\forall \underline{x})(\forall y)[y = f(\underline{x}) \leftrightarrow F(\underline{x}, y)], \text{ onde } F \text{ e } f \text{ são como acima.}$$

Na verdade essa definição de extensão por definição clássica está incompleta em relação à definição usual na literatura. É que se preferiu ressaltar as condições de existência e de unicidade como condições fundamentais para a conservatividade, como veremos a seguir.

Considere as fórmulas $\exists$ e U chamadas, respectivamente, de condições de *existência* e de *unicidade*:

$$\begin{array}{ll} \exists : & (\forall \underline{x})(\exists y)F(\underline{x}, y); \\ U : & (\forall \underline{x})(\forall y', y'')[F(\underline{x}, y') \wedge F(\underline{x}, y'') \rightarrow y' = y'']. \end{array}$$

Elas garantem que a relação 'extraída' da fórmula F é uma função total.

A proposição a seguir (proposição 3.2 de VELOSO e VELOSO [4]) nos dá as condições necessárias e suficientes para a conservatividade das extensões por definição clássica de símbolo funcional.

Proposição II.2 *Considere uma extensão $T \subseteq T'$ por definição clássica de símbolo funcional. As afirmativas seguintes são equivalentes:*

1. *$T \subseteq T'$ é conservativa clássica;*
2. *$\exists$ e U são consequências lógicas de T;*
3. *Todo modelo de T tem uma única expansão para um modelo de T'.*

A proposição acima nos mostra que, se garantirmos que as condições de existência e unicidade para a fórmula F valem na teoria inicial, a extensão clássica dessa teoria com o novo axioma que define o novo símbolo de função em termos de F é uma extensão conservativa clássica.

Entretanto, as condições de existência e de unicidade são bastante restritivas, pois exigem que a relação denotada por F já seja uma função total. Isso se deve ao bicondicional do axioma A. Estamos interessados no caso mais geral de termos liberdade de escolher a função a ser extraída da relação. Em termos do processo de desenvolvimento formal de programas, que é o motivo básico de nosso estudo, essa escolha da função a ser extraída equivale a decisões de projeto ao se passar da especificação para um programa.

VELOSO e VELOSO [4] estudam várias maneiras menos restritivas de se introduzir símbolos de função em teorias. Apresentaremos aqui os casos em que o novo axioma força que o novo símbolo de função denote uma função contida na relação denotada por F, ou uma função que inclua a relação denotada por F, e não mais a função idêntica à F.

Chamamos $T' = T + A$ uma *extensão por restrição exterior (do símbolo funcional f)* se e somente se o axioma A é da forma

$$A : (\forall x)(\forall y)[y = f(x) \rightarrow F(x, y)]$$

A proposição a seguir (proposição 4.1 de VELOSO e VELOSO [4]) ressalta que agora somente a condição de existência é a condição necessária e suficiente para a conservatividade.

Proposição II.3 *Considere uma extensão $T \subseteq T'$ por restrição exterior. As afirmativas seguintes são equivalentes:*

1. *$T \subseteq T'$ é conservativa clássica;*
2. *$\exists$ é consequência lógica de T;*
3. *Todo modelo de T tem pelo menos uma expansão para um modelo de T'.*

Esse novo axioma A requer que o novo símbolo de função denote uma função contida na relação denotada por F.

Chamamos $T' = T + A$ uma *extensão por restrição interior (do símbolo funcional f)* se e somente se o axioma A é da forma

$$A : (\forall x)(\forall y)[F(x, y) \rightarrow y = f(x)]$$

A proposição a seguir (proposição 4.2 de VELOSO e VELOSO [4]) ressalta que agora somente a condição de unicidade é a condição necessária e suficiente para a conservatividade.

Proposição II.4 *Considere uma extensão $T \subseteq T'$ por restrição interior. As afirmativas seguintes são equivalentes:*

1. *$T \subseteq T'$ é conservativa clássica;*
2. *U é consequência lógica de T;*
3. *Todo modelo de T tem pelo menos uma expansão para um modelo de T'.*

Esse novo axioma A requer que o novo símbolo de função denote uma função que inclua a relação denotada por F.

A relação de conservatividade é transitiva. Isto é, se T1, T2 e T3 são teorias de primeira ordem e se temos $T1 \leq T2$ e $T2 \leq T3$, então temos $T1 \leq T3$.

Um outro conceito bastante importante para o tópico de especificações formais se refere à tradução de uma teoria em outra, preservando as propriedades: o conceito de *interpretação clássica* entre teorias, o equivalente, para teorias de primeira ordem, aos homomorfismos entre estruturas matemáticas.

Definiremos o que vem a ser interpretações entre teorias T e T' em três etapas: primeiro definiremos interpretações entre duas linguagens L e L', depois entre uma linguagem L e uma teoria T', e finalmente entre duas teorias T e T'.

Uma *interpretação clássica* **I** *de L* em L', representada por $I : L \rightarrow L'$, consiste de:

- um símbolo de predicado unário N de L', chamado de *predicado de relativização* para I;
- um mapeamento π dos símbolos de função e predicado de L nos de L', onde cada símbolo de função n-ário de L é mapeado em um símbolo de função n-ário de L', e cada símbolo de predicado n-ário de L é mapeado em um símbolo de predicado n-ário de L'.

Usando o mapeamento π, podemos definir a tradução dos termos formados com o alfabeto de L em termos formados com o alfabeto de L', denotada por I. As fórmulas de L podem ser traduzidas nas de L' (chamaremos a tradução também de I), da seguinte maneira: seja A uma fórmula de L, então

$$A^I = ((N(x_1) \wedge \ldots \wedge N(x_n)) \rightarrow A^{trI})$$

onde $x_1, \ldots, x_n$ são as variáveis livres em A e trI h definido abaixo indutivamente na estrutura das fórmulas:

$$trI : L \mapsto L'$$

1. Se A é uma fórmula atômica da forma $r(t_1, \ldots, t_n)$, onde r é um símbolo de predicado do alfabeto de L e $t_1, \ldots, t_n$ são termos de L, então

$$(r(t_1, \ldots, t_n))^{trI} = \pi(r)((t_1)^I, \ldots, (t_n)^I);$$

2. Se A tem a forma de um dos esquemas de fórmulas abaixo da esquerda, sua tradução é dada da seguinte maneira:

$$\begin{aligned}
(B \vee C)^{trI} &= B^{trI} \vee C^{trI} \\
(B \wedge C)^{trI} &= B^{trI} \wedge C^{trI} \\
(B \rightarrow C)^{trI} &= B^{trI} \rightarrow C^{trI} \\
(B \leftrightarrow C)^{trI} &= B^{trI} \leftrightarrow C^{trI} \\
(\neg B)^{trI} &= \neg((B)^{trI})
\end{aligned}$$

3. Se A é da forma $\forall xB$ ou $\exists xB$, então:

$$\begin{aligned}(\forall xB)^{trI} &= \forall x(N(x) \rightarrow B^{trI})\\ (\exists xB)^{trI} &= \exists x(N(x) \wedge B^{trI})\end{aligned}$$

Note-se que os quantificadores e o domínio das variáveis livres da fórmula A são relativizadas quando traduzidos. Isso se deve ao fato dessas variáveis e dos quantificadores se referirem a todo o domínio inicial de L, mas, quando traduzidos, deverem se referir a apenas um subconjunto de L', especificado pelo predicado de relativização.

Devemos, também, perceber que **I** induz estruturas no sentido inverso. Seja A' uma estrutura para L'. A estrutura $\mathcal{A}$ para L induzida via **I**, denotada por $I^{-1}(\mathcal{A}')$ é a seguinte:

- domínio de $\mathcal{A}$ = domínio de $\mathcal{A}'$;

  $\mathcal{A}(f) = \mathcal{A}'(\pi(f))$, para cada símbolo de função f do alfabeto de L;
- $\mathcal{A}(p) = \mathcal{A}'(\pi(p))$, para cada símbolo de predicado p do alfabeto de L.

Uma *interpretação clássica* $I : L \rightarrow T'$ de uma linguagem L em uma teoria T' com linguagem L' é uma interpretação $I : L \rightarrow L'$, onde

$$\begin{aligned}T' &\models \exists xN(x) \text{ e}\\ T' &\models (N(x_1) \wedge \ldots \wedge N(x_n)) \rightarrow N(\pi(f)(x_1, \ldots, x_n))\\ &\quad \text{para cada símbolo de função } f \text{ do alfabeto de L.}\end{aligned}$$

A primeira restrição requer que o domínio definido pelo predicado de relativização seja não vazio, e a segunda que $\pi(f)$ represente uma função cuja restrição ao domínio do predicado de relativização tome valores que também satisfaçam tal predicado. Em outras palavras, a segunda restrição diz que o domínio de valores definidos pelo predicado de relativização é fechado sob funções traduzidas.

Finalmente, uma *interpretação clássica* $I : T \rightarrow T'$ entre teorias T e T' sobre linguagens L e L', respectivamente, é uma interpretação $I : L \rightarrow T'$ tal que, para cada fórmula A tal que $T \models A$, temos $T' \models A^I$.

Um resultado interessante é dado pelo próximo teorema.

Teorema II.1 (Teorema da Interpretação) *Sejam T e T' teorias de primeira ordem cujas apresentações de teorias são $< L, S >$ e $< L', S' >$, respectivamente. Se $I : L \rightarrow T'$ é uma interpretação clássica de L em T', então I é uma interpretação clássica de T em T' se e somente se para cada $A \in S$, $T' \models A^I$.*

Logo, para garantirmos que $I : L \to T'$ é uma interpretação clássica entre T e T', basta mostrarmos que as traduções dos axiomas de T são consequências de T'.

As interpretações clássicas possuem uma propriedade bastante importante: se I é uma interpretação clássica de T em T', então para cada modelo A' de T', $I^{-1}(A')$ é um modelo de T. Isto é, as interpretações clássicas garantem que modelos da segunda teoria induzem modelos para a primeira teoria.

Definiremos, agora, como se compor interpretações clássicas.

Sejam T, T', T'' teorias de primeira ordem. Considere as interpretações $I : T \to T'$ e $I' : T' \to T''$, a primeira definida em termos do predicado de relativização $N(x)$ e mapeamento π, e a segunda em termos do predicado de relativização $N'(x)$ e mapeamento π'. A *composição* $I \cdot I'$ de I e I' é definida como:

1. O predicado de relativização para $I \cdot I'$ é definido como $N'(x) \wedge \pi'(N)(x)$. Aqui desejamos afrouxar nossa definição inicial de predicado de relativização, que exigia que ele fosse um símbolo de predicado unário;
2. O mapeamento das símbolos de L nos de L'' é $\pi \cdot \pi'$ (a composição de π e π');

A composição de interpretações é associativa. Isto é, dadas $I0 : T0 \to T1$, $I1 : T1 \to T2$ e $I2 : T2 \to T3$, onde Ii e Ti, para $i = 0, 1, 2$, são interpretações clássicas e teorias de primeira ordem, respectivamente, temos:

$$I0 \cdot (I1 \cdot I2) = (I0 \cdot I1) \cdot I2.$$

II.2 Especificação formal de programas

Os engenheiros de *software* têm tido a preocupação de enumerar características desejáveis de um programa: ele deve ser eficiente, robusto, portátil, fácil de modificar, correto, *etc.* De todas, nos parece que uma delas é a mais fundamental: a corretude (esse é o ponto de vista de TURSKI e MAIBAUM [3]). Afinal, não há nenhum interesse em um programa que não execute corretamente as tarefas a ele destinadas.

Mas qual é o real significado da palavra *correto*? Em primeiro lugar, esse termo não tem sentido se não estiver acompanhado de um esquema de referência, pois algo é *correto* se satisfizer algum critério que é definido independentemente desse algo.

Quando nos referimos a programas, existem dois critérios de corretude, cada qual associado a um esquema de referência diferente: a corretude *sintática* e a corretude *semântica*.

Um programa está *sintaticamente correto* se satisfizer às regras gramaticais da linguagem de programação sobre a qual está definido. Aqui, o esquema de referência é a gramática da linguagem.

Um programa está *semanticamente* correto se ele expressa um sentido declarado *a priori*. A esse sentido pretendido para um programa chamaremos de *especificação do programa*. Essa especificação é definida através de um conjunto de sentenças que expressam o que o programa deve fazer. A especificação será as consequências desse conjunto de sentenças.

Os termos 'conjunto de sentenças' e 'consequências' indicam a presença de um sistema formal subjacente. De fato, é desejável que as especificações sejam expressas de forma não-ambígua, o que torna necessário que sejam especificadas através de um aparato formal. No contexto de especificação de programas, a lógica de primeira ordem poli-sortida é bastante usada como linguagem de especificação. Nesse trabalho, consideraremos nossas especificações de programas como teorias de primeira ordem (não-polisortidas), especificadas por apresentações de teorias, sendo a extensão para a lógica clássica poli-sortida relativamente simples.

Um ponto a se observar é que um programa de computador também pode ser representado através de um conjunto de sentenças, apesar de estarmos acostumados a representá-lo descrevendo sua execução. Assim, determinar a corretude semântica se limita a comparar duas teorias de primeira ordem: a especificação do programa e a teoria que representa o programa de computador. Essa comparação pode ser evitada através do uso de metodologias de desenvolvimento de programas que garantam a corretude semântica.

A especificação do programa serve para dois propósitos: como esquema de referência para determinar a corretude semântica no processo de desenvolvimento, implementação e verificação do programa, e para representar o próprio programa (desde que este esteja semanticamente correto em relação à especificação), uma vez que é mais sucinta e abstrata, sendo, portanto, mais fácil de entender e modificar.

II.2.1 Passo canônico

Após essa introdução, vamos agora examinar o processo de desenvolvimento de programas. No contexto acima discutido, esse processo consiste em, partindo da especificação inicial, chegar ao programa pronto para ser rodado em um computador existente. Entretanto, os manuais de engenharia de *software* rezam que essa transformação da especificação inicial no programa final não deve se dar de uma vez, mas passo-a-passo, por refinamentos sucessivos, onde em cada refinamento chegamos mais próximos do programa final.

Dessa forma, o processo de desenvolvimento de programas consiste na repetição de um mesmo passo básico, sendo que esses pequenos passos podem ser fácil e mecanicamente compostos. Os advérbios 'fácil' e 'mecanicamente' não são gratuitos. Afinal, a utilidade desses passos pequenos é podermos dividir a parte criativa do processo de desenvolvimento em partes menores, fáceis de se manipular. Se após essas etapas individuais tivéssemos que usar de criatividade para compô-las na solução final, elas não seriam de muita valia.

Em cada passo básico lidamos com especificações. Partimos da especificação inicial do programa e, eventualmente, chegamos à especificação do programa final, passando por várias especificações intermediárias.

A seguir, estudaremos os conceitos matemáticos embutidos nesse processo de desenvolvimento de programas por refinamentos sucessivos.

Em cada passo do processo de desenvolvimento de programas, mais detalhes são acrescentados à especificação para torná-la executável em uma máquina existente. Isso é de certo modo o oposto do que fazemos quando criamos a especificação inicial do programa: ignoramos detalhes desnecessários, 'abstraímos' da realidade somente aquilo que é importante para descrever o problema.

Sejamos mais formais. Sejam S e $S1$ especificações (teorias de primeira de ordem (ou apenas suas apresentações) sobre linguagens L e L1, respectivamente). Uma implementação de S em $S1$ é o par $< I1, e1 >$, onde $e1 : Th(S1) \leq Th(S1')$ é uma extensão conservativa clássica e $I1 : Th(S) \to Th(S1')$ é uma interpretação clássica entre teorias. Em forma de diagrama:

$$\begin{tikzcd} S \arrow[r, "I1"] \arrow[dr, "< I1, e1 >"'] & S1' \\ & S1 \arrow[u, "e1"'] \end{tikzcd}$$

Intuitivamente, essa definição vem do fato de que implementar X em termos de Y consiste de duas partes: representar os dados de X em termos dos dados de Y e definir as operações de X em termos das operações de Y. Essa representação (definição) pode não ser direta, daí a necessidade de acrescentar a Y símbolos e procedimentos para que possa representar as operações e objetos de X. No entanto, esse acréscimo não deve interferir nas propriedades dos objetos de Y, daí a necessidade da conservatividade dessa extensão de Y, que denominaremos Y'. Através dessa extensão conservativa, pode-se, também, acrescentar novos detalhes a respeito desses novos símbolos. Os objetos de Y' que podem representar os objetos de X são delimitados pelo predicado de relativização associado à interpretação. A correspondência entre os objetos de X e os de Y' é dada pelo mapeamento π associado à interpretação.

Em termos de modelos, podemos pensar da seguinte maneira: os modelos que realmente existem são os de $S1$ (eventualmente um programa de computador). A extensão expansiva[3] garante que esses modelos fazem parte de modelos

[3]Toda extensão expansiva é conservativa, mas o contrário nem sempre vale. Portanto, queremos mesmo que $e1$ seja expansiva.

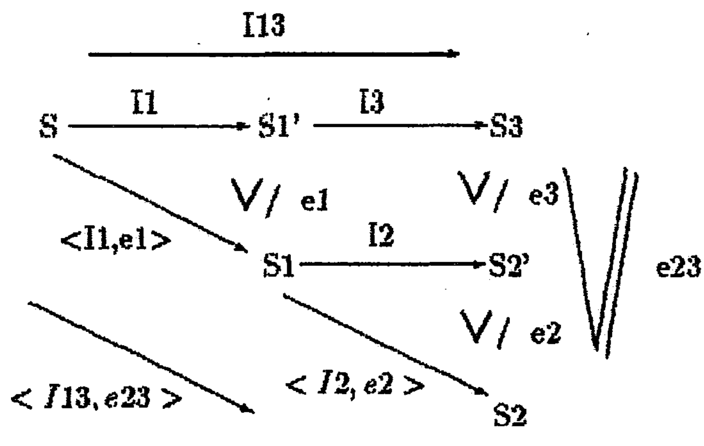

Figura II.1: Implementação de S em $S2$

de $S1'$. A interpretação $I1$ assegura que esses modelos de $S1'$ induzem modelos para, S. Assim, cada modelo de $S1$ dá origem a um modelo de S.

Como já foi extensivamente afirmado, o passo canônico pode ser aplicado várias vezes. Por exemplo, $S1$ pode não ser o programa final, mas apenas um estágio intermediário do processo de desenvolvimento do programas. Logo, implementamos $S1$ sobre $S2$ com a implementação $< I2, e2 >$, onde $e2 : S2 \leq S2'$ e $I2 : S1 \rightarrow S2'$. Entretanto, como já mencionamos anteriormente, temos interesse em que essas duas implementações separadas valham apenas por uma, a implementação de S sobre $S2$. Mais uma vez ressaltamos que essa implementação final deve resultar diretamente das duas anteriores. O teorema da modularização [3] garante a existência de um $S3$ como mostrado na figura II.1. Podemos compor i 1 e $I3$ obtendo uma interpretação clássica $I13 : S \rightarrow S3$ e, lembrando que a relação de conservatividade é transitiva, obter a extensão conservativa $e23 : S2 \leq S3$, Assim, obtemos a implementação $< I13, e33 >$ de S em $S2$.

II.2.2 Especificação de tipos abstratos de dados

VELOSO [6] usa a teoria acima na especificação de tipos abstratos de dados. Um tipo abstrato de dados (TAD) é uma forma de abstração de dados que se baseia no 'princípio da invisibilidade da informação' de Parnas, que prega que as estruturas de dados que representam os objetos do programa devem ficas escondidas em um módulo, sendo acessadas apenas por um elenco de operações fornecidas. Para que os TAD's possam ser implementados em linguagens de programação, estas precisam garantir *encapsulamento*, que permite que a representação doe objetos e a implementação das operações abstratas sejam reunidas em uma única região do programa, e *proteção* que determina quais operações podem manipular os objetos do TAD, garantindo uma disciplina de manipulação.

Nesse contexto, as abstrações são classificadas não por suas similaridades de estruturas de representação, mas por similaridades no seu comportamento observável. Um mesmo TAD pode ter duas implementações diferentes, mas isso não fica aparente para quem o usa. Esse fato determina uma fatoração do programa desenvolvido em duas partes: o *programa abstrato*, que manipula apenas operações abstratas sobre objetos abstratos e o *módulo* de *implementação*, que contém a representação dos objetos e a codificação das operações. O primeiro está mais perto do nível do problema, enquanto o segundo está mais próximo da máquina existente.

Essa independência entre programa abstrato e módulo de implementação traz uma série de características desejáveis: *legibilidade*, pois o programa abstrato e o módulo de implementação podem ser entendidos e verificados isoladamente, *modificabilidade*, pois qualquer alteração nas estruturas de representação se localizam no módulo de implementação, não afetando o programa abstrato, e *portabilidade*, pois se desejarmos transportar o programa abstrato para outra máquina, as alterações necessárias estão apenas na parte do módulo de implementação que depende da máquina.

Aqui, também, especificaremos nossos programas com a lógica de primeira ordem. A especificação do programa abstrato chamaremos de PA (lembrar que um programa de computador pode ser representado através de um conjunto de sentenças). Em relação ao módulo de implementação, este deve ser suportado por um tipo concreto, disponível nw máquina, cuja especificação chamaremos de CONCR. A especificação de módulo de implementação, que chamaremos de implementação declarativa (ID), deve ser uma extensão conservativa clássica de CONCR (CONCR $\leq$ ID), pois os novos elementos de ID usados para suportar as operações do programa abstrato devem respeitar (não acrescentar e não modificar) as características do tipo concreto.

Assim, a implementação do programa abstrato sobre o tipo concreto se dá via uma interpretação clássica do PA em ID, que por sua vez é uma extensão conservativa clássica de CONCR.

$$\begin{tikzcd} PA \arrow[r, dash] \arrow[dr] & ID \arrow[d, phantom, "\leq"] \\ & CONCR \end{tikzcd}$$

O teorema da modularização garante que podemos repetir sem medo o passo acima, i.e., se por acaso o tipo concreto não estiver disponível na máquina, mas sim nm outro tipo, cuja especificação chamaremos de DISP, basta construirmos urna implementação de CONCR em BISP que teremos diretamente uma implementação de PA em DISP.

Concluímos o capitulo enfatizando um pontu já comentado: devido à importância das extensões conservativas clássicas nesse contexto, é necessário se caracterizar mais formas de se estender uma teoria de primeira urdem de forma que o resultado seja uma extensão conservativa clássica da teoria original, ampliando ainda mais as possibilidades da lógica clássica como linguagem de especificação.

Capítulo III

Lógica de Defaults

Esse capítulo se dedica ao estudo da lógica de *defaults*. Depois de comentários preliminares, apresenta-se propriedades de teorias com *defaults* fechados. Em seguida, passa-se ao estudo de propriedades de teorias com *defaults* normais e semi-normais fechados. Dentro desse último tópico, apresenta-se uma condição suficiente para a coerência de teorias com *defaults* semi-normais. Depois, um procedimento que constrói extensões *defaults* para teorias com *defaults* finitos é estudado. O capítulo termina com uma semântica para a lógica de *defaults*.

A lógica de *defaults* foi proposta por REITER [7] e visa capturar o raciocínio que se enquadra no seguinte padrão:

- 'Tipicamente o fato A vale', ou
- 'Na ausência de informação em contrário, o fato A vale'.

Esse raciocínio é obviamente não-monotônico, já que podemos inicialmente inferir A e, posteriormente, com o acréscimo de alguma informação que negue A, retirar essa conclusão.

Esse modelo de raciocínio é capturado, na lógica de *defaults*, através dos *defaults*. Um *default* é uma expressão da forma $\frac{\alpha(\underline{x}):\beta_1(\underline{x}),\ldots,\beta_m(\underline{x})}{\omega(\underline{x})}$, onde $\alpha(\underline{x})$, chamada de *pré-requisito*; $\beta_1(\underline{x}),\ldots,\beta_m(\underline{x})$, chamadas de *justificativas*, e $\omega(\underline{x})$, chamada de *consequente*, são fórmulas de primeira ordem cujas variáveis livres estão entre as de $\underline{x}$. Um *default* é fechado se e somente se $\alpha,\beta_1,\ldots,\beta_m,\omega$ são fórmulas fechadas, i.e., sem variáveis livres.

Um *default* da forma $\frac{\alpha(\underline{x}):\omega(\underline{x})}{\omega(\underline{x})}$ é chamado de *normal*. Entre os *defaults* não-normais, os de forma $\frac{\alpha(\underline{x}):\beta(\underline{x})\wedge\omega(\underline{x})}{\omega(\underline{x})}$ são de especial interesse e são chamados de *semi-normais*. LUKASZEWICZ [8] argumenta que o uso de *defaults* semi-normais parece cobrir os casos práticos existentes (isso nos parece verdade quando a tarefa é formalizar o raciocínio do senso comum). ETHERINGTON [9] afirma não ser conhecida nenhuma aplicação que requeira *defaults* com múltiplas justificativas.

Uma teoria com *defaults* é uma teoria de primeira ordem acrescida de regras de inferência (os *defaults*). Essas regras diferem daquelas da lógica clássica, pois se referem tanto ao que já foi quanto ao que não foi provado,

Os defaults 'mapeiam' a teoria de primeira ordem inicial em várias teorias de primeira ordem, as extensões *default*,[1] que representam os vários conjuntos de crenças possíveis de se ter a partir do mundo especificado incompletamente com a teoria, de primeira ordem inicial. Semanticamente, os *defaults* restringem os modelos da teoria de primeira ordem subjacente de uma forma que cada classe restrita de modelos satisfaz às wffs de uma extensão *default* e que cada, extensão *default* tem todas as suas wffs satisfeitas por uma dessas classes restritas de modelos.

Formalmente, uma teoria com *defaults* é um par (D,W), definido sobre uma linguagem L. L são as fórmulas bem formadas usuais construídas sobre um alfabeto A formado por um número contável de variáveis, de símbolos de função e de símbolos de predicado, e pelos símbolos lógicos e quantificadores padrões. Inicialmente consideraremos esse alfabeto e a linguagem constantes. D é um conjunto de *defaults* e W um conjunto de wffs fechadas.

Vamos interpretar um *default* aberto (cujos componentes possuem variáveis livres) como um conjunto de *defaults* fechados obtido pela substituição das suas variáveis livres por termos básicos. Assim, para cada teoria com *defaults* abertos, obtemos uma teoria com *defaults* fechados. Se o conjunto de termos básicos é infinito, obtemos uma teoria com um conjunto infinito de *defaults*. Na teoria que se segue, consideraremos apenas teorias com *defaults* fechados, apesar da maioria dos resultados valerem também para teorias com *defaults* abertos. Maiores detalhes sobre teorias com *defaults* abertos são fornecidos por REITER [7].

III.1 Teorias com *defaults* fechados

Como já mencionamos anteriormente, chamaremos de *extensões default* os vários conjuntos de crenças possíveis de se ter a partir da teoria com *defaults* inicial. Como o que é provado tanto determina quanto é determinado pelo o que não é provado, não é fácil se obter as extensões *default*. Para superar essa aparente circularidade, usa-se uma construção de ponto fixo.

Definição III.1 *Seja $\Delta = (D, W)$ uma teoria com* defaults *fechados. Para cada conjunto de fórmulas fechadas $S \subseteq L$, seja $\Gamma(S)$ o menor conjunto satisfazendo as três propriedades seguintes:*

- *P1.* $W \subseteq \Gamma(S)$;

[1] Usaremos extensões *default*, ao invés de apenas extensões, como é comum na literatura, para as diferenciarmos das extensões de teorias com novos axiomas. Note que, no primeiro caso, usa-se extensões *default para* uma teoria com *defaults*, enquanto que, no segundo, extensão *de* uma teoria (com *defaults* ou não).

- *P2.* $Th(\Gamma(S)) = \Gamma(S)$;
- *P3. Se* $\frac{\alpha:\beta_1,\ldots,\beta_m}{\omega} \in D$ *e* $\alpha \in \Gamma(S)$, *e* $\neg\beta_1,\ldots,\neg\beta_m \notin S$, *então* $\omega \in \Gamma(S)$.

Um conjunto de *wffs* $E \subseteq L$ *h uma* extensão *default para* Δ se e *somente* se $\Gamma(E) = E$, *i.e.*, se e somente se E *é* u m ponto *fixo do operador* Γ.

O teorema a seguir apresenta uma caracterização mais intuitiva das extensões *default* para teorias com *defaults* fechados.

Teorema III.1 (Teorema 2.1 de REITER [7]) *Sejam E um conjunto de sentenças da linguagem e* $\Delta = (D, W)$ *uma teoria com* defaults *fechados. Definimos*

$$\begin{aligned} E_0 &= W, \text{ e para todo } i \geq 0: \\ E_{i+1} &= Th(E_i) \cup \left\{\omega / \frac{\alpha:\beta_1,\ldots,\beta_m}{\omega} \in D, \text{ onde } \alpha \in E_i \text{ e } \neg\beta_1,\ldots,\neg\beta_m \notin E\right\}. \end{aligned}$$

Então, E é uma extensão default *para* Δ *se e somente se* $E = \bigcup_{i=0}^{\infty} E_i$

Preferimos considerar $Th(S)$ como o conjunto de consequências de S (definido no capítulo anterior), ao invés de, como na literatura sobre *defaults*, como o fecho transitivo sobre a operação $\vdash$ (é provado a partir dei), pois julgamos que o uso de tal operação envolve um sistema formal especifico com suas regras de inferência. Se considerarmos um sistema formal correto e completo, as duas definições são equivalentes.

Note que o teorema não dá uma forma construtiva de se obter extensões *default*, pois faz uso de um oráculo que informa se uma fórmula pertence às extensões *default* ou não.

Consideramos que uma fórmula F é consequência lógica de uma teoria com *defaults* $\Delta = (D, W)$, o que é representado por $W \models_D F$, se e somente se $E \models F$ para alguma extensão *default* E para Δ.

Existem teorias com *defaults* que não possuem extensões *default*, chamadas de *incoerentes*. Teorias com *defaults* que não possuem uma extensão *default* inconsistente (que contém o conjunto de todas as wffs da linguagem) são chamadas de *consistentes*.

Seja uma teoria com *defaults* fechados $\Delta = (D, W)$ com uma extensão *default* E. O conjunto de *defaults* geradores para E com respeito a Δ é o conjunto:

$$\text{GD}(E, \Delta) = \left\{\frac{\alpha:\beta_1,\ldots,\beta_m}{\omega} \in D / \alpha \in E \text{ e } \neg\beta_1,\ldots,\neg\beta_m \notin E\right\}.$$

Dados um conjunto D do *defaults* e $d = \frac{\alpha:\beta_1,\dots,\beta_m}{\omega}$, definimos:

$$\begin{aligned} \text{CONSEQUENTES}(D) &= \left\{\omega / \frac{\alpha:\beta_1,\dots,\beta_m}{w} \in D\right\}, \\ \text{CONSEQUENTE}(d) &= \{\omega\}. \end{aligned}$$

Teorema III.2 (Teorema 2.5 de REITER [7]) *Suponha que E é uma extensão* default *para uma teoria com* defaults *fechados* $\Delta = (D, W)$. *Então,*

$$E = Th(W \cup CONSEQUENTES(GD(E, \Delta))).$$

Outro resultado importante é dado abaixo:

Teorema III.3 (Minimalidade de extensões *default*) *Se* E *e* F *são extensões* default para *uma teoria com* defaults fechados (D, W) *e se* $E \subseteq F$, *então* $E = F$.

III.2 Teorias com *defaults* normais fechados

Como já foi mencionado, acredita-se que todos os casos práticos existentes são formalizados com teorias com *defaults* normais e semi-normais. Nessa e na próxima seção, estudaremos essas classes de teorias, restritas a *defaults* fechados.

As teorias com *defaults* normais fechados possuem uma série de propriedades peculiares:

1. Toda teoria com *defaults* normais fechados tem ao menos uma extensão *default*;
2. (Semi-monotonicidade): Suponha que D e D' são conjuntos de *defaults* normais fechados tais que $D' \subseteq D$. Seja E' uma, extensão *default* para a teoria com *defaults* normais fechados $\Delta' = (D', W)$. Seja $A = (D, W)$, então Δ tem uma extensão *default* E tal *que*
   (a) $E' \subseteq E$ e
   (b) $\text{GD}(E', \Delta') \subseteq \text{GD}(E, \Delta)$;
3. (Ortogonalidade de Extensões *Default*): Se uma teoria com *defaults* normais fechados (D, W) tem duas extensões *default* distintas E e F, então $E \cup F$ é inconsistente;
4. O número de extensões *default* de uma teoria com *defaults* normais fechados não decresce com a adição de novos *defaults* normais fechados.

Para a prova dessas assertivas, conferir REITER [7].

III.3 Teorias com *defaults* semi-normais fechados

Apesar de possuir propriedades bastante desejáveis (que garantem a existência de uma teoria de provas para teorias com *defaults* normais, como aquela de REITER [7]), as teorias com *defaults* normais não são suficientes para formalizar todas as situações possíveis. Quando os *defaults* interagem une com os outros, faz-se necessário o uso de teorias com *defaults* semi-normais. Considere o exemplo abaixo, de REITER e CRISCOULO [10]:

Exemplo III.1 *Devemos expressar os seguintes fatos:*

- *Tipicamente, adultos estão empregados;*
- *Tipicamente, quem abandona a escola é adulto;*
- *Tipicamente, quem abandona a escola não está empregado.*

Os defaults *normais abaixo formalizam esses fatos:*

$$\left\{\frac{Adulto(x){:}Empregado(x)}{Empregado(x)}, \frac{Abandonou(x){:}Adulto(x)}{Adulto(x)}, \frac{Abandonou(x){:}\neg Empregado(x)}{\neg Empregado(x)}\right\}$$

Dada uma pessoa que abandonou a escola, essa teoria com defaults *possui duas extensões* default, *uma que afirma que essa pessoa está desempregada, e uma segunda que afirma que está empregada. A segunda extensão* default *não é intuitiva. Ela se deve ao fato de que pessoas que abandonam a escola não são adultos típicos.*

O default *semi-normal abaixo, que deve substituir o primeiro* default *acima, resolve o problema colocando as pessoas que abandonam a escola explicitamente com exceção a regra:*

$$\left\{\frac{Adulto(x) : Empregado(x) \wedge \neg Abandonou(X)}{Empregado(x)}\right\}$$

Entretanto, as propriedades da seção anterior não valem para teorias com *defaults* semi-normais.

Os exemplos a seguir mostram que a semi-monotonicidade e a ortogonalidade de extensões *default* não valem nesse contexto.

Exemplo III.2 (LUKASZEWICZ [11]) *Seja* $\Delta = (D, W)$, *onde:*

$$W = \{R(a)\} \text{ e } D = \left\{\frac{R(a) : P(a) \wedge Q(a)}{Q(a)}\right\}.$$

Δ *tem uma extensão* default $E = Th(\{R(a), Q(a)\})$.

Ao adicionarmos o default $\frac{:\neg P(a)}{\neg P(a)}$ *à* D, *obtemos uma nova teoria com* defaults *com a única extensão* default $F = Th(R(a), \neg P(a))$. *Obviamente,* $E \not\subseteq F$. *Logo, a semi-monotonicidade não vale nesse contexto.*

Exemplo III.3 *Seja* $\Delta = (D, W)$, *onde* $W = \{\}$ *e* $D = \left\{\frac{:\neg P(a) \wedge Q(a)}{Q(a)}, \frac{:\neg Q(a) \wedge P(a)}{P(a)}\right\}$. Δ *possui duas extensões* default $E = Th(\{Q(a)\})$ e $F = Th(\{P(a)\})$. *Porém,* $E \cup F$ *é consistente. Logo, as extensões* defaults *não são ortogonais.*

A teoria com *defaults* $\Delta = (D, W)$, onde:

$$\begin{aligned} W &= \{\} \\ D &= \left\{\frac{: A \wedge \neg B}{A}, \frac{: B \wedge \neg C}{B}, \frac{: C \wedge \neg A}{C}\right\} \end{aligned} \tag{III.1}$$

não possui extensão *default*. Isso acontece, pois A depende da ausência de B, B da ausência de C, e C da ausência de A. Assim, ao inferirmos A bloqueamos C, mas permitimos a inferência de B, que invalida a inferência de A. Se D fosse composto só dos dois primeiros *defaults*, Δ teria $Th(\{B\})$ como extensão *default*, negando a propriedade 4 da seção anterior nesse contexto.

Entretanto, ETHERINGTON [9] caracterizou uma classe de teorias com *defaults* gemi-normais que são coerentes. Apesar de considerar apenas teorias com *defaults* fechados, oa resultados a seguir podem ser generalizados para teorias com *defaults* abertos.

Se observarmos, agora, o comportamento da teoria com *defaults* semi-normais (III.1), perceberemos uma característica comum às teorias com *defaults* semi-normais incoerentes: por ser necessário que todos os *defaults* possíveis sejam aplicados, alguns destes têm consequentes que levam à negação das justificativas de outros *defaults* já aplicados. Esse problema só ocorre se alguma fórmula depende da ausência de outras e, ao mesmo tempo, garante a inferência daquela. A caracterização abaixo de uma classe de teorias com *defaults* semi-normais cujos elementos são teorias coerentes se baseia nessa idéia: determinar se as dependências involvem circularidades.

Definição III.2 (**$\lll$ e $\leqq$**) *Seja* $\Delta = (D, W)$ *uma teoria com* defaults *semi-normais fechados. Sem perda de generalidade, consideremos que todas as fórmula estejam na forma clausal.*

Definimos as relações parciais $\lll$ *e* $\leqq$ *sobre* Literais $\times$ Literais *como:*

1. *Se* $\alpha \in W$, *então* $\alpha = (\alpha_1 \vee \ldots \vee \alpha_n)$, *para algum* $n \geq 1$;
Para todo $\alpha_i, \alpha_j \in \{\alpha_1, \ldots, \alpha_n\}$, *se* $\alpha_i \neq \alpha_j$, *então* $\neg\alpha_i \lessapprox \alpha_j$.

1. *Seja* $d \in D$, $d = \frac{\alpha:\beta\wedge\gamma}{\beta}$. *Sejam* $\alpha_1, \ldots, \alpha_r, \beta_1, \ldots, \beta_S$, e $\gamma_1, \ldots, \gamma_t$, *os literais das formas clausais de* a, β e γ, *respectivamente. Então:*

   (a) *Se* $\alpha_i \in \{\alpha_1, \ldots, \alpha_r\}$ *e* $\beta_j \in \{\beta_1, \ldots, \beta_S\}$, *então* $\alpha_i \lessapprox \beta_j$;

   (b) *Se* $\gamma_i \in \{\gamma_1, \ldots, \gamma_t\}$ *e* $\beta_j \in \{\beta_1, \ldots, \beta_S\}$, *então* $\neg\gamma_i \lll \beta_j$;

   (c) *Seja* $\beta = \beta_1 \wedge \ldots \wedge \beta_m$, *para algum* $m \geq 1$.
   Para cada $i \leq m$, $\beta_i = \beta_{i,1} \vee \ldots \vee \beta_{i,m_i}$, *onde* $m_i \geq 1$.
   Dessa forma, se $\beta_{i,j}, \beta_{i,k} \in \{\beta_{1,1}, \ldots, \beta_{m,m_m}\}$ *e* $\beta_{i,j} \neq \beta_{i,k}$, *então* $\neg\beta(i,j) \lessapprox \beta(i,k)$;

3. *As seguintes relações de transitividade valem:*

   (a) *Se* $\alpha \lessapprox \beta$ *e* $\beta \lessapprox \gamma$, *então* $\alpha \lessapprox \gamma$;

   (b) *Se* $\alpha \lll \beta$ *e* $\beta \lll \gamma$, *então* $\alpha \lll \gamma$;

   (c) *Se* $\alpha \lll \beta$ *e* $\beta \lessapprox \gamma$ *ou* $\alpha \lessapprox \beta$ *e* $\beta \lll \gamma$, *então* $\alpha \lll \gamma$.

Essa definição complexa considera que $a \lll \beta$ ou que $a \lessapprox \beta$ se a pode participar de alguma maneira da inferência de β. Estamos mesmo interessados em $\lll$, que trata da relação da parte da justificativa que não faz parte do consequente com o próprio consequente, e a relação $\lessapprox$ é auxiliar. A intuição por traz de algumas das regras está no fato de

$$(\alpha_1 \vee \ldots \vee \alpha_n) \equiv [(\neg\alpha_1 \wedge \ldots \wedge \neg\alpha_{j-1} \wedge \neg\alpha_{j+1} \wedge \ldots \wedge \neg\alpha_n) \rightarrow \alpha_j], \forall 1 < j < n.$$

Definição III.3 (Ordenação) *Uma teoria com* defaults *semi-normais é dita* ordenada *se e somente se não existe um literal* α, *tal que* $\alpha \lll \alpha$.

Teorema III.4 (Coerência) *Uma teoria com* defaults *semi-normais ordenada tem ao menos uma extensão* default.

Exemplo III.4 *Voltando a (III.1), notamos que aquela teoria não é ordenada, pois* $B \lll A$, $C \lll B$ *e* $A \lll C$; *logo*, $A \lll A$.

Como teorias com *defaults* normais não produzem a relação '$\lll$', elas são sempre ordenadas e, portanto, coerentes, como era de se esperar devido aos resultadas já obtidos.

A ordenação é uma condição apenas suficiente, mas não necessária para a existência de extensões *default*, como ilustra o próximo exemplo, de ETHERINGTON [9]:

Exemplo III.5 *Seja* $\Delta = (D, W)$ *uma teoria com* defaults *semi-normais,* onde

$$W = \{\}$$
$$D = \left\{ \frac{: A \wedge \neg B}{A}, \frac{B \wedge \neg D}{B}, \frac{: (C \rightarrow D) \wedge \neg A}{(C \rightarrow D)} \right\}.$$

Dela surgem $\{B \lll A\}$, $\{D \lll B\}$, *e* $\{C \lesssim D, \neg D \lesssim \neg C, A \lll \neg C, A \lll D\}$. *Daí, obtemos* $A \lll D \lll B \lll A$.

Essa teoria não é ordenada, mas possui uma extensão default*:*

$$Th(\{B, C \rightarrow D\}).$$

Contudo, se acrescentarmos C *à* W, *a teoria resultante não tem extensões* default.

III.4 Construção de extensões *default*

REITER [7] fornece uma teoria de provas para a lógica de *defaults*, i.e., dada uma fórmula β da linguagem, dedermina-se se β pertence à alguma extensão *default* da teoria. Para essa teoria de provas existir, precisa-se usar a propriedade da semi-monotonicidade, valendo, portanto, somente para teorias com *defaults* normais. Contudo, essa teoria de provas não é nem semi-decidível, como mostrado por REITER [7].

Se desejarmos construir extensões *default* para teorias com *defaults*, mesmo *que* normais, o resultado do parágrafo anterior (sobre indecidibilidade) frustra qualquer expectativa. Teremos que nos contentar, então, com subcasos tratáveis.

ETHERINGTON [9] apresenta um procedimento para gerar todas as extensões *default* de uma teoria com *defaults* finita arbitrária. Uma teoria com *defaults* finita é uma com um número finito de *defaults* cuja linguagem possui um número finito de variáveis, de símbolos de predicado, de constantes, e não possui símbolos de função. Essas restrições garantem que existirão um número finito de instâncias fechadas dos *defaults* abertos.

A idéia do procedimento, apresentado na figura III.1, é de construir as extensões através de aproximações sucessivas. Em cada uma dessas aproximações, utilizam-se os *defaults* possíveis de serem aplicados (e ainda não aplicados), um de cada vez. Quando não for possível se aplicar mais nenhum *default*, passamos à próxima aproximação. Quando duas aproximações forem idênticas, dizemos que o procedimento convergiu.

A operação choose from introduz um certo grau de não-determinismo, o que é necessário quando existem mais de uma extensão *default*.

Teorema III.5 *Existe uma computação que converge de forma que* $H_n = H_{n-1}$ *e* $Th(H_n) = E$ *se e somente se* E *é uma extensão* default *para a teoria com* defaults (D, W).

$H_0 \leftarrow W;\ j \leftarrow 0;$
<u>repeat</u>
  $j \leftarrow j+1;\ h_0 \leftarrow W;\ GD_0 \leftarrow \{\};\ i \leftarrow 0;$
  <u>repeat</u>
    $D_i \leftarrow \left\{\frac{\alpha:\beta}{\omega} \in D/(h_i \vdash \alpha), (h_i \not\vdash \neg\beta), (H_{j-1} \not\vdash \neg\beta)\right\};$
    <u>if</u> $\neg$null$(D_i - GD_i)$ <u>then</u>
      <u>choose</u> δ <u>from</u> $(D_i - GD_i)$;
      $GD_{i+1} \leftarrow GD_i \cup \{\delta\};$
      $h_{i+1} \leftarrow h_i \cup \{$CONSEQUENTE$(\delta)\}$; <u>endif</u>;
    $i \leftarrow i+1;$
  <u>until</u> null$(D_{i-1} - GD_{i-1})$;
  $H_j \leftarrow h_{i-1}$
<u>until</u> $H_j = H_{j-1}$.

Figura, III.1: Procedimento que gera as extensões *default* para uma teoria com *defaults* finita arbitrária

Isto é, o procedimento pode retornar todas as extensões *default*, e somente extensões *default* podem ser retornadas.

Entretanto, apesar do resultado acima, existem dois poréns: continua-se fazendo testes de não-provabilidade, que só são computáveis em subcasos restritos (um desses subcasos são as teorias com *defaults* finitas, desde que W seja também finito); e existem algumas teorias com *defaults* finitas que admitem computações não convergentes.

Devido à segunda limitação exposta no parágrafo acima, ETHERINGTON [9] estuda subcasos que não admitem computações não-convergentes. Mencionaremos somente o mais simples deles: teorias com *defaults* normais finitas. Para maiores detalhes, veja [9].

Teorema III.6 (Convergência *Forte*) *Para teorias com* defaults *normais finitas, o procedimento acima sempre converge em uma extensão* default *imediatamente, i.e., Th(H1) é sempre uma extensão* default.

Em face ao resultado acima, podemos simplificar o procedimento da figura III.1 para encontrar extensões *default* de teorias com *defaults* normais finitas, como mostrado na figura III.2.

Exemplo III.6 *Para a teoria com* defaults $\Delta = (D, W)$, *onde:*

$$W = \{A \rightarrow \neg B\}$$
$$D = \left\{\frac{:A}{A}, \frac{:B}{B}\right\}$$

$h_0 \leftarrow W$; $GD_0 \leftarrow \{\}$; $i \leftarrow 0$;
<u>repeat</u>
  $D_i \leftarrow \left\{\frac{\alpha:\beta}{w} \in D/(h_i \vdash \alpha), (h_i \not\vdash \neg\beta)\right\}$;
  <u>if</u> $\neg$null($D_i - GD_i$) <u>then</u>
    <u>choose</u> δ <u>from</u> ($D_i - GD_i$);
    $GD_{i+1} \leftarrow GD_i \cup \{\delta\}$;
    $h_{i+1} \leftarrow h_i \cup$ {CONSEQUENTE(δ)}; <u>endif</u>;
  $i \leftarrow i + 1$;
<u>until</u> null($D_{i-1} - GD_{i-1}$);
$H_1 \leftarrow h_{i-1}$.

Figurw III.2: 'Procedimento que gera as extensões *default* para uma teoria com *defaults* normais finita

que possui as extensões default $Th(\{A \rightarrow \neg B, A, \neg B\})$ *e* $Th(\{A \rightarrow \neg B, B, \neg A\})$, *o procedimento possui duas diferentes computações:*

Computação 1

$$\begin{array}{lllllllll} h_0 & \leftarrow & \{A \rightarrow \neg B\}, & GD_0 & \leftarrow & \{\}, & D_0 & \leftarrow & \left\{\frac{:A}{A}, \frac{:B}{B}\right\} \\ h_1 & \leftarrow & \{A \rightarrow \neg B, A\}, & GD_1 & \leftarrow & \left\{\frac{:A}{A}\right\}, & D_1 & \leftarrow & \left\{\frac{:A}{A}\right\} \\ H_1 & \leftarrow & \{A \rightarrow \neg B, A\} & & & & & & \end{array}$$

Computação 2

$$\begin{array}{lllllllll} h_0 & \leftarrow & \{A \rightarrow \neg B\}, & GD_0 & \leftarrow & \{\}, & D_0 & \leftarrow & \left\{\frac{:A}{A}, \frac{:B}{B}\right\} \\ h_1 & \leftarrow & \{A \rightarrow \neg B, B\}, & GD_1 & \leftarrow & \left\{\frac{:B}{B}\right\}, & D_1 & \leftarrow & \left\{\frac{:B}{B}\right\} \\ H_1 & \leftarrow & \{A \rightarrow \neg B, B\} & & & & & & \end{array}$$

III.5 Semântica

Quando a lógica de *defaults* foi proposta por REITER [7], faltou uma caracterização semântica da mesma. Ao contrário, apresentou-se uma caracterização de ponto fixo das extensões *default* para uma teoria com *defaults*. Inicialmente, a falta da semântica era suprida com essa caracterização intuitiva e com a semântica Tarskiana das extensões individuais.

O que faltava era uma caracterização das extensões *default* de (D, W) através de uma teoria de modelos da seguinte maneira: os *defaults* restringem os modelos de W de uma maneira que qualquer uma dessas classes de modelos restritas resultantes satisfaça todas e somente as fórmulas de uma extensão *default* para (i),W), e, conversamente, qualquer extensão *default* para (D, W) seja um conjunto de wffs satisfeito por alguma dessas classes restritas de modelos de W.

Felizmente, alguns trabalhos posteriores supriram essa lacuna. No momento, temos conhecimento do trabalho da LUKASZEWICZ [8], que apresenta uma caracterização semântica das extensões *default* para teorias com *defaults* normais, e dos trabalhos de ETHERINGTON [12,9] e GUERREIRO e CASANOVA [13] que caracterizam extensões *default* para teorias com *defaults* arbitrárias. Apresentaremos aqui apenas a versão de ETHERINGTON [9], mencionando ligeiramente o trabalho de LUKASZEWICZ [8].

LUKASZEWICZ [8] usou o fato da semi-monotonicidade ser uma propriedade de teorias com *defaults* normais para caracterizar as extensões *default* para essas teorias da seguinte forma: a medida que os *defaults* vão sendo aplicados à teoria de primeira ordem inicial, a classe de modelos inicial vai se restringindo devido à adição dos consequentes dos *defaults* (e esses consequentes são consistentes com o conjunto de fórmulas anterior). Devido à semi-monotonicidade, quando mais nenhum *default* puder ser aplicado, estaremos com uma extensão *default*, e a classe restrita de modelos será a classe de modelos para esta extensão *default*.

Lukaszewicz tentou justificar o fato de sua semântica lidar apenas com teorias com *defaults* normais apresentando uma maneira de se transformar *defaults* arbitrários em *defaults* semi-normais, que é sempre válida, e uma maneira de se tranformar *defaults* semi-normais em *defaults* normais, que nem sempre é válida. Além disso, somente *defaults* com justificativas únicas são considerados. Por isso, ainda era necessário uma semântica que lidasse com teorias com *defaults* arbitrárias.

ETHERINGTON [9] apresenta essa caracterização semântica para teorias com *defaults* arbitrárias com justificativas múltiplas. Aqui, deve-se atentar para o fato que, diferentemente do que acontecia com os *defaults* normais (devido à semi-monotonicidade), o consequente de um *default* que pode ser aplicado pode ser inconsistente com a justificativa de algum *default* já aplicado.

As definições de $\geq_\delta$ e $\geq_D$ abaixo capturam a idéia da preferência por classes de modelos mais especializadas, onde os consequentes dos *defaults* valem, a classes de modelos onde os pré-requisitos valem e as justificativas são consistentes, mas que não necessariamente satisfazem os consequentes.

Definição III.4 ($\Gamma_1 \geq_\delta \Gamma_2$ (δ prefere Γ_1 a Γ_2))

Considere um default $\delta = \frac{\alpha:\beta_1,\ldots,\beta_m}{\omega}$, *uma classe de modelos*, Γ, *e* Γ_1, $\Gamma 2 \in 2^\Gamma$. *A ordem parcial correspondente a* δ, $\geq_\delta$, *sobre* 2^Γ *é definida como:*

$$\begin{aligned} \Gamma_1 \ \geq_\delta \Gamma_2 & \textit{ se e somente se} \\ & \forall A \in \Gamma 2, \textit{ tal que } A \models \alpha, \ \exists A_1, \ldots, A_n \in \Gamma 2, \textit{ tal que } A_i \models \beta_i, \textit{ e} \\ & \Gamma_1 = \Gamma_2 - \{A / A \models \neg\omega\}. \end{aligned}$$

Definição III.5 ($\Gamma_1 \geq_D \Gamma_2$ (D prefere Γ_1 a Γ_2))

Considere um conjunto de defaults D, *uma classe de modelos*, Γ, *e* $\Gamma_1, \Gamma 2 \in 2^\Gamma$. *A ordem parcial correspondente a* D, $\geq_D$, *sobre* 2^Γ *é definida como a união da ordens parciais dadas pelos* defaults *de* D, *i.e.,*

$\Gamma_1 \geq_D \Gamma_2$ *se e somente se* $\exists\delta \in D$, *tal que* $\Gamma_1 \geq_\delta \Gamma 2$ *ou* $\exists\Gamma'$, *tal que* $\Gamma 1 \geq_D \Gamma' \geq_D \Gamma_2$.

Até esse ponto, os elementos maximais de $2^{MOD(W)}$, sendo $MOD(W)$ a classe de modelos de W, em relação a $\geq_D$ são as classes de modelos das extensões *default* para a teoria com *defaults* normais $\Delta = (D, W)$. Para teorias com *defaults* arbitrários é necessária a propriedade de *estabilidade* , que definimos abaixo.

Definição III.6 (Estabilidade) *Sejam* $\Delta = (D, W)$ *uma teoria com* defaults *e* Γ *um elemento de* $2^{MOD(W)}$. Γ *é estável para* Δ *se e somente se existe um* $D' \subseteq D$ *tal que* $\Gamma \geq_{D'} MOD(W)$, *e para cada* $\frac{\alpha:\beta_1,\ldots,\beta_m}{\omega} \in D'$, $\exists A_1, \ldots, A_2 \in \Gamma / A_i \models \beta_i$.

Isto é, uma classe de modelos é estável para uma teoria com *defaults* $\Delta = (D, W)$ se é uma especialização da classe de modelos de W, e não refuta as justificativas de qualquer dos *defaults* usados na especialização.

Os teoremas abaixo garantem que as extensões *default* para uma teoria com *defaults* arbitrária correspondem às subclasses maximais e estáveis da classe de modelos para W. Se essas subclasses não existem, a teoria não tem extensões *default*.

Teorema III.7 (Corretude) *Se* E é *uma extensão* default para Δ, *então* $\{M/M \models E)$ é *estável e maximal para* A.

Teorema III.8 (Completude) *Se* Γ *é uma classe de modelos para* Δ *maximal e estável, então* Γ *é a classe de modelos para alguma extensão* default *para* Δ. *Isto é* $\{w / \forall A \in \Gamma,$ *tal que* $A \models w\}$ *é uma extensão* default *para* Δ.

Para as provas desses teoremas, consulte ETHERINGTON[9].

Os exemplos abaixo [9] ilustram os resultados.

Exemplo III.7 *Considere a teoria com* defaults $\Delta = (D, W)$, *onde:*

$$D = \left\{\delta_1 = \frac{A : B \wedge \neg C}{B}, \delta_2 = \frac{A : C \wedge \neg B}{C}\right\} \text{ e } W = \{A\}$$

Essa teoria possui duas extensões default, $Th(\{A, B\})$ *e* $Th(\{A, C\})$, *que correspondem a duas classes maximais de modelos obtidos a partir da classe de modelos de* W:

$$\{M/M \models A\} \leq_{\delta_1} \{M/M \models A, B\} \text{ e}$$
$$\{M/M \models A\} \leq_{\delta_2} \{M/M \models A, C\}$$

Além disso, é fácil verificar que essas classes de modelos maximais são estáveis.

Exemplo III.8 *A teoria com* defaults $\Delta = \left(\left\{\frac{:\neg A}{A}, \{\}\right\}\right)$ *é incoerente. A única classe de modelos maximal, $\{M/M \models A\}$, não é estável.*

Essa semântica é bem diferente da semântica tradicional, onde as noções de validade, satisfatibilidade, etc., dependem apenas da classe de modelos inicial. Entretanto, esse tipo de semântica não se encaixa com a maneira como se completa o conhecimento na lógica de *defaults*. Para sabermos quais os modelos que satisfazem uma teoria com *defaults*, é necessário informações sobre qual o *default* aplicado, acalmado com a possibilidade de uma semântica estática. Além disso, se considerarmos os *defaults* como operadores que transformam uma teoria com *defaults* em várias teorias de primeira ordem, a semântica subjacente tem que captar essa idéia de transformação da teoria de primeira ordem incompleta inicial nas várias teorias de primeira ordem finais.

Devido à essa complexidade intrínseca à semântica para a lógica de *defaults*, parece-nos mais fácil se continuar trabalhando com as extensões *default* e seus modelos.

O exemplo a seguir [11] fecha a seção e o capítulo.

Exemplo III.9 *Considere* a *teoria* com defaults $A = (D, W)$, onde:

$$\begin{aligned} D &= \left\{ d_1 = \frac{p : r \wedge \neg s}{r}, d_2 = \frac{q : s}{s} \right\} \\ W &= \{p, q\} \end{aligned}$$

Essa teoria possui uma extensão default, $Th(\{p, q, s\})$. *Porém, existem duas classes maximais de modelos obtidas a partir da classe de modelos de W:*

$$\begin{aligned} &\{M/M \models p, q\} \leq_{d_2} \{M/M \models p, q, s\} \text{ e} \\ &\{M/M \models p, q\} \leq_{d_1} \{M/M \models p, q, r\} \leq_{d_2} \{M/M \models p, q, r, s\}, \end{aligned}$$

mas só o primeiro é estável, pois, no segundo caso, não existe $m \in M$, tal que $m \models r \wedge \neg s$.

Capitulo IV

Especificação de Programas com a Lógica de Defaults

Nesse capitulo, estudamos como a lógica de *defaults* pode ser usada para especificar programas. Inicialmente, mostramos exemplos da utilização desse formalismo nessa tarefa. Depois, introduzimos os conceitos de extensões conservativas e de interpretações nesse contexto. Finalmente, discorremos sobre como esses conceitos se combinam para representar o processo de desenvolvimento de programas por refinamentos sucessivos.

IV.1 Aplicações

Nessa seção apresentamos três exemplos que mostram como a lógica de *defaults* pode ser usada para especificar programas e tipos abstratos de dados.

IV.1.1 Especificação de programas gerais

A lógica de *defaults* pode ser utilizada para resumir, em uma única especificação, as especificações de vários programas.

Por exemplo, sejam $car1$, $car2$, $car3$, $car4$ fórmulas de primeira-ordem que especificam características distintas. Estamos interessados em especificar alguns programas que se comportem de maneiras diferentes em relação às características acima. Desta forma, um certo PROG1 possui as características 1 e 2, mas não as 3 e 4. Outro, PROG2, possui as características 1, 2 e 3, mas não a C Um terceiro, PROG3, possui as quatro características.

Ao invés de termos três especificações separadas, podemos especificar um programa geral com a teoria com *defaults* $\Delta = (D, W)$, onde

$$
\begin{aligned}
W &= \{(car1),(car2)\};\\
D &= \left\{\frac{:\neg(car3)}{\neg(car3)},\frac{:(car3)}{(car3)},\frac{:\neg(car4)}{\neg(car4)},\frac{:(car4)\wedge(car3)}{(car4)\wedge(car3)},\right\}
\end{aligned}
$$

Essa teoria com *defaults* possui três extensões *default:*

$$
\begin{aligned}
E1 &= Th((car1),(car2),\neg(car3),\neg(car4));\\
E2 &= Th((car1),(car2),(car3),\neg(car4));\\
E3 &= Th((car1),(car2),(car3),(car4)).
\end{aligned}
$$

PROG1, PROG2 e PROG3 são especificados por E1, E2 e E3, respectivamente.

Nesse contexto, as extensões *default* para a teoria com *defaults* usada para especificar o programa geral seriam as várias especificações de. programas resumidas na especificação do programa, geral. A escolha de uma dada extensão *default* seria uma opção de projeto.

IV.1.2 Tratamento de erros em TAD's

Agora, vamos tratar do problema de manipulação de erros em tipos abstratos de dados (TAD's).

Quando especificamos um TAD, estamos preocupados em descrever seu comportamento natural. Não podemos garantir, entretanto, que o programa abstrato não usará esse TAD de forma, indevida. Por isso, é interessante especificar o que se fazer numa situação de erro,

Uma razão para não se abordar esse problema é a seguinte: programas corretos nunca usam o TAD indevidamente.

Entretanto, durante o processo de desenvolvimento e depuração de programas, é inevitável que, eventualmente, erros ocorram. Desta forma, pelo menos durante essa fase, é interessante alguma forma de se manipular incorreções. Mais ainda, TAD's públicos, que são usados em várias situações por programadores diferentes, devem alertar para seu uso indevida.

Veloso [6] optou por estender a teoria com um elemento que indique erro. Porém, esse elemento, agora pertencente à linguagem da especificação, pude ser manipulado pelas operações do TAD. Isso causa o problema da propagação de erros que, se apropriadamente tratado, gera extensões não expansivas.

Podemos usar a lógica de *defaults* nesse problema. Partimos do pressuposto que, ao usarmos o TAD incorretamente, ele não se comportará mais como o TAD em questão. Isto é, ao aplicarmos uma operação inválida sobre o TAD, não podemos mais garantir que este se comporte como a especificação original. Cabe ao sistema resolver o impasse.

Um TAD que possui suas características especificadas pelas fórmulas $car1$, $car2$ e $car3$ seria especificado pela teoria com *defaults* $\Delta = (D, W)$, onde

$$D = \left\{ \frac{: (car1) \wedge (car2) \wedge (car3) \wedge \neg[(cond1) \vee (cond2) \vee (cond3)]}{(car1) \wedge (car2) \wedge (car3)} \right\};$$
$$W = \{\}$$

$Cond1$, $cond2$ e $cond3$ expressam as situações onde o TAD está sendo mal utilizado.

Essa teoria com *defaults* possui duas possíveis extensões *default*: um que especifica o TAD em questão e outra que não especifica nada (quando alguma condição de erro aparece), indicando que nada mais se pode fazer sobre aquele TAD.

Poderíamos, também, acrescentar à W a fórmula:

$$cond1 \vee cond2 \vee cond3 \rightarrow \neg(car1) \wedge \neg(car2) \wedge \neg(car3)$$

indicando que, diante de uma operação inválida, nenhuma das características anteriores do TAD valem.

IV.1.3 Passagem de parâmetros

Veloso [6] mostrou que a especialização por passagem de parâmetros é, na verdade, uma interpretação de especificações. Quando o TAD funciona da mesma maneira para todas as possíveis instanciações, a lógica de primeira ordem se adequa para especificá-lo. Mas suponha um TAD que representa uma lista de elementos. Suponha também que, por alguma razão, deseja-se acessar esta lista de formas diferentes, de acordo com o tipo dos dados instanciados. Por exemplo, se furem sequências finitas de caracteres iniciadas por letra maiúscula, deseja-se acessá-la como uma pilha; senão, como uma fila.

A teoria com *defaults* $\Delta = (D, W)$, onde

$$W = \left\{ \begin{array}{lcl} [\text{axiomas de} & \text{lista} & \text{de elementos}] \\ \text{acesso(fifo)} & \rightarrow & [\text{axiomas de acesso fifo}] \\ \text{acesso(lifo)} & \rightarrow & [\text{axiomas de acesso lifo}] \\ \text{acesso(lifo)} & \rightarrow & \neg\text{acesso(fifo)} \end{array} \right\}$$

$$D = \left\{ \frac{:carseq \wedge acesso(lifo)}{acesso(lifo)}, \frac{:\neg\, carseq \wedge acesso(fifo)}{acesso(fifo)} \right\}$$

especifica esse comportamento onde *carseq* é uma wff que especifica urna sequência finita de caracteres iniciada por letra maiúscula. [axiomas de lista de elementos] é um conjunto de axiomas que especifica as características de uma lista de elementos sem definir a forma de acesso dessa lista, [axiomas de acesso fifo] e [axiomas de acesso lifo] são conjuntos de axiomas que especificam que a lista será acessada de forma fifo ('first-in-first-out'), no primeiro caso, ou de forma lifo ('last-in-first-out'), no segundo. Δ possui duas extensões *default*: uma que especifica uma pilha e outra que especifica uma fila, Dependendo da instanciação, uma, delas será escolhida.

IV.2 Desenvolvimento de programas por refinamentos sucessivos

O processo de desenvolvimento de programas por refinamentos sucessivos divide a construção de um programa a partir de sua especificação em várias etapas pequenas. Para entendermos a natureza de cada uma dessas etapas, quando a especificação é feita com a lógica de *defaults*, precisamos definir o que são extensões conservativas de teorias com *defaults* e interpretações entre teorias com *defaults*. Esses conceitos devem ser de tal moda que o teorema da modularização continue valendo nesse novo contexto, garantindo a corretude do desenvolvimento de programas por refinamentos sucessivos.

IV.2.1 Extensões conservativas e expansivas

Quando a lógica de *defaults* foi proposta por Reiter [7], assumiu-se que a linguagem sobre a qual se define uma teoria com *defaults* fosse uma linguagem L de primeira ordem formada pelas wffs usuais sobre um alfabeto fixo consistindo de variáveis, de símbolos de função, de símbolos de predicado e de símbolos lógicos. Para nossos propósitos, não consideraremos o alfabeto como fixo, pois estamos interessados em estender R linguagem. Faremos, então, explícita referência sobre qual linguagem a teoria com *defaults* está definida

Na discussão abaixo, consideraremos apenas teorias com *defaults* fechada~.Portanto, quando nos referirmos apenas a teorias com *defaults*, entenda-se que os *defaults* em discussão são fechados.

Precisamos, inicialmente, definir o que vem a ser extensões clássicas e extensões conservativas de teorias no contexto da lógica de *defaults*. Estamos interessados em estender uma teoria com *defaults* sem perder as suas consequências antes da extensão, i.e., monotonicamente. A maneira mais natural (e ingênua) seria tentar adaptar essas definições diretamente daquelas análogas de lógica de primeira ordem.

Sejam $\Delta 1 = (D1, W1)$ e $\Delta 2 = (D2, W2)$ teorias com *defaults* fechados sobre linguagens L1 e L2, respectivamente. Seja F uma wff de L1.

Definição IV.1 *Dizemos que $\Delta 2$ é uma extensão monotônica de $\Delta 1$ se e somente se:*

$$L1 \subseteq L2 \text{ e se } W1 \models_{D1} F, \text{ então } W2 \models_{D2} F$$

Definição IV.2 *Seja $\Delta 2$ uma extensão monotônica de $\Delta 1$. Essa extensão é dita conservativa se e somente se:*

$$W2 \models_{D2} F, \text{ então } W1 \models_{D1} F$$

De outra forma, dizer que estendemos um teoria com *defaults* $\Delta 1$ para uma teoria com *defaults* $\Delta 2$ monotonicamente significa dizer que qualquer fórmula pertencente a uma extensão *default* para $\Delta 1$ estará em uma extensão *default* para $\Delta 2$. Dizer que essa extensão monotônica é conservativa significa dizer que qualquer fórmula da linguagem pequena que pertence a alguma extensão *default* para $\Delta 2$ já pertencia a alguma extensão *default* para $\Delta 1$.

Essas definições, entretanto, se mostraram muito fracas para nossos propósitos. Isso acontece, pois se assume que o conjunto de consequências de uma teoria com *defaults* é a união de suas extensões *default*. Ora, quando usamos uma teoria com *defaults* para especificar várias teorias de primeira ordem, cada extensão *default* assume um significado próprio. Assim, ao estendermos essa teoria com *defaults*, essa extensão deve se dar em cada uma de suas extensões *default*. Dessa forma, não podemos afirmar que uma fórmula isolada é consequência lógica da teoria com *defaults*; tudo vai depender do contexto, isto é, das outras fórmulas que também são consequências. Logo, por esse ponto de vista, conjuntos de fórmulas (e não apenas fórmulas) são consequências lógicas de teorias com *default*; mais especificamente, cada extensão *default* é uma consequência lógica de sua teoria com *defaults*. Partimos, então, para extensões monotônicas e conservativas mais restritas, baseadas no último ponto de vista, como expresso nas definições a seguir.

Em toda a tese, usaremos $\mathcal{E}i$ para representar o conjunto de todas as extensões *default* para a teoria com *defaults* Δi.

Sejam $\Delta 1$ e $\Delta 2$ teorias com *defaults* sobre linguagens L1 e L2, respectivamente..

Definição IV.3 *$\Delta 2$ é uma extensão monotônica restrita de $\Delta 1$, o que é representado por $\Delta 1 \subseteq_D \Delta 2$, se e somente se:*

Existe uma função injetiva $f : \mathcal{E}1 \mapsto \mathcal{E}2$, de forma que, se $(E1, E2)$ é um par dessa função, E2 é uma extensão clássica de E1 ($E1 \subseteq E2$).

Obviamente a extensão da definição IV.3 é uma extensão monotônica.

Definição IV.4 *Seja $\Delta 2$ uma extensão monotônica restrita de $\Delta 1$. $\Delta 1 \subseteq_D \Delta 2$ é conservativa restrita, o que é representado por $\Delta 1 \leq_D \Delta 2$, se e somente se:*

A função f da definição de extensão monotônica restrita é, na verdade, uma função bijetiva, e para cada par $(E1, E2)$ dessa função, $E1 \subseteq E2$ é uma extensão conservativa clássica ($E1 \leq E2$).

Já mencionamos que as definições IV.1 e IV.2 se referem a uma noção de consequência para a lógica de *defaults* e as definições IV.3 e IV.4 a outra. Contudo, apesar das duas primeiras não relacionarem diretamente as extensões *default* para as teorias com *defaults* envolvidas, implicitamente tal relação existe quando as extensões *default* para as teorias com *defaults* envolvidas são finitamente axiomatizáveis, como veremos nos próximos parágrafos.

Consideraremos que as teorias com *defaults* da discussão abaixo possuem extensões *default* finitamente axiomatizáveis.

Sejam $\Delta 1 = (D1, W1)$ e $\Delta 2 = (D2, W2)$ teorias com *defaults* sobre linguagens L1 e L2, respectivamente, tal que $L1 \subseteq L2$. Se A3 6 uma extensão monotônica de $\Delta 1$, então para cada fórmula F de L1, se $W1 \models_{D1}$ F então $W2 \models_{D2}$ F. Dizer que $W1 \models_{D1} F$, significa dizer que F pertence a uma extensão *default* para $\Delta 1$. Consideremos uma extensão *default* $E1$ para $\Delta 1$. Suponha que $E1$ possui $n \geq 1$ axiomas, que chamaremos de $A1, \ldots An$; então A1 $\wedge \ldots \wedge$ An pertence à $E1$. Logo,

$$W1 \models_{D1} A1 \wedge \ldots \wedge An.$$

Por definição, $W2 \models_{D2} A1 \wedge \ldots \wedge$ Ara. Logo A1 $\wedge \ldots \wedge An$ está em unia, extensão *default* $E2$ para $\Delta 2$, o que implica que $A1, \ldots, An$ estão em $E2$. Logo, $E1 \subseteq E2$. Assim, para cada extensão *default* $E1$ para $\Delta 1$, existe uma extensão *default* $E3$ para $\Delta 2$, tal que $E1 \subseteq E2$. Dessa forma, quando AS é uma extensão monotônica de $\Delta 1$, existe uma função que relaciona suas extensões *default*, de forma que para cada par dessa função existe uma extensão clássica. O que diferencia as extensões monotônicas das monotônicas restritas é que ri. função subjacente da primeira não é necessariamente injetiva, enquanto a da segunda é.

Se $\Delta 2$ é uma extensão conservativa de $\Delta 1$, então A3 é uma extensão monotônica de $\Delta 1$ e para toda fórmula F de L1, se $W2 \models_{D2} F$, então $W1 \models_{D1} F$. Como $\Delta 2$ é uma extensão monotônica de $\Delta 1$, existe uma função $f : C1 \mapsto \mathcal{E}2$, tal que se $(E1, E2)$ é um par dessa função, então $E1 \subseteq E2$. Mostraremos que, se essa extensão monotônica for também conservativa, então para cada par $(E1, E2)$ de f, $E1 \leq E2$. Para isso suponha, por contradição, que $E1 \not\leq E2$. Então, existe uma wff $B \in L1$, tal que $E2 \models$ B e $E1 \not\models B$. Observe que, tomo $E1 \subseteq E2$ e B E $E2$, então a wff T formada pela conjunção dos axiomas de $E1$ com B pertence à $E2$. Logo $W2 \models_{D2} T$. Por definição, $W1 \models_{D1} T$. Logo T está em uma extensão *default* $E1'$ para $\Delta 1$. Então, $E1 \subseteq E1'$, o que contradiz ri. minimalidade de extensões *default*. Logo, $E1 \leq E2$.

Vamos mostrar que f é injetiva. Por contradição, suponha que f não é injetiva. Assim, existem duas extensões *default* $E1$ e $E1'$ para $\Delta 1$, tal que

$E1 \cup E1' \subseteq E2$, onde $E2$ é uma extensão *default* para $\Delta 2$. Seja S a wff formada pela conjunção dos axiomas de $E1$ e $E1'$. Logo, $E2 \models S$. Daí, $W2 \models_{D2} S$. Por definição, $W1 \models_{D1} S$. Dessa forma, existe uma extensão *default* E para $\Delta 1$, tal que $S \subseteq E$. Assim, $E1 \subseteq E$ e $E1' \subseteq E$, o que contradiz a minimalidade de extensões *default*. Logo, f é injetiva. O que diferencia as extensões conservativas das conservativas restritas é que a função subjacente da primeira é injetiva, enquanto a da segunda, bijetiva.

Exemplo IV.1 *Seja* $\Delta 1 = (D1, W1)$ *uma teoria com* defaults *sobre uma linguagem* L *de alfabeto* $\{A, B\}$, *onde*

$$D1 = \left\{\frac{: A \wedge \neg B}{A}, \frac{: B \wedge \neg A}{B}\right\} \text{ e } W1 = \{\ \}$$

Existem duas extensões default*:*

$$E1 = Th(\{A\}) \text{ e } E2 = Th(\{B\}).$$

Já a teoria com defaults $\Delta 2 = (D2, W2)$, *sobre a mesma linguagem* L, *onde*

$$D2 = \left\{\frac{: A}{A}, \frac{: B}{B}\right\} \text{ e } W2 = \{\}$$

possui a seguinte extensão default*:*

$$E3 = Th(\{A, B\}).$$

$\Delta 2$ *é uma extensão monotônica de* $\Delta 1$, *mas não monotônica restrita.*

Exemplo IV.2 *Seja a teoria com* defaults $\Delta 1 = (D1, W1)$ *do exemplo anterior.*

Seja a teoria com defaults $\Delta 2 = (D2, W2)$, *sobre uma linguagem* $L2$ *de alfabeto* $\{A, B, F, G\}$, *onde*

$$\begin{aligned} D2 &= \left\{\frac{: A \wedge F \wedge \neg B \wedge \neg G}{A \wedge F}, \frac{: B \wedge G \wedge \neg F \wedge \neg A}{B \wedge G}, \frac{: F \wedge G \wedge \neg A \wedge \neg B}{F \wedge G}\right\} \text{ e} \\ W2 &= \{\}. \end{aligned}$$

$\Delta 2$ *possui as seguintes extensões* default*:*

$$E3 = Th(\{A, F\}), E4 = Th(\{B, G\}) \text{ e } E5 = Th(\{F, G\}).$$

$\Delta 2$ *é uma extensão conservativa de* $\Delta 1$, *mas não conservativa restrita.*

Exemplo IV.3 *Seja* $\Delta 1 = (D1, W1)$ *unta teoria com* defaults *sobre uma linguagem L de alfabeto* $(A,\ B,\ C,\ D\}$, *onde*

$$D1 = \left\{\frac{:\neg C \wedge \neg D}{A}, \frac{:\neg C \wedge \neg D}{B}, \frac{:\neg A \wedge \neg B}{C}, \frac{:\neg A \wedge \neg B}{D}\right\} \; e \; W1 = \{\ \}$$

Existem duas extensões default*:*

$$E1 = Th(\{A, B\}) \; e \; E2 = Th(\{C, D\}).$$

Já a teoria com defaults $\Delta 2 = (D2, W2)$, *sobre a mesma linguagem L, onde*

$$D2 = \left\{\frac{:A}{A}, \frac{:\neg D}{B}, \frac{:\neg B}{D}\right\} \; e \; W2 = \{C\}$$

possui as seguintes extensões default*:*

$$E3 = Th(\{C\} \cup \{A, B\}) \; e \; E4 = Th(\{C\} \cup \{A, D\}).$$

$\Delta 2$ *é uma extensão monotônica restrita de* $\Delta 1$, *já que* $E1 \subsetneq E3$ *e* $E2 \subsetneq E4$. *Note que* $\Delta 1 \subseteq_D \Delta 2$ *não é uma extensão conservativa, pois, por exemplo, a wff* $A \wedge C$ *é consequência lógica da segunda teoria, mas não da primeira, apesar de pertencer à linguagem desta.*

Precisamos, agora, da contrapartida semântica da nossa definição de extensão conservativa para a lógica de *defaults*: a noção de expansividade.

Sejam novamente $\Delta 1 = (D1, W1)$ e $\Delta 2 = (D2, W2)$ teorias com *defaults* fechados sobre linguagens L1 e L2 respectivamente.

Definição IV.5 $\Delta 2$ *é uma extensão expansiva restrita de* $\Delta 1$ *se e somente se:*

Existe uma função bijetiva $f : \mathcal{E}1 \mapsto \mathcal{E}2$, *de forma que se* $(E1, E2)$ *é um par dessa função, então E2 será uma extensão expansiva clássica de E1.*

Note que para cada um desses pares $(E1, E2)$ provenientes da definição acima, E2 é uma extensão clássica de E1. Logo, se $\Delta 2$ é uma extensão expansiva restrita de $\Delta 1$, então $\Delta 2$ é uma extensão monotônica restrita de $\Delta 1$.

A proposição abaixo garante que a expansividade continua sendo, nesse contexto, condição suficiente para a conservatividade.

Proposição IV.1 *Se* $\Delta 1 \subseteq_D \Delta 2$ *é uma extensão expansiva restrita, então* $\Delta 1 \subseteq_D \Delta 2$ *é conservativa restrita.*

Prova:

Como $\Delta 1$ é uma extensão expansiva restrita de $\Delta 2$, existe uma função bijetiva $f : \mathcal{E}1 \mapsto \mathcal{E}2$, de forma que se $(E1, E2)$ é um par dessa função, então $E2$ é uma extensão expansiva clássica de $E1$. Pela proposição II.1, $E2$ é uma extensão conservativa clássica de $E1$.

Lago, está caracterizada uma função bijetiva $g : \mathcal{E}1 \mapsto \mathcal{E}2$. de forma que se $(E1, E2)$ é rim pas dessa função, então $E1 \leq E2$.

Logo, $\Delta 1 \subseteq_D \Delta 2$ é conservativa restrita.

C.Q.D.

Exemplo IV.4 *Voltando ao exemplo IV.3, $\Delta 1 \subseteq_D \Delta 2$ não é expansiva restrita, pois um modelo $\mathcal{A}$ de E1 que assume A e B com valores verdade True e C e D com False não pode ser expandido para nenhum modelo de alguma extensão para $\Delta 2$.*

Exemplo IV.5 *Seja $\Delta 1 = (D1, W1)$ uma teoria com* defaults *sobre uma linguagem L1 com alfabeto $\{A, B, C\}$ onde,*

$$D1 = \left\{\frac{:\neg C}{A}, \frac{:\neg A}{C}\right\} \text{ e } W1 = \{B\}$$

Existem duas extensões default*:*

$$E1 = Th(\{B\} \cup \{A\}) \text{ e } E2 = Th(\{B\} \cup (C)].$$

Já a teoria $\Delta 2 = (D2, W2)$ sobre uma linguagem L2 com alfabeto $\{A, B, C, F, G\}$, onde

$$D2 = \left\{\frac{:\neg C \wedge \neg G}{A}, \frac{:\neg A \wedge \neg F}{C}, \frac{:\neg C \wedge \neg G}{F} \frac{:\neg A \wedge \neg F}{G}\right\} \text{ e } W2 = \{B\}$$

possui duas extensões default*:*

$$E3 = Th(\{B\} \cup \{A, F\}) \text{ e } E4 = Th(\{B\} \cup \{C, G\}).$$

$\Delta 2$ é uma extensão expansiva restrita de $\Delta 1$, pois todo modelo de E1 pode ser expandido para um modelo de E3 e todo modelo de E2 pode ser expandido para um modelo de E4 e, mais ainda, todo modelo de E3 é uma expansão de um modelo de E1 e todo modelo de E4 é uma expansão de um modelo de E2. Note que uma estrutura para E1 e E2 são atribuições de valores verdade para A, B e C, não considerando nem F, nem G, pois estes não fazem parte de L1.

Logo, pela proposição IV.1, $\Delta 1 \subseteq_D$ A2 é conservativa restrita, como *podemos facilmente constatar.*

O exemplo a seguir mostra porque é necessário que a função f da definição IV.4 seja bijetiva e não apenas injetiva como condição para a conservatividade.

Exemplo IV.6 *Considere a teoria com* defaults $\Delta 1$ *do exemplo IV.5 . Seja* $\Delta 2 = (D2, W2)$ *uma teoria com* defaults *sobre uma linguagem L2 com alfabeto* $\{A, B, C, F, G\}$, *onde*

$$
\begin{aligned}
D2 &= \left\{\frac{:\neg C \wedge \neg G \wedge A}{A}, \frac{:\neg A \wedge \neg F \wedge C}{C}, \frac{:\neg C \wedge \neg G \wedge F}{F}, \frac{:\neg A \wedge \neg F \wedge G}{G},\right. \\
&\quad \left.\frac{:\neg A \wedge \neg C \wedge \neg F \wedge \neg G}{\neg A \wedge \neg C \wedge \neg F \wedge \neg G}\right\} \; e \\
W2 &= \{B\}.
\end{aligned}
$$

Existem três extensões default*:*

$$
\begin{aligned}
E3 &= Th(\{B\} \cup \{A, F\}), \\
E4 &= Th(\{B\} \cup \{C, G\}), \\
E5 &= Th(\{B\} \cup \{\neg A \wedge \neg C \wedge \neg F \wedge \neg G\})
\end{aligned}
$$

Temos que $E1 \leq E3$ *e* $E2 \leq E4$, *mas* $\Delta 1 \subseteq_D \Delta 2$ *não é conservativa, pois podemos concluir* $\neg A \wedge \neg C$ *(wff* $\in$ *L1) em* $\Delta 2$, *mas não podemos chegar a essa conclusão em* $\Delta 1$.

Comparando o exemplo acima com a discussão após a definição IV.4, percebemos que a existência de uma função injetiva entre o conjunto de extensões *default* para a teorias com *defaults* envolvidas, onde para cada par $(E1, E2)$ dessa função ocorre $E1 \leq E2$, não garante a conservatividade, mas a conservatividade garante a existência de tal função, no caso em que as extensões *default* são finitamente axiomatizáveis.

Se $\Delta 1, \Delta 2, \Delta 3$ são teorias com *defaults*, onde $\Delta 1 \leq_D \Delta 2$ e $\Delta 2 \leq_D \Delta 3$, temos $\Delta 1 \leq_D \Delta 3$. Isso se deve ao fato das funções associadas a $\leq_D$ serem bijetivas e da relação $\leq$ para a lógica clássica ser transitiva.

IV.2.2 Interpretações

Sejam $\Delta 1 = (D1, W1)$ e $\Delta 2 = (D2, W2)$ teorias com *defaults* sobre linguagens *L1* e L2, respectivamente.

Definição IV.6 *I é uma interpretação* default *de* $\Delta 1$ *em* $\Delta 2$, *o que é representado por* $I : \Delta 1 \rightarrow_D \Delta 2$, *se e somente se:*

Existe uma função bijetiva $f : \mathcal{E}1 \mapsto \mathcal{E}2$, *de forma que, se* $(E1, E2)$ *é um par dessa função, existe uma interpretação clássica de E1 em E2.*

Sejam $\Delta 1$, $\Delta 2$, A3 teorias com *defaults*, onde $I1 : \Delta 1 \rightarrow_D \Delta 2$ e $I2$: AS $\rightarrow_D \Delta 3$. Logo, $I1 \cdot I2 : \Delta 1 \rightarrow_D$ A3 é obtida através da composição das interpretações clássicas entre as extensões default para $\Delta 1$, A2, $\Delta 3$, seguindo-se o sentido das funções bijetivas associadas às definições das interpretações default $I1$ e $I2$.

IV.2.3 Passo canônico

Agora já temos elementos suficientes para definirmos o que significa implementar uma, especificação $\Delta 1$ em uma especificação $\Delta 2$.

Definição IV.7 *Uma implementação de $\Delta 1 = (D1, W1)$, teoria com* defaults *sobre linguagem L1, em $\Delta 2 = (D2, W2)$, teoria com* defaults *sobre linguagem L2, é o par $< I1, e1 >$, onde $e1 : \Delta 2 \leq_D \Delta 2'$ e $I1 : \Delta 1 \rightarrow_D \Delta 2'$.*

Essa operação é o passo básico no processo de desenvolvimento de programas por refinamentos sucessivos, chamada de *passo canônico*.

Quando especificamos programas com a lógica de *defaults*, cada especificação feita com uma teoria com *defaults* corresponde a várias especificações feitas com a lógica clássica.

A construção da extensão conservativa restrita $e1$ introduz e caracteriza símbolos correspondentes aos de $\Delta 1$ em termos de eua relação com a linguagem de $\Delta 2$, formando $\Delta 2'$. A função associada à definição de extensão conservativa restrita deve ser bijetiva para continuar de acordo com a idéia de conservatividade vinda da, lógica clássica: de que não podemos provar nada mais sobre fórmulas de L2 em $\Delta 2'$ do que provamos em $\Delta 2$, como mostramos no exemplo IV.6.

A interpretação *default* entre as teorias garante que os termos de $\Delta 1$ serão representados em função dos termos de $\Delta 2'$. O predicado de relativização associado à interpretação clássica subjacente define quais objetos de $\Delta 2'$ representarão objetos de $\Delta 1$. Essa interpretação clássica subjacente garantirá, também, que os modelos das especificações embutidas em $\Delta 2'$ induzam modelos para as especificações embutidas em $\Delta 1$.

As observações do final do parágrafo anterior sugerem que, na definição de interpretação *default*, poderíamos exigir da função f apenas a injetividade e não a bijetividade, pois precisamos garantir apenas que as especificações embutidas em $\Delta 1$ tenham modelos induzidos pelos modelos das especificações embutidas em $\Delta 2'$, e não que todas as especificações embutidas em $\Delta 2'$ induzam modelos para $\Delta 1$. Porém, a bijetividade será importante para que possamos ter uma prova do teorema da modularização nesse contexto.

Independentemente do número de vezes que o passo canônico é executado, eventualmente chegaremos no momento de implementarmos nosso programa

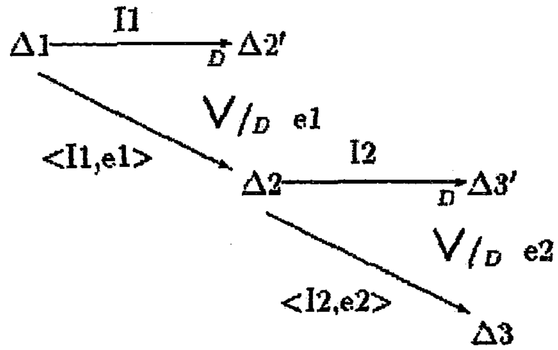

Figura IV.1: Implementações consecutivas

sobre um tipo concreto, disponível na linguagem de programação. A especificação desse tipo concreto, devido à sua natureza determinística, é feita com a lógica clássica (a não ser que esteja especificada a manipulação de erros), Um enfoque é considerar que, até esse ponto, estávamos garantindo a implementação correta de várias teorias de primeira ordem. Agora,, vamos escolher uma delas e implementá-la no tipo concreta Outro enfoque é trabalharmos com uma teoria com *defaults* que resume as extensões conservativas do tipo concreto necessárias para as implementações de todas as teorias de primeira ordem.

IV.2.4 Encadeamento de implementações

Como mencionamos na seção anterior, o passo canônico é o passo básico no desenvolvimento de! programas por refinamentos sucessivos. Assim, cada vez que construímos uma implementação, estamos mais pertos do sistema final, pronto para ser executado em um computador. Entretanto, quando aplicamos mais um refinamento, não queremos perder o trabalho feito no passo anterior. Como garantir isso? Precisamos de uma versão do teorema da modularização para a lógica de *defaults* para responder essa pergunta,

Observe o esquema da figura IV.1, onde, após termos implementado $\Delta 1$ em $\Delta 2$, implementamos agora $\Delta 2$ em $\Delta 3$, teoria com *defaults* sobre linguagem L3.

É desejável que, de imediato, a partir das implementações anteriores, possamos construir uma implementação de AI em termos de $\Delta 3$. Isto é, não queremos refazer nenhum trabalho criativo já feito nos passos anteriores. Em outras palavras, queremos garantir que, no esquema da figura IV.2, exista uma teoria com *defaults* que substitua o ?. A interpretação *default* 112 seria a composição de $I1$ e $I2'$, e e12 seria a composição de e3 e $e1'$ (lembrar que a relação conservativa restrita é transitiva).

Analisando somente parte do diagrama da figura IV.2, queremos ga-

$$
\begin{tikzcd}
\Delta 1 \arrow[r, "I1", "D"'] \arrow[dr, "{<I1,e1>}"'] \arrow[ddrr, "{<I12,e12>}"'] \arrow[rr, bend left, "I12", "D"'] & \Delta 2' \arrow[r, "I2'", "D"'] \arrow[d, "\vee/_D\ e1"] & ? \arrow[d, "\vee/_D\ e1'"] \arrow[dd, bend left, Rightarrow, "e12", "D"'] \\
 & \Delta 2 \arrow[r, "I2", "D"'] \arrow[dr, "{<I2,e2>}"'] & \Delta 3' \arrow[d, "\vee/_D\ e2"] \\
 & & \Delta 3
\end{tikzcd}
$$

Figura IV.2: Composição das implementações

rantir que existe uma teoria com *defaults* $\Delta 4$, uma interpretação *default* $I2'$ e uma extensão conservativa restrita $e1'$, como mostrado abaixo:

$$
\begin{tikzcd}
\Delta 2' \arrow[r, "I2'", "D"'] \arrow[d, "\vee/_D\ e1"] & \Delta 4 \arrow[d, "\vee/_D\ e1'"] \\
\Delta 2 \arrow[u, bend left, "f"] \arrow[r, "I2", "D"'] \arrow[r, bend right, "g"'] & \Delta 3'
\end{tikzcd}
$$

A extensão conservativa restrita $e1$ garante que existe uma função bijetiva $f : \mathcal{E}_2 \mapsto \mathcal{E}_{2'}$, tal que se $(E2, E2')$ é um par dessa função, então $E2 \leq E2'$.

$I2$ garante que existe uma função bijetiva $g : \mathcal{E}_2 \mapsto \mathcal{E}_{3'}$, tal que se $(E2, E3')$ é *um* par dessa função, então existe uma interpretação clássica $I2$, tal que $I2 : E2 \rightarrow E3'$.

Considere uma extensão *default* $E2'$ para $\Delta 2'$. Como a função f é bijetiva, sua inversa nos fornece uma extensão *default* $E2$ para $\Delta 2$, tal que $E2 \leq E2'$. A função g nos fornece uma extensão *default* $E3'$ para $\Delta 3'$, tal que existe uma interpretação clássica $I2$, tal que $I2 : E2 \rightarrow E3'$. O teorema da modularização da lógica clássica garante que existe uma teoria de primeira ordem $E4$ com as seguintes características:

- A linguagem de E4 é L4, onde L4 é igual a L3' mais símbolos correspondentes aqueles introduzidos em L2 via $e1$;
- Existe uma interpretação clássica $I2' : E2' \rightarrow E4$. O mapeamento associado a $I2'$ é o mapeamento π associado a $I2$ estendido com a tradução dos novos símbolos introduzidos via $e1$ nos novos símbolos correspondentes introduzidos em L4, e o predicado de relativização associado a $I2'$ é o mesmo de $I2$;

- Com o mapeamento do passo anterior, podemos traduzir fórmulas de L2' em fórmulas de L4. Assim, podemos traduzir as fórmulas introduzidas em $E2$ para se obter $E2'$. $E4$ é formado acrescentando-se essas traduções à $E3'$. Temos que $E3' \preceq E4$.

Em forma de diagrama:

$$\begin{array}{ccc} E2' & \xrightarrow{I2'} & E4 \\ \vee/\ e1 & & \vee/\ e1' \\ E2 & \xrightarrow{I2} & E3' \end{array}$$

Seja $\mathcal{F}$ o conjunto de teorias de primeira urdem obtidas como no parágrafo acima a partir das extensões *default* para $\Delta 2'$. Vamos mostrar que o conjunto $\mathcal{F}$ é um conjunto de teorias de primeira ordem que não estão contidas dentro das outras. Para isso, sejam f e g as bijeções relacionadas a e1 e $I2$, respectivamente, Por contradição, suponha que $E2'*$ e $E2'$, extensões *default* para $\Delta 2'$, gerem, respectivamente, $E4'$ e $E4$ pelo processo descrito acima, tal que $E4 \subseteq E4'$. Sejam $E2*$ e $E2$ as extensões *default* para $\Delta 2$, correspondentes a $E2'*$ e $E2'$, respectivamente, dadas por f. Sejam $E3'*$ e $E3'$ as extensões *default* para $\Delta 3'$ correspondentes a $E2*$ e E2, respectivamente, dadas por g. Temos que $E3' \preceq E4$ e que $E3'* \preceq E4'$. Como $E3'$ e $E3'*$ são extensões *default* para $\Delta 3'$, pela minimalidade de extensões *default*, existem algum elementos que as diferenciam, Sem perda de generalidade, considere uma wff G, tal que

$$G \in E3' - E3'*, \text{i.e.}, E3' \models G \text{ e } E3'* \not\models G.$$

Como $E3' \subseteq E4$, temos $E4 \models G$. Como $E4 \subseteq E4'$, temos que $E4' \models G$. Como $E3'* \preceq E4'$ e $G \in L3'$, temos que $E3'* \models G$. Contradição. Logo, para cada tripla $E2'$, $E2$ e $E3'$ obtemos um $E4$ diferente, que não está contido em nenhum dos outros $E4$'s gerados pelas outras triplas. Temos, então, caracterizada uma função bijetiva $h : \mathcal{E}_{2'} \mapsto \mathcal{F}$, de forma que se $(E2', F)$ é um par dessa função, então existe uma interpretação clássica $I2'$, tal que $I2' : E2' \rightarrow F$.

Seja, agora, $E3'$ uma extensão *default* para $\Delta 3'$. Como g é bijetiva,[1] sua inversa nos fornece uma extensão *default* E2 para AS, tal que existe uma interpretação clássica $I2'$, tal que $I2' : E2 \rightarrow E3'$. A função f nos fornece um $E2'$, extensão *default* para $\Delta 2'$, tal que $E2 \preceq E2'$. O teorema da modularização para a lógica clássica nos fornece um $E4$ como descrito acima. Como f e g são bijetivas, os mesmos $E4$'s da análise feita nos parágrafos anteriores aparecerão. Logo, fica caracterizada uma função bijetiva i : $\mathcal{E}_{3'} \mapsto \mathcal{F}$, de forma que se $(E3', F)$ é *um* par dessa função, então $E3' \preceq$ F.

O diagrama abaixo ilustra o resultado obtido:

[1] É por isso que exigimos que a função associada à interpretação *default* seja bijetiva, e não apenas injetiva.

$$
\begin{tikzcd}
\mathcal{E}2' \arrow[rrr, dashed, "h"] \arrow[ddd, dash, "f"'] & & & \mathcal{F} \\
& E2' \arrow[r, "I2'"] & E4 & \\
& E2 \arrow[r, "I2"] \arrow[u, phantom, "V/\ e1"] & E3' \arrow[u, phantom, "V/\ e1'"] & \\
\mathcal{E}2 \arrow[rrr, "g"'] & & & \mathcal{E}3' \arrow[uuu, dotted, "i"']
\end{tikzcd}
$$

Teríamos uma prova para o teorema da modularização nesse contexto, se $\mathcal{F}$ fosse o conjunto de extensões *default* para uma teoria com *defaults* $\Delta 4$. Infelizmente, não temos como garantir isso. Vamos analisar qual o motivo.

Se $\mathcal{F}$ fosse finito e se cada $F_i \in \mathcal{F}$ fosse finitamente axiomatizável, nossos problemas acabariam, como mostra a proposição abaixo.

Proposição IV.2 *Seja $\mathcal{F}$ um conjunto finito de teorias de primeira ordem finitamente axiomatizáveis que não estão contidas uma dentro das outras. Então existe uma teoria com* defaults *que possui $\mathcal{F}$ como o conjunto de suas extensões* default.

Prova:

$\mathcal{F} = \{F_1, \ldots, F_n\}$ e cada F_i é finitamente axiomatizável.

Vamos supor que S_i é a conjunção dos axiomas de F_i. Isto é,

$$Th(S_i) = F_i, 0 \leq i \leq n.$$

Consideremos a teoria com *defaults* $\Delta 4 = (D4, \{\})$, onde:

$$D4 = \left\{ \frac{: S_0 \wedge \neg(S_1 \vee \ldots \vee S_n)}{S_0}, \frac{: S_n \wedge \neg(S_0 \vee \ldots \vee S_{n-1})}{S_n}, \right.$$
$$\left. \frac{: S_j \wedge \neg(S_0 \vee \ldots \vee S_{j-1} \vee S_{j+1} \vee \ldots \vee S_n)}{S_j}, \text{ para } 0 < j < n \right\}.$$

Quando um *default* é disparado, ele bloqueia todos os outros. Assim, somente um S_i pode estar em uma extensão *default*. Mais ainda,,a justificativa desse *default* disparado não é negada por $Th(S_i) = F_i$. Para percebermos isso, suponha, por contradição, que $S_j \in Th(S_i)$, para algum j tal que $0 \leq j \leq n$ e $j \neq i$, negando a justificativa do *default disparado*. Contudo, teríamos $Th(S_j) = F_j \subseteq Th(S_i) = F_i$, o que nega o fato dos F_i's possuirem elementos que os diferenciam uns dos outros. Logo, $\Delta 4$ possui n extensões *default*:

$$E_i = Th(S_i) = F_i,\ 0 \leq i \leq n$$

e é justamente a teoria com *defaults* que estamos procurando.

C.Q.D.

Entretanto, nem sempre podemos afirmar que esse conjunto é finito e finitamente axiomatizável. Abaixo, analisamos um subcaso em que isso acontece.

Se $\Delta 1$, A%e $\Delta 3$ forem teorias com *defaults* com a seguinte forma: $\Delta = (D, W)$, onde D é um conjunto com um número finito de *defaults* fechados e W é uma teoria de primeira ordem finitamente axiomatizável; se as extensões conservativas clássicas subjacentes a $e1$ e e2 forem compostas por um número finito de axiomas, então o conjunto $\mathcal{F}$ é finito e composto por teorias de primeira ordem finitamente axiomatizáveis, garantindo a existência de um A4 com um número finito de *defaults*.

Isso acontece, pois cada extensão *default* para essas teorias com *defaults* será finitamente axiomatizável, já que W é finitamente axiomatizável e apenas um número finito de consequentes de *defaults* será acrescido à W. Isto é,

$$Th(W \cup \text{CONSEQUENTES}(GD(E, \Delta)))$$

é finitamente axiomatizável. Mais ainda, como o número de *defaults* é finito, o número de extensões *default* para essa teorias com *defaults* também será, finito.

Assim, podemos considerar os pares $(E2, E2') \in f$ e $(E2, E3') \in$ g de uma forma mais restrita. E2 é finitamente axiomatizável. Como $e1$ introduz apenas um número finito de axiomas, $E2'$ 6, também, finitamente axiomatizável. Da mesma forma, $E3'$ é uma extensão conservativa clássica de uma extensão *default* $E3$ para $\Delta 3$. Como E3 é finitamente axiomatizável c $e2$ introduz um número finito de axiomas, $E3'$ é finitamente axiomatizável. Por fim, a extensão conservativa $e1'$ é composta pela tradução dos axiomas de $e1$, *e* por isso é também composta por um número finito de axiomas. Como, por construção, $e1' : E3' \leq E4$, $E4$ A finitamente axiomatizável.

Como o número de extensões *default* para 82, $\Delta 2'$ e $\Delta 3'$ é finito, o número de $E4$'s contruídos também será finito. Logo, $\mathcal{F}$ é um conjunto composto de um número finito de. teorias da primeira ordem finitamente axiomatizáveis.

O caso analisado, apesar de mais específico, não é tão distante da realidade. Segundo nosso ponto de vista, as teorias com *defaults* são resumos de especificações de programas. Essas especificações, geralmente, são apresentadas por um conjunto finito de axiomas, sendo, evidentemente, finitamente axiomatizáveis. Os métodos que conhecemos para construir extensões clássicas que são garantidamente conservativas clássicas acrescentam um axioma por símbolo novo introduzido na linguagem. Na maioria absoluta das vezes, o número de símbolos novos é finito, sendo também finito o número de axiomas acrescentados.

No escopo desse trabalho, não conseguimos resolver o problema para o caso geral. Conseguimos, ao menos, um conjunto de teorias de primeira ordem que faz o papel dessa teoria com *defaults* $\Delta 4$ (não devemos nos esquecer que uma teoria com *defaults* pode ser vista como um resumo de teorias de primeira ordem).

No próximo capítulo, apresentaremos formas específicas de se estender teorias com *defaults* da forma que essa extensão seja conservativa restrita, e formas de se interpretar teorias com *defaults* de forma que o resultado seja uma interpretação *default*. Para esses casos restritos, provaremos o teorema da modularização.

Capítulo V

Garantindo Extensões Conservativas Restritas e Interpretações Default

Dadas duas teorias com *defaults*, é difícil afirmarmos, sem uma análise mais profunda, que uma é extensão conservativa restrita da outra, ou se existe uma interpretação *default* entre elas. Estudaremos, nesse capítulo, uma maneira de estender uma teoria com *defaults* de modo que o resultado seja uma extensão conservativa restrita desta, e uma forma de interpretar uma teoria com *defaults* em outra, de modo que o resultado seja uma interpretação default da primeira na segunda.

Nossos resultados se restringirão a teorias com *defaults* fechados sem pré-requisitos. Isso se deve ao fato de usarmos em nossas provas uma caracterização de extensões *default*, apresentada na próxima seção, que só vale no caso em que os *defaults* não possuem pré-requisitos. Na verdade, no caso de extensões conservativas restritas, possuímos provas que valem para *defaults* arbitrários, que apresentamos em um apêndice, mas preferiremos a versão mais restrita por três motivos:

- As provas são muito mais simples;
- Quando tratamos de interpretações *default*, não possuímos provas para *defaults* arbitrários. Assim, seríamos obrigados a nos deter nessas teorias com *defaults* mais restritos ao tratarmos do passo canônico;
- Guerreiro et alii [14] argumenta que podemos transformar um *default* $\frac{\alpha:\beta_1,\ldots,\beta_m}{\omega}$ em um default $\frac{:\alpha\to\beta_1,\ldots,\alpha\to\beta_m}{\alpha\to\omega}$, pois as duas versões capturam a mesma informação. A única diferença seria que, no primeiro caso, daríamos preferência a ω, e no segundo a $\alpha\to\omega$.

V.1 Caracterização alternativa de extensões default

Apresentaremos, agora, uma caracterização alternativa de extensões *default* para teorias com *defaults* fechados sem pré-requisitos, que usaremos nas provas das proposições do resto do capítulo.

Sejam D um conjunto de *defaults* fechados sem pré-requisitos e W um conjunto de fórmulas fechadas.

Definição V.1 *D é aplicável ao W, o que é representado por $A(D, W)$, se e somente se para qualquer $d = \frac{:\beta_1,\ldots,\beta_m}{\omega} \in D$,*

$$W \cup CONSEQUENTES(D) \not\models \neg\beta_i, 1 \leq i \leq m$$

Definição V.2 *Um conjunto de fórmulas T é* estático *em relação à uma teoria com* defaults *fechados sem pré-requisitos $\Delta = (D, W)$ se e somente se:*

$$Para\ todo\ d = \frac{:\beta_1,\ldots,\beta_m}{\omega} \in D,\ se\ \neg\beta_1,\ldots,\neg\beta_m \notin T,\ então\ \omega \in T.$$

Proposição V.1 *Seja E uma extensão* default *para $\Delta = (D, W)$, teoria com* defaults *fechados sem pré-requisitos.*

E é um conjunto de fórmulas estático em relação à Δ.

Prova: Pelo teorema III.2, $E = Th(W \cup \text{CONSEQUENTES}(GD(E, \Delta)))$,onde

$$GD(E, \Delta) = \left\{\frac{:\beta_1,\ldots,\beta_m}{\omega} \in D / \neg\beta_1,\ldots,\neg\beta_m \notin E\right\}.$$

Logo, para todo $d = \frac{:\beta_1,\ldots,\beta_m}{\omega} \in D$, tal que $\neg\beta_1,\ldots,\neg\beta_m \notin E$, $d \in GD(E, \Delta)$. Então, obviamente, $\omega \in E$. Logo, E é estático em relação à Δ. **C.Q.D.**

Proposição V.2 *Seja uma teoria com* defaults *fechados sem pré-requisitos $\Delta = (D, W)$. Seja $D' \subseteq D$, tal que $A(D', W)$. Seja*

$$E = Th(W \cup CONSEQUENTES(D')).$$

Se E é estático em relação à Δ, então E é extensão default *para Δ.*

Prova:

$E = Th(W' \cup \mathrm{CONSEQUENTES}(D'))$. Queremos mostrar que E é extensão *default* para A. Para isso, mostraremos que $E = \bigcup_{i=0}^{\infty} E_i$, onde:

$$\begin{aligned} E_0 &= W, \text{ e para todo } i \geq 0: \\ E_{i+1} &= Th(E_i) \cup \left\{ \omega / \frac{:\beta_1, \ldots, \beta_m}{\omega} \in D, \text{ onde } \neg\beta_1, \ldots, \neg\beta_m \notin E \right\}. \end{aligned}$$

- $\bigcup_{i=0}^{\infty} E_i \subseteq E$:

  Por indução em i, provaremos que $E_i \subseteq E$, para $i \geq 0$:

  Base: i=0: $E_0 = W \subseteq E$;

  h.i: $E_i \subseteq E$;

  p/ i+1:

  $E_{i+1} = Th(E_i) \cup \left\{ \omega / \frac{:\beta_1, \ldots, \beta_m}{\omega} \in D, \text{ onde } \neg\beta_1, \ldots, \neg\beta_m \notin E \right\}$.

  Como E é estático e $\neg\beta_1, \ldots, \neg\beta_m \notin E$, então $\omega \in E$. Além disso, por hipótese de indução, $E_i \subseteq E$. Como E é teoria de primeira ordem, $Th(E_i) \subseteq E$. Logo, $E_{i+1} \subseteq E$, completando a indução.

  Então, $\bigcup_{i=0}^{\infty} E_i \subseteq E$;

- $E \subseteq \bigcup_{i=0}^{\infty} E_i$:

  Lembremo-nos que $E = Th(W' \cup \mathrm{CONSEQUENTES}(D'))$ e $W' \subseteq E_0$.

  Como $A(D', W')$, então para qualquer $d = \frac{:\beta_1, \ldots, \beta_m}{\omega} \in D'$, $\neg\beta_1, \ldots, \neg\beta_m \notin E$. Logo,

  $$\mathrm{CONSEQUENTES}(D') \subseteq E_1.$$

  Então, $E \subseteq \bigcup_{i=0}^{\infty} E_i$.

Dessa forma, $Th(W \cup \mathrm{CONSEQUENTES}(D'))$ é extensão *default* para Δ.

C.Q.D.

Para entendermos melhor porque é necessário que os *defaults* sejam sem pré-requisitos, considere a teoria com *defaults* fechados arbitrários $A = (D, W)$, onde

$$W = \{A\} \text{ e } D = \left\{ \frac{B : C}{C}, \frac{C : B}{B} \right\}.$$

Acrescente à definição de $A(D, W)$ a condição:

$$W \cup \mathrm{CONSEQUENTES}(D) \models \alpha, \text{ para todo } default\ \frac{\alpha:\beta_1, \ldots, \beta_m}{\omega} \in D,$$

e inclua $\alpha \in T$ como pré-condição de estático. Note que $A(D, W)$, pois $\{A, B, C\} \models \{B, C\}$ e $\{A, B, C\} \not\models \{\neg B, \neg C\}$, e que $Th(\{A, B, C\})$ é estático em relação à Δ, mas não é extensão *default* para Δ.

Já a teoria com *defaults* abaixo, que já vimos não possuir extensões *default*, ilustra porque é necessário que o conjunto de fórmulas seja estático para ser uma extensão *default*:

$$W = \{\},\ D = \left\{d1 = \frac{: A \wedge \neg B}{A}, d2 = \frac{: B \wedge \neg C}{B}, d3 = \frac{: C \wedge \neg A}{C}\right\}.$$

O conjunto de *defaults* $\{d1\}$ é um subconjunto maximal de D que é aplicável ao W $(A(d1, W))$. Contudo, $Th(\{A\})$ não é uma extensão *default*, pois não é estático, já que $\{A\} \not\models \neg\{B \wedge \neg C\}$, mas $B \notin Th(\{A\})$.

Proposição V.3 *Para cada E extensão* default *para $A = (D, W)$ teoria com* defaults *fechados tem pré-requisitos, $GD(E, \Delta)$ é tal que $A(GD(E, \Delta), W)$.*

Prova:

Pelo teorema III.2, $E = Th(W \cup \text{CONSEQUENTES}(GD(E, \Delta)))$, onde

$$GD(E, \Delta) = \left\{\frac{:\beta_1, \ldots, \beta_m}{\omega} \in D / \neg\beta_1, \ldots, \neg\beta_m \notin E\right\}$$

Logo, vemos que para todo $\frac{:\beta_1, \ldots, \beta_m}{\omega} \in GD(E, \Delta)$:

$W \cup \text{CONSEQUENTES}(GD(E, \Delta)) \not\models \neg\beta_i, 1 \leq i \leq m.$

Então, temos $A(GD(E, \Delta), W)$.

C.Q.D.

V.2 Estendendo teorias com *defaults*

No capítulo II, estudamos maneiras de estender teorias de primeira ordem com símbolos de função, de forma que o resultado fosse uma extensão conservativa clássica da teoria original. Agora, faremos o mesmo com teorias com *defaults*. Deter-nos-emos apenas no caso em que, dada uma teoria com *defaults* fechados sem pré-requisitos $\Delta = (D, W)$, a extensão dar-se-á apenas no W, deixando o conjunto de *defaults* D inalterado. Queremos que a teoria com *defaults* $\Delta' = (D, W')$, onde $Th(W')$ é uma extensão clássica de $Th(W)$, seja tal que $\Delta \leq_D \Delta'$.

Provaremos que, se estendermos a teoria com *defaults* fechados sem pré-requisitos $\Delta = (D, W)$, formando $\Delta' = (D, W + A)$, e se $Th(W) \leq Th(W + A)$, então $\Delta \leq_D \Delta'$, onde A é uma fórmula fechada.

Antes, a proposição a seguir apresenta um resultado sobre extensões conservativas clássicas que será importante nas provas seguintes:

Proposição V.4 *Sejam S e H conjuntos de fórmulas de uma linguagem L e S' um conjunto de fórmulas de uma linguagem L', tal que $L \subseteq L'$. Se $Th(S) \leq Th(S')$, então $Th(S + H) \leq Th(S' + H)$.*

Prova: Seja F uma fórmula fechada de L tal que

$$S' + H \models F.$$

Por compacidade, existe um $I \subseteq H$, sendo I finito, tal que:

$$S' + I \models F.$$

Logo, $S' \models I \rightarrow F$.

Como $Th(S) \leq Th(S')$ e $I \rightarrow F \in L$:

$$S \models I \rightarrow F.$$

Então, $S + I \models F$.

Por monotonicidade,

$$S + H \models F.$$

Dessa forma, $Th(S + H) \leq Th(S' + H)$.

C.Q.D.

Consideremos, primeiro, a seguinte situação. Sejam $\Delta 1 = (D, W1)$ e $\Delta 2 = (D, W2)$ teorias com *defaults* fechados sem pré-requisitos sobre linguagens L1 e L2, respectivamente, sendo que $L1 \subseteq L2$. $W1$ e $W2$ são apresentações de teorias, $W2 = W1 + A$, onde A é um novo axioma, e $Th(W1) \leq Th(W2)$.

Proposição V.5 *Seja $E1$ uma extensão* default *para $\Delta 1$. Seja $E2 = Th(E1 + A)$. $E2$ é extensão* default *para $\Delta 2$.*

Prova:

Como $E1$ é extensão *default* para $\Delta 1 = (D, W1)$, pela proposição V.3, temos que $A(GD(E1, \Delta 1), W1)$. Pelo teorema 111.2, temos que

$$E1 = Th(W1 \cup \text{CONSEQUENTES}(GD(E1, \Delta 1))).$$

Pela proposição V.1, $E1$ é estático em relação à $\Delta 1$.

Como $A(GD(E1,\Delta 1), W1)$, por definição temos que para qualquer *default* $\frac{:\beta_1,\ldots,\beta_m}{\omega} \in GD(E1,\Delta 1)$:

$$W1 + \text{CONSEQUENTES}(GD(E1,\Delta 1)) \not\models \neg\beta_i, 1 \leq i \leq m.$$

Como $Th(W1) \leq Th(W2)$ e $\text{CONSEQUENTES}(GD(E1,\Delta 1)) \subseteq L1$, pela proposição V.4 temos que:

$$Th(W1 + \text{CONSEQUENTES}(GD(E1,\Delta 1))) \leq$$
$$Th(W1 + A + \text{CONSEQUENTES}(GD(E1,\Delta 1))).$$

Isto é, $E1 \leq E2$.

Como $\neg\beta_1,\ldots,\neg\beta_m \in L1$:

$$W1 + A + \text{CONSEQUENTES}(GD(E1,\Delta 1)) \not\models \neg\beta_i, 1 \leq i \leq m,$$

de onde vemos que $A(GD(E1,\Delta 1), W1 + A)$.

Provaremos agora que

$$E2 = Th(W1 + A + \text{CONSEQUENTES}(GD(E1,\Delta 1)))$$

é estático em relação à $\Delta 2$. Para isso, vamos supor, por contradição, que existe um $d = \frac{:\beta_1,\ldots,\beta_m}{\omega} \in D$, tal que $\neg\beta_1,\ldots,\neg\beta_m \notin E2$, mas $\omega \notin E2$.

Como $E1 \subseteq E2$, então $\neg\beta_1,\ldots,\neg\beta_m, \omega \notin E1$. Portanto, E1 não é estático em relação à $\Delta 1$, o que é uma contradição.

Logo, para todo $d = \frac{:\beta_1,\ldots,\beta_m}{\omega} \in D$, se $\neg\beta_1,\ldots,\neg\beta_m \notin E2$, então $\omega \in E2$. Dessa forma, $E2$ é estático em relação à $\Delta 2$.

Pela proposição V.2,

$$Th(W1 + A + \text{CONSEQUENTES}(GD(E1,\Delta 1))) = Th(E1 + A) = E2$$

é extensão *default* para $\Delta 2$.

C.Q.D

A proposição V.5 afirma que, ao acrescentarmos o axioma A à uma extensão *default* para $\Delta 1$, as consequências dessa união formam uma extensão *default* para $\Delta 2$. Isso define uma função $f : \mathcal{E}1 \mapsto \mathcal{E}2$, onde para cada par $(E1, E2)$ de f, E2 é uma extensão conservativa clássica de E1.

Proposição V.6 *f é injetiva.*

Prova:

Vamos supor, por contradição, que f não é injetiva. Logo, existem pelo menos duas extensões *default* diferentes para $\Delta 1$, que chamaremos de $E1$ e $E1'$, tal que

$$Th(E1 + A) = Th(E1' + A) = E2,$$

sendo que $E2$ é extensão *default* para $\Delta 2$ e $E1 \leq E2$ e $E1' \leq E2$.

Como, pela minimalidade de extensões *default*, uma extensão *default* nunca está contida em outra, cada extensão *default* para uma teoria com *defaults* possui elementos que a diferencia das outras.

Sem perda de generalidade, seja G uma wff tal que $E1' \models G$ e $E1 \not\models G$ (G é um elemento que diferencia $E1'$ de $E1$). Como $E1' \subseteq E2$, temos que

$$E2 \models G.$$

Contudo, $E1 \not\leq E2$, o que 6 uma contradição. Portanto, f é injetiva.

C.Q.D.

Consideremos, agora, a seguinte situação: sejam $\Delta 1$ e $\Delta 2$ os mesmos da discussão anterior. Seja E2 uma extensão *default* para $\Delta 2$. Pelo teorema III.2,

$$E2 = Th(W1 + A + \text{CONSEQUENTES}(GD(E2, \Delta 2))).$$

Queremos que $Th(W1 + \text{CONSEQUENTES}(GD(E2, \Delta 2)))$ seja uma extensão *default* para $\Delta 1$.

Proposição V.7 *$Th(W1 + CONSEQUENTES(GD(E2, \Delta 2)))$ é extensão default para* $\Delta 1$.

Prova: Como $E2$ é extensão *default* para A2 = $(D, W1 + A)$, pela proposição V.3, temos que $A(GD(E2, \Delta 2), W1 + A)$. Pela proposição V,1, E2 é estático em relação à $\Delta 2$.

Como $A(GD(E2, \Delta 2), W1 + A)$, temos que para qualquer $\frac{:\beta_1, \ldots, \beta_m}{\omega} \in GD(E2, \Delta 2)$:

$$W1 + A + \text{CONSEQUENTES}(GD(E2, \Delta 2)) \not\models \neg\beta_i, 1 \leq i \leq m.$$

Dessa forma:

$$W1 + \text{CONSEQUENTES}(GD(E2, \Delta 2)) \not\models \neg\beta_i, 1 \leq i \leq m,$$

de onde vemos que $A(GD(E2, \Delta 2), W1)$.

Seja $E1 = Th(W1 + \text{CONSEQUENTES}(GD(E2, \Delta 2)))$. Provaremos que $E1$ é estático em relação à $\Delta 1$. Para isso, vamos supor, por contradição, que *existe* um d = $\frac{:\beta_1, \ldots, \beta_m}{\omega} \in D$, tal que $\neg\beta_1, \ldots, \neg\beta_m \notin E1$, mas ω 4 $E1$.

Note que $Th(W1) \leq Th(W2)$ e CONSEQUENTES($GD(E2, \Delta 2)$) $\in L1$, o que implica, pela proposição V.4:

$$Th(W1 + \text{CONSEQUENTES}(GD(E2, \Delta 2))) \leq$$
$$Th(W1 + A + \text{CONSEQUENTES}(GD(E2, \Delta 2))),$$

isto é, $E1 \leq E2$.

Como $\neg\beta_1, \ldots, \neg\beta_m, \omega \in L1$, então $\neg\beta_1, \ldots, \neg\beta_m, \omega \notin E2$, de onde concluímos que E2 não é estático em relação à $\Delta 2$, o que é uma contradição.

Logo, para todo $d = \frac{:\beta_1, \ldots, \beta_m}{\omega} \in D$, se $\neg\beta_1, \ldots, \neg\beta_m \notin E1$, então $\omega \in E1$. Dessa forma, $E1$ é estático em relação à $\Delta 1$.

Pela proposição V.2,

$$E1 = Th(W1 + \text{CONSEQUENTES}(GD(E2, \Delta 2)))$$

é extensão *default* para $\Delta 2$.

C.Q.D

A proposição V.7 afirma que, ao retirarmos de uma extensão *default* para $\Delta 2$ o axioma A, obtemos uma extensão default para $\Delta 1$. Isso define uma função $g : \mathcal{E}2 \mapsto \mathcal{E}1$, onde para cada par $(E2, E1)$ de g, E2 é extensão conservativa clássica de E1.

Proposição V.8 *g é injetiva*

Prova:

Como E2 é extensão *default* para $\Delta 2$:

$$E2 = Th(W1 + \text{A} + \text{CONSEQUENTES}(GD(E2, \Delta 2))).$$

Como, pela minimalidade de extensões *default*, uma extensão default nunca é subconjunto próprio de outra, sempre haverá algum elemento que diferenciará uma extensão *default* de outra. Como $W1 + A$ está contido em todas as extensões *default* para $\Delta 2$, percebemos que esse elemento diferenciador tem que estar nos consequentes dos *defaults* geradores da extensão *default*. Assim, esse elemento diferenciador também estará em $Th(W1 + \text{CONSEQUENTES}(GD(E2, \Delta 2)))$. Portanto, g é injetiva.

C.Q.D.

Proposição V.9 *Sejam $\Delta 1 = (D, W1)$ e $\Delta 2 = (D, W2)$ teorias com* defaults *fechados sem pré-requisitos sobre linguagens L1 e L2, respectivamente, onde $W2 = W1 + A$, onde A é uma fórmula fechada. Se $Th(W1) \leq Th(W2)$, então $\Delta 1 \leq_D \Delta 2$.*

Prova:

Note que se $\Delta 1$ ou $\Delta 2$ for incoerente, então ambos são incoerentes, pois se $\Delta 2$ possui uma extensão *default*, pela proposição V.7, $\Delta 1$ também possui, e se $\Delta 1$ tem uma extensão *default*, pela proposição V.5, $\Delta 2$ também tem. Assim, é vacuamente verdadeiro que $I : \Delta 1 \leq_D \Delta 2$.

Vamos supor, agora, que $\Delta 1$ e $\Delta 2$ possuem extensões *default*.

As proposições V.5 e V.7 nos fornecem, respectivamente, as funções $f : \mathcal{E}1 \mapsto \mathcal{E}2$ e $g : \mathcal{E}2 \mapsto \mathcal{E}1$, onde para cada par $(E1, E2) \in f$ e $(E2, E1) \in g$, E2 é extensão conservativa clássica de E1. As proposições V.6 e V.8 garantem que f e g são injetivas.

Vamos tomar um $(E1, E2) \in f$. Sabemos que $E2 = Th(E1 + A)$. Logo, $(E2, E1) \in g$.

Tomemos, agora, um $(E2, E1) \in g$. Sabemos que

$$E2 = Th(W1 + A + \text{CONSEQUENTES}(GD(E2, \Delta 2)))$$

e $E1 = Th(W1 + \text{CONSEQUENTES}(GD(E2, \Delta 2)))$. Portanto, $E2 = Th(E1 + A)$, implicando $(E1, E2) \in f$.

Isso caracteriza uma bijeção de $\mathcal{E}1$ em $\mathcal{E}2$, onde para cada par $(E1, E2)$ da bijeção, $E1 \leq E2$.

Logo, $\Delta 1 \leq_D$ AS,

C.Q.D.

V.3 Interpretações *default* garantidas

Estudaremos, agora, uma maneira de traduzir teorias com *defaults* de modo que o resultado seja uma interpretação *default*.

Sejam $\Delta 1 = (D1, W1)$ e $\Delta 2 = (D2, W2)$ teorias com *defaults* fechados sem pré-requisitos sobre linguagens L1 e L2, respectivamente, A uma wff de L1 e $d = \frac{:\beta_1, \ldots, \beta_m}{\omega}$ um *default* de D1. W1 e W2 são apresentações de teorias.

Seja F uma interpretação clássica de L1 em L2. Logo, $F(A)$ será o resultado da tradução de A por F, e $F(d)$ será o default com justificativas que são o resultado das traduções de $\beta_1, \ldots, \beta_m$, e com consequente que é o resultado da tradução de ω por F $\left(F(d) = \frac{:F(\beta_1), \ldots, F(\beta_m)}{F(\omega)}\right)$. Podemos, por extensão, definir F(conjunto de wffs) e F(conjunto de *defaults*) como a resultante da aplicação de F sobre suas partes componentes.

Definição V.3 *I será uma tradução de $\Delta 1$ em $\Delta 2$ se:*

- *I for uma interpretação clássica de Th(W1) em Th(W2);*
- $I(D1) = D2$;
-
  - *para* cada $D' \subseteq D1$, se $A(D', W1)$, *então* $A(I(D'), W2)$;
  - para cada $D' \subseteq D2$, se $A(D', W2)$, *então* $A(G, W1)$, *para algum* G, tal *que* $I(G) = D'$;
- *Para todo* $D' \subseteq D2$, *se* $I(d) = \frac{:I(\beta_1),\ldots,I(\beta_m)}{I(\omega)} \in D2$, *tal que*

  $I(\neg\beta_i) \in Th(W2 + CONSEQUENTES(D'))$, *para algum* $0 \leq i \leq m$,

  então $\neg\beta_i \in Th(W1 + CONSEQUENTES(G))$, *tal que* $I(G) = D'$.

A primeira condição traz como consequência, através do teorema da interpretação, que para cada axioma θ de W1, $I(\theta) \in Th(W2)$ $[I(W1) \subseteq Th(W2)]$.

A penúltima condição afirma que, se podemos disparar um conjunto de *defaults* a partir de W1, podemos disparar a tradução do conjunto a partir de W2, e vice-versa.

A última condição impede os casos em que $W2$ consegue bloquear um *default*, mas $W1$, por ter menos restrições (lembrar que $I(W1) \subseteq Th(W2)$), não consegue. Para exemplificar esse fato, considere a teoria com *defaults* $\Delta = (D, W)$ sobre linguagem L, onde:

$$
\begin{aligned}
W &= \{p(a)\} \text{ e} \\
D &= \left\{ d1 = \frac{: q(a)}{\neg p(a)} \right\},
\end{aligned}
$$

e a teoria com *defaults* $\Delta' = (D', W')$ sobre linguagem L', onde:

$$
\begin{aligned}
W' &= \{m(b), \neg n(b), r(b)\} \text{ e} \\
D' &= \left\{ d2 = \frac{: n(b)}{\neg m(b)} \right\}.
\end{aligned}
$$

Δ não possui extensão *default* e Δ' possui $Th(W')$ como extensão *default*.

Seja $I : L \longrightarrow L'$ da seguinte forma:

- $\pi(p) = m$, $\pi(q) = n$, $\pi(a) = b$;
- o predicado de relativização associado à I é r.

Observe que $I : Th(W) \to Th(W')$, pois $W' \models \exists x(r(x))$ e $W' \models m(b) = \pi(p(a))$.

Além disso, note que $\neg A(\{d1\}, W)$ e $\neg A(\{d2\}, W')$. Contudo, $Th(W)$ não é estático em relação à h, mas $Th(W')$ é estático em relação a Δ'. Isso acontece, pois W' consegue bloquear o *default* $d2$ que causaria o problema, mas W, por ser menos rico, permite seu disparo, tornando Δ uma teoria incoerente.

Proposição V.10 *Sejam* $\Delta 1 = (D1, W1)$ *sobre linguagem L1 e* $\Delta 2 = (D2, W2)$ *sobre linguagem L2 teorias com* defaults *fechados sem pré-requisitos. Seja I uma tradução de* $\Delta 1$ *em* $\Delta 2$. *Então, para cada extensão* default $E1$ *para* $\Delta 1$, *existe uma extensão* default $E2$ *para* $\Delta 2$, *tal que I é uma interpretação clássica de* $E1$ *em* $E2$.

Prova:

Seja E1 uma extensão *default* para $\Delta 1$. Pela proposição V.3, temos que $A(GD(E1, \Delta 1), W1)$. Pelo teorema III.2, temos que

$$E1 = Th(W1 \cup \text{CONSEQUENTES}(GD(E1, \Delta 1))).$$

Pela proposição V.1, $E1$ é estático em relação à $\Delta 1$.

Pela definição de I, temos $A(I(GD(E1, \Delta 1)), W2)$. Seja

$$E2 = Th(W2 \cup \text{CONSEQUENTES}(I(GD(E1, \Delta 1)))).$$

Vamos mostrar que $E2$ é estático em relação à $\Delta 2$. Para isso, vamos supor, por contradição, que existe um *default* $\frac{:I(\beta_1), \ldots, I(\beta_m)}{I(\omega)} \in D2$, de forma que $I(\neg\beta_1), \ldots, I(\neg\beta_m) \notin E2$ e $I(\omega) \notin E2$. Como $I(W1) \subseteq Th(W2)$, $I(E1) \subseteq E2$. Logo, $\neg\beta_1, \ldots, \neg\beta_m, \omega \notin E1$, o que implica que $E1$ não é estático. Contradição. Dessa forma, $E2$ é estático em relação à $\Delta 2$. Logo, pela proposição V.2,

$$Th(W2 \cup \text{CONSEQUENTES}(I(GD(E1, \Delta 1))))$$

é extensão default para $\Delta 2$.

Sabemos que I é uma interpretação clássica de L1 em L2 e também de Li em W2. Isto é,

$$W2 \models \exists x N(x) \text{ e } W2 \models (N(x_1) \wedge \ldots \wedge N(x_n) \to N(I(f)(x_1, \ldots, x_n)),$$

para cada símbolo funcional f do alfabeto de L1, onde N C o predicado de relativização associado à i. Como

$$W2 \subseteq Th(W2 \cup \text{CONSEQUENTES}(I(GD(E1, \Delta 1)))),$$

temos também, como consequência, que I é uma interpretação clássica de L1 em $Th(W2 \cup \text{CONSEQUENTES}(I(GD(E1, \Delta 1))))$.

Observemos que:

$$I(Th(W1 \cup \text{CONSEQUENTES}(GD(E1, \Delta 1)))) \subseteq$$
$$Th(I(W1) \cup \text{CONSEQUENTES}(I(GD(E1, \Delta 1)))).$$

Além disso,

$$Th(I(W1) \cup \text{CONSEQUENTES}(I(GD(E1, \Delta 1)))) \subseteq$$
$$Th(W2 \cup \text{CONSEQUENTES}(I(GD(E1, \Delta 1)))),$$

pois $I(W1) \subseteq Th(W2)$.

Então, I é uma interpretação clássica de

$$Th(W1 \cup \text{CONSEQUENTES}(GD(E1, \Delta 1)))$$

em

$$Th(W2 \cup \text{CONSEQUENTES}(I(GD(E1, \Delta 1)))),$$

isto é, $I : E1 \rightarrow E2$.

Logo, para cada E1 extensão *default* para $\Delta 1$, existe um E2 extensão *default* para $\Delta 2$, tal que I é uma interpretação clássica de E1 em E2.

C.Q.D.

Proposição V.11 *Sejam os mesmos $\Delta 1$, $\Delta 2$ e I. Então, para cada extensão* default *E2 para $\Delta 2$, existe uma extensão* default *E1 para $\Delta 1$, tal que I é uma interpretação clássica de E1 em E2.*

Prova:

Seja E2 uma extensão *default* para $\Delta 2$. Pela proposição V.3, temos que $A(GD(E2, \Delta 2), W2)$. Pela teorema III.2,

$$E2 = Th(W2 \cup \text{CONSEQUENTES}(GD(E2, \Delta 2))).$$

Pela proposição V.1, E2 é estático em relação à 42.

Par definição, temos $A(G, W1)$, tal que $I(G) = GD(E2, \Delta 2)$. Seja

$$E1 = Th(W1 \cup \text{CONSEQUENTES}(G)).$$

Vamos provar que $E1$ é estático em relação à $\Delta 1$. Para isso, por contradição, vamos supor que existe um $d = \frac{:\beta_1, \ldots, \beta_m}{\omega} \in D1$, tal que $\neg\beta_1, \ldots, \neg\beta_m \notin E1$ e $\omega \notin E1$. Logo, $d \notin G$, pois, como $A(G, W1)$, para todo $\frac{:\beta_1, \ldots, \beta_m}{\omega} \in G$, $\neg\beta_1, \ldots, \neg\beta_m \notin E1$ e $\omega \in E1$. Dessa forma, $\frac{:I(\beta_1), \ldots, I(\beta_m)}{I(\omega)} \notin GD(E2, \Delta 2)$. Assim,

$I(\neg\beta_i) \in E2$, para algum $0 \leq i \leq m$, pois, pela definição de $GD(E2, \Delta2)$, se $I(d) = \frac{:I(\beta_1),\ldots,I(\beta_m)}{I(\omega)} \in D2$ e $\neg I(\beta_1), \ldots, \neg I(\beta_m) \notin E2$. então $I(d) \in GD(E2, \Delta2)$. Portanto, por definição de I, $\neg\beta_i \in El$, o que é uma contradição. Então, $E1$ é estático em relação à $\Delta1$.

Logo, pela proposição V.2, $Th(W1 \cup \text{CONSEQUENTES}(G))$ é uma extensão *default* para $\Delta1$.

Sabemos que I é uma interpretação clássica de L1 em W2. Como

$$W2 \subseteq Th(W2 \cup \text{CONSEQUENTES}(GD(E2, \Delta2))),$$

I é uma interpretação clássica de L1 em $Th(W2 \cup \text{CONSEQUENTES}(GD(E2, \Delta2)))$.

Observe que:

$$I(Th(W1 \cup \text{CONSEQUENTES}(G))) \subseteq Th(I(W1) \cup \text{CONSEQUENTES}(I(G))).$$

Além disso, como $I(W1) \subseteq Th(W2)$, e $I(G) = GD(E2, \Delta2)$:

$$Th(I(W1) \cup \text{CONSEQUENTES}(I(G))) \subseteq$$
$$Th(W2 \cup \text{CONSEQUENTES}(GD(E2, \Delta2))).$$

Então, I é uma interpretação clássica de

$$Th(W1 \cup \text{CONSEQUENTES}(G)) \text{ em } Th(W2 \cup \text{CONSEQUENTES}(GD(E2, \Delta2))),$$

isto é, $I : E1 \rightarrow E2$.

Portanto, para cada E2 extensão *default* para $\Delta2$, existe um E1 extensão *default* para $\Delta1$, tal que I é uma interpretação clássica de E1 em E2.

C.Q.D.

Proposição V.12 *Sejam* $\Delta1 = (D1, W1)$ e $\Delta2 = (D2, W2)$ *teorias com* defaults *fechados sem pré-requisitos. Seja I uma tradução de* $\Delta1$ *em* $\Delta2$. *Então, I será* uma *interpretação* default *de* $\Delta1$ *em* $\Delta2$ $(I : \Delta1 \rightarrow_D \Delta2)$.

Prova:

Note que se $\Delta1$ ou $\Delta2$ for incoerente, ambos são incoerentes, pois se AS possui uma extensão *default*, pela proposição *V.11*, $\Delta1$ também possui, e se $\Delta1$ tem uma extensão *default*, pela proposição V.10, $\Delta2$ também tem. Assim, é vacuamente verdadeiro que $I : \Delta1 \rightarrow_D \Delta2$ é uma interpretação *default*.

Vamos supor, agora, que $\Delta1$ e $\Delta2$ possuem extensões *default*.

A proposição V.10 afirma que existe uma função $g : C1 \mapsto E2$, de forma que, se $(E1, E2)$ é um par dessa função, I é uma interpretação clássica da $E1$ em $E2$.

Vamos provar que g é injetiva.

Para isso, por contradição, vamos supor que g não é injetiva. Então, para duas extensões *default* para $\Delta 1$, $E1$ e $E1'$, g nos fornece um mesmo $E2$, extensão *default* para $\Delta 2$. Pelo teorema II1.2, temos que

$$\begin{aligned} E1 &= Th(W1 \cup \text{CONSEQUENTES}(GD(E1, \Delta 1))) \text{ e} \\ E1' &= Th(W1 \cup \text{CONSEQUENTES}(GD(E1', \Delta 1'))). \end{aligned}$$

Observe que $E2$ é obtida acrescentando-se ao $W2$ as traduções dos consequentes dos *defaults* geradores de $E1$. Logo, g nos fornece a partir de $E1$ e $E1'$, respectivamente

$$\begin{aligned} E2 &= Th(W2 \cup \text{CONSEQUENTES}(I(GD(E1, \Delta 1)))), \text{ e} \\ E2' &= Th(W2 \cup \text{CONSEQUENTES}(I(GD(E1', \Delta 1')))), \end{aligned}$$

que são extensões *default* para $\Delta 2$.

Como g não é injetiva, $E2 = E2'$.

Pela minimalidade das extensões *default*, $E1$ e $E3'$ possuem elementos que os diferenciam. Isto é, existe ao menos uma wff F, tal que $E1 \models F$ e $E1' \not\models F$, e ao menos uma wff G, tal que $E1' \models G$ e $E1 \not\models G$.

Sabemos que, como $W1$ está contido em $E1$ e $E1'$, F e G pertencem aos consequentes dos defaults geradores de $E1$ e $E1'$, respectivamente, Como $F \in \text{CONSEQUENTES}(GD(E1, \Delta 1))$, existe um *default* $\frac{\beta_{1F}, \ldots, \beta_{mF}}{F} \in D1$, tal que $\neg\beta_{1F}, \ldots, \neg\beta_{mF}$ 4 $E1$. Como $F \notin \text{CONSEQUENTES}(GD(E1', \Delta 1'))$, então $\neg\beta_{iF} \in E1'$, para algum $1 \leq i \leq m$. Como $I(E1') \subseteq E2' = E2$, então $I(\neg\beta_{iF}) \in E2$. Logo, por definição de I,

$$\neg\beta_{iF} \in Th(W1 \cup \text{CONSEQUENTES}(GD(E1, \Delta 1))),$$

i.e., $\neg\beta_{iF} \in E1$, o que é uma contradição.

Logo, g é injetiva.

A proposição V.11 afirma que existe uma função $h : \mathcal{E}2 \mapsto \mathcal{E}1$, de forma que, se $(E2, E1)$ é um par dessa função, I é uma interpretação clássica de $E1$ em $E2$. Dado que:

- $E1$ é obtida acrescentando ao $W1$ as traduções dos consequentes dos *defaults* geradores de $E2$;
- para cada $E2$, pela minimalidade de extensões *default*, os consequentes desses *defaults* geradores sempre possuem alguns elementos que os diferenciam;
- $I(W1) \subseteq Th(W2)$;

percebemos que esses elementos diferenciadores também estão nos $E1$'s. Logo, h é injetiva.

Se $(E1, E2) \in g$, então

$$I(\text{CONSEQUENTES}(GD(E1, \Delta1))) = \text{CONSEQUENTES}(GD(E2, \Delta2)).$$

Logo, $(E2, E1) \in h$. Da mesma forma, se $(E2, E1) \in h$, então $(E1, E2) \in g$. Isso caracteriza a bijeção de $\mathcal{E}1$ em $\mathcal{E}2$, tal que, se $(E1, E2)$ é um par dessa função, I é uma interpretação clássica de E1 em E2. Logo, $I : \Delta1 \rightarrow_D \Delta2$.

C.Q.D.

Exemplo V.1 *Seja uma linguagem L1 com um único símbolo de predicado binário* pr *e uma linguagem L2 com um símbolo de predicado binário* mnr, *um símbolo de predicado unário* nat *e uma constante* zero.

Seja uma teoria com defaults $\Delta1 = (W1, D1)$ *sobre L1, onde:*

$$\begin{aligned} W1 &= \{\forall i \neg pr(i,i); \\ &\quad \forall i \forall j \forall k[(pr(i,j) \wedge pr(j,k)) \rightarrow pr(i,k)]\} \end{aligned}$$

$$D1 = \left\{ \delta_1 = \frac{:\forall i \forall j (pr(i,j) \vee i=j \vee pr(j,i))}{\forall i \forall j (pr(i,j) \vee i=j \vee pr(j,i))} \right\}$$

Essa teoria com defaults *possui uma única extensão* default

$$E1 = Th(W1 \cup \{\forall i \forall j (pr(i,j) \vee i = j \vee pr(j,i))\})$$

que especifica uma ordem total.

Vamos interpretar $\Delta1$ *em* $\Delta2 = (W2, D2)$, *teoria com* defaults *sobre L2, onde:*

$$\begin{aligned} W2 &= \{zero = i \vee mnr(zero, i); \\ &\quad \forall i \forall j \forall k[(mnr(i,j) \wedge mnr(j,k)) \rightarrow mnr(i,k)] \\ &\quad \forall i \forall j (\neg mnr(i,j) \leftrightarrow i = j \vee mnr(j,i)) \\ &\quad \forall x nat(x)\} \end{aligned}$$

$$D2 = \left\{ \delta_2 = \frac{:\forall x \forall y[(nat(x) \wedge nat(y)) \rightarrow mnr(x,y) \vee x=y \vee mnr(y,x)]}{\forall x \forall y[(nat(x) \wedge nat(y)) \rightarrow mnr(x,y) \vee x=y \vee mnr(y,x)]} \right\}$$

$\Delta2$ *possui uma extensão* default*:*

$$E2 = Th(W2 \cup \{\forall x \forall y[(nat(x) \wedge nat(y)) \rightarrow mnr(x,y) \vee x = y \vee mnr(y,x)]\}).$$

Vamos definir um mapeamento π *que associa* pr *a* mnr. *Definiremos uma interpretação clássica I de* $Th(W1)$ *em* $Th(W2)$ *que tem como um de seus elementos* π, *sobre a qual fazemos as seguintes considerações:*

r Tomemos nat como o *predicado* de *relativização*.

- *I é uma interpretação de $L1$ em $Th(W2)$, pois $W2 \models \exists x(nat(x))$;*

- *I é uma interpretação clássica de $Th(W1)$ em $Th(W2)$, pois $\forall x[nat(x) \rightarrow \neg mnr(x,x)]$ e $\forall x \forall y \forall z[(nat(x) \wedge nat(y) \wedge nat(z)) \rightarrow (mnr(x,y) \wedge mnr(y,z) \rightarrow mnr(x,z))]$ são consequências de $W2$;*

Note também que:

- $I(D1) = \left\{ \frac{:\forall x \forall y[(nat(x) \wedge nat(y)) \rightarrow mnr(x,y) \vee x=y \vee mnr(y,x)]}{\forall x \forall y[(nat(x) \wedge nat(y)) \rightarrow mnr(x,y) \vee x=y \vee mnr(y,x)]} \right\} = D2;$

- *Temos $A(\delta 1, W1)$, pois*

$$W1 + CONSEQUENTE(\delta 1) \not\models \neg[\forall i \forall j(pr(i,j) \vee i = j \vee pr(j,i))],$$

e $A(\delta 2, W2)$, pois

$$W2 + CONSEQUENTE(\delta 2) \not\models \neg\{\forall x \forall y[(nat(x) \wedge nat(y)) \rightarrow mnr(x,y) \vee x = y \vee mnr(y,x)]\}.$$

- *Como δ_2 é normal, não ocorre que*

$$\neg\{\forall x \forall y[(nat(x) \wedge nat(y)) \rightarrow mnr(x,y) \vee x = y \vee mnr(y,x)]\}$$

pertence a $Th(W2 \cup CONSEQUENTE(\delta_2))$.

Levando em conta as considerações acima, a proposição V.12 garante que I é uma interpretação default *de $\Delta 1$ em $\Delta 2$.*

V.4 Teorema da modularização

Agora, provaremos que, para extensões conservativas restritas da forma da seção V.2 e para interpretações default da forma da seção V.3, o teorema da modularização vale.

Sejam $\Delta = (D, W)$ sobre linguagem L e $\Delta 2 = (D, W2)$ sobre linguagem L2 teorias com *defaults* fechados sem pré-requisitos, onde $W2 = W + A$ e $Th(W) \leq Th(W2)$. Pela proposição V.9, $\Delta \leq_D \Delta 2$.

Seja $\Delta 1 = (D1, W1)$ sobre linguagem L1 uma teoria com *defaults* fechados sem pré-requisitos. I é uma tradução de Δ em $\Delta 1$, i.e, $I : Th(W) \rightarrow Th(W1)$; $I(D) = D1$; para cada $D' \subseteq D$ se $A(D', W)$, então $A(I(D'), W1)$; para cada $D'' \subseteq D1$, se $A(D'', W1)$, então $A(G, W)$, onde $I(G) = D''$; e para todo $D' \subseteq D2$, se para algum $\frac{:I(\beta_1),\ldots,I(\beta_m)}{I(w)} \in D2$,

$$I(\neg\beta_i) \in Th(W1 + \text{CONSEQUENTES}(D')), \text{ para algum } 0 \leq i \leq m,$$

então $\neg\beta_i \in Th(W + \text{CONSEQUENTES}(G))$, tal que $I(G) = D'$. Como vimos na proposição V.12, $I : \Delta \rightarrow_D \Delta 1$.

Teorema V.1 (Teorema da Modularização (versão restrita)) *Sejam Δ, $\Delta 1$ e $\Delta 2$ como descritos acima. Então, existe um $\Delta' = (D', W')$ teoria com* defaults *fechados sem pré-requisitos sobre linguagem $L' = L1$ + {símbolos equivalentes aos introduzidos em L para obtermos L2}, uma interpretação* default *$I' : \Delta 2 \rightharpoonup_D \Delta'$ e uma extensão conservativa restrita $\Delta 1 \leq_D \Delta'$.*

$$\begin{tikzcd} \Delta 2 \arrow[r, "I'", "D"'] & \Delta' \\ \Delta \arrow[u, "\vee/_D"] \arrow[r, "I", "D"'] & \Delta 1 \arrow[u, "\vee/_D"'] \end{tikzcd}$$

Prova

Provaremos que $\Delta' = (D', W')$, onde $D' = D1$ e $W' = Th(W1 + I(A))$.

1. $\Delta 1 \leq_D \Delta'$

   Sabemos que $I : Th(W) \to Th(W1)$ é uma interpretação clássica e $Th(W) \leq Th(W + A)$. Pelo teorema da modularização para a lógica clássica, temos $Th(W1) \leq Th(W1 + I(A))$. Logo, pela proposição V.9, temos que:

   $$\Delta 1 \leq_D \Delta';$$

2. $I' : \Delta 2 \rightharpoonup_D \Delta'$

   (a) Sabemos que $I : Th(W) \to Th(W1)$ é uma interpretação clássica e que $Th(W) \leq Th(W + A)$. Construiremos I' estendendo I de forma a mapear os novos símbolos introduzidos por A nos novos símbolos equivalentes introduzidos por $I(A)$. O teorema da modularização para a lógica clássica garante que $I' : Th(W + A) \to Th(W1 + I(A))$ é uma interpretação clássica;

   (b) Vamos supor que para algum $D' \subseteq D$, temos $A(D', W)$, mas $\neg A(D', W + A)$. Logo, para algum $\frac{:\beta_1,\ldots,\beta_m}{\gamma} \in D$, para algum $\beta \in \beta_1, \ldots, \beta_m$, temos que:

   $$W + A + \text{CONSEQUENTES}(D') \models \neg\beta.$$

   Como $Th(W) \leq Th(W + A)$, CONSEQUENTES$(D') \in L$, pela proposição V.4, temos

   $$Th(W + \text{CONSEQUENTES}(D')) \leq Th(W + A + \text{CONSEQUENTES}(D')).$$

   Como $\neg\beta \in L$:

   $$W + \text{CONSEQUENTES}(D') \models \neg\beta,$$

   implicando $\neg A(D', W)$, o que é uma contradição. Logo, se $A(D, 'W)$, então $A(D', W + A)$.

Pela mesma argumentação, se $A(D'', W1)$, então $A(D'', W1+I(A))$, para cada $D'' \subseteq D1$.

O contrário é evidente: se $A(D', W + A)$, então $A(D', W)$, para cada $D' \subseteq D$; e se $A(D'', W1+I(A))$, então $A(D'', W1)$, para cada $D'' \subseteq D1$. Portanto, para cada $D' \subseteq D$ se $A(D', W+A)$ então $A(I(D'), W1+I(A))$, e para cada $D'' \subseteq D1$, se $A(D'', W1 + I(A))$, então $A(G, W + A)$, onde $I(G) = D''$;

(c) Vamos supor, agora, que para algum subconjunto $D' \subseteq D2$, existe um $\frac{:I(\beta_1),\ldots,I(\beta_m)}{I(\omega)} \in D2$, tal que

$W1+I(A)+\text{CONSEQUENTES}(D') \models I(\neg(\beta_i))$, para algum $0 \leq i \leq m$.

Como $Th(W1) \leq Th(W1 + I(A))$ e $\text{CONSEQUENTES}(D') \in L1$, pela proposição V.4,

$$Th(W1 + \text{CONSEQUENTES}(D')) \leq$$
$$Th(W1 + I(A) + \text{CONSEQUENTES}(D')).$$

Logo, $W1+\text{CONSEQUENTES}(D') \models I(\neg\beta_i)$, $0 \leq i \leq m$. Por definição de I,

$$W + \text{CONSEQUENTES}(G) \models \neg\beta_i,\ I(G) = D'.$$

Por monotonicidade

$$W + A + \text{CONSEQUENTES}(G) \models \neg\beta_i;$$

(d) Par (a], (b) e(c} e como $I(D) = D1$, pela proposição V.12,

$$I' : \Delta 2 \rightarrow_D \Delta'.$$

C.Q.D.

Concluindo o capítulo, vejamos um exemplo da aplicação do teorema da modularização a um passo canônico.

Exemplo V.1 *Seja a teoria com* defaults $\Delta = (D, W)$ *sobre linguagem* L *com alfabeto* A *formado por* suc, *símbolo de função unária,* mnr, *símbolo de predicado binário, e* nought, *constante, onde:*

$$
\begin{aligned}
W = \{&\forall x(\neg mnr(x,x)) && \text{(V.1)}\\
&\forall x,y,z[(mnr(x,y) \land mnr(y,z)) \rightarrow mnr(x,z)] && \text{(V.2)}\\
&\forall x,y[mnr(x,y) \lor x = y \lor mnr(y,x)] && \text{(V.3)}\\
&\forall x,y[suc(x) = suc(y) \rightarrow x = y] && \text{(V.4)}\\
&\neg\exists x[suc(x) = nought]\} && \text{(V.5)}
\end{aligned}
$$

$$D = \left\{ d1 = \frac{:\forall x[mnr(x, suc(x))]}{\forall x[mnr(x, suc(x))]}, d2 = \frac{:\forall x[mnr(suc(x), x)]}{\forall x[mnr(suc(x), x)]} \right\}.$$

Δ *possui duas extensões* default*:*

- $Th(W \cup \{\forall x[mnr(x, suc(x))]\})$. *Um modelo para essa teoria são os naturais com sucessor, sendo* nought *o primeiro elemento (zero);*

- $Th(W \cup \{\forall x[mnr(suc(x), x)]\})$. *Um modelo para essa teoria são os inteiros negativos com predecessor, sendo* nought *o último elemento (zero).*

Seja $\Delta 1 = (D1, W1)$ *uma teoria com* defaults *sobre linguagem L1 com alfabeto* $A1$, *formado pelo símbolo de predicado binário* pr, *pelo símbolo de predicado unário* num, *e pelo símbolo de função unária* prox, *onde:*

$$
\begin{aligned}
W1 \;=\; & \{\forall x(\neg pr(x, x)) && (V.6)\\
& \forall x, y, z[(pr(x, y) \wedge pr(y, z)) \rightarrow pr(x, z)] && (V.7)\\
& \forall x, y[pr(x, y) \vee x = y \vee pr(y, x)] && (V.8)\\
& \forall x, y(pr(x, y) \rightarrow \exists z[pr(x, z) \wedge \forall s(pr(x, s) \rightarrow pr(z, s) \vee z = s)]) && (V.9)\\
& \forall x, y[prox(x) = prox(y) \rightarrow x = y] && (V.10)\\
& \forall x(num(x))\} && (V.11)\\
D1 \;=\; & \left\{ d3 = \frac{: \forall x[num(x) \rightarrow pr(x, prox(x))]}{\forall x[num(x) \rightarrow pr(x, prox(x))]}, \right.\\
& \left. d4 = \frac{: \forall x[num(x) \rightarrow pr(prox(x), x)]}{\forall x[num(x) \rightarrow pr(prox(x), x)]} \right\}.
\end{aligned}
$$

$\Delta 1$ *possui duas extensões* default*:*

- $E1' = Th(W1 \cup \{\forall x[num(x) \rightarrow pr(x, prox(x))]\})$. *Um modelo para essa teoria é uma ordem total discreta com sucessor;*

- $E1'' = Th(W1 \cup \{\forall x[num(x) \rightarrow pr(prox(x), a]])$. *Um modelo para essa teoria é uma ordem total discreta* com *predecessor.*

Vamos implementar Δ *em* $\Delta 1$. *Para isso, precisamos estender* $\Delta 1$ *de forma que ele possua uma constante equivalente a* nought.

Para isso, fazemos $A1' = A3 + \mathrm{nil}$, *onde* nil *é uma constante, obtendo a linguagem L1', e* $W1' = W1 + A$, *onde*

$$A : \forall x\{x = nil \leftrightarrow [\forall z(prox(z) \neq a]]. \qquad (V.12)$$

Como introduzimos uma constante, a condição de existência e de unicidade são verdadeiras.

Logo, $Th(W1')$ *é uma extensão clássica por definição de* $Th(W1)$ *e, como vimos no capítulo II,* $Th(W1) \leq Th(W1')$. *Então, pela proposição V.9,* $\Delta 1 \leq_D \Delta 1' = (D1, W1')$. $\Delta 1'$ *possui duas extensões* default, *uma que é uma extensão conservativa clássica de* $E1'$ *e tem como um de seus modelos uma ordem total* *discreta com sucessor e primeiro elemento, e outra que é uma extensão conservativa clássica de $E1''$ e tem como um de seus modelos uma ordem total discreta com predecessor e último elemento.*

Seja I a interpretação clássica de L em $L1'$, $I : L \longrightarrow L1'$, onde:

1. num *é o predicado de relativização;*

2. $\pi(suc) = prox$, $\pi(mnr) = pr$ e $\pi(nought) = nil$.

Como, por V.11, concluímos:

$$\begin{aligned} W1' &\models \exists x(num(x)) \text{ e} \\ W1' &\models num(x) \rightarrow num(prox(x)), \end{aligned}$$

então $I : L \rightarrow Th(W1')$.

Se provarmos que a tradução das fórmulas de W são consequências de $W1'$, então saberemos, pelo teorema da interpretação, que I é ama *interpretação clássica de $Th(W)$ em $Th(W1')$ (I : ThfW) $\rightarrow Th(W1')$).*

Vamos fazer isso para a primeira fórmula de W:

$$(\forall x[\neg mnr(x,x)])^I = \forall x[num(x) \rightarrow \neg pr(x,x)] \equiv \forall x[\neg num(x) \vee \neg pr(x,x)].$$

Sabemos que $W1' \models \forall x[\neg pr(x,x)]$. Logo, temos

$$W1' \models \forall x[\neg num(x) \vee \neg pr(x,x)].$$

Do mesmo modo, a tradução das próximas três fórmulas de W são consequências de. $W1'$. Então, precisamos provar que:

$$W1' \models (\neg\exists x(suc(x) = nought))^I.$$

Note que:

$$\begin{aligned} (\neg\exists x(suc(x) = nought))^I &\equiv (\forall x(suc(x) \neq nought))^I \\ &= \forall x(num(x) \rightarrow prox(x) \neq nil). \end{aligned}$$

Por V.12, temos que:

$$W1' \models \forall x\{x = nil \leftrightarrow [\forall z(prox(z) \neq x)]\}.$$

Daí concluímos que:

$$W1' \models \forall z(prox(z) \neq nil).$$

Mudando o nome da variável e usando equivalências lógicas, temos:

$$W1' \models \forall x(num(x) \rightarrow prox(x) \neq nil).$$

Então, $I : Th(W) \rightarrow Th(W1')$.

Vamos agora analisar $I(D)$. *Para isso vamos traduzir suas partes componentes:*

- $(\forall x[mnr(x, suc(x))])^I = \forall x[num(x) \rightarrow pr(x, prox(x))]$;
- $(\forall x[mnr(suc(x), x)])^I = \forall x[num(x) \rightarrow pr(prox(x), x)]$.

Logo $I(D) = D1$.

É fácil perceber que ou $A(d1, W)$, *ou* $A(d2, W)$, *e que ou* $A(d3.W1')$, *ou* $A(d4, W1')$, *pois não vale ao mesmo tempo* $mnr(x, suc(x))$ *e* $mnr(suc(x), x)$, *nem* $pr(x, prox(x))$ *e* $pr(prox(x), x)$. *Quando* $A(d1, W)$, *também* $A(d3, W1')$, *e vice-versa. Da mesma forma, quando* $A(d2, W)$, *também* $A(d4, W1')$, *e vice-versa. Mais ainda, como os defaults do exemplo são normais, nunca acontece*

$$\begin{array}{l} \neg(\forall x[num(x) \rightarrow pr(x, prox(x))]) \in \\ \quad Th(W1' \cup \{\forall x[num(x) \rightarrow pr(x, prox(x))]\}), \text{ ou} \\ \neg(\forall x[num(x) \rightarrow pr(prox(x), x)]) \in \\ \quad Th(W1' \cup \{\forall x[num(x) \rightarrow pr(prox(x), x)]\}). \end{array}$$

Logo, pela proposição V.12, $\Delta \rightarrow_D \Delta 1'$.

Assim, temos uma implementação de Δ *em* $\Delta 1$.

Capitulo VI

Conclusão

Este trabalho se propôs a estudar a aplicabilidade de alguns conceitos da lógica clássica na lógica de *defaults*, motivados pela utilização dessa lógica em especificações formais de programas.

Propomos, inicialmente, aplicações do uso da lógica de *defaults* nessa tarefa, que julgamos suficientes para motivar o estudo posterior, pois essa lógica se mostra adequada à:

- especificação de programas gerais?o que permite representar em uma única especificação uma biblioteca de programas;
- manipulação correta de errou e exceções em tipos abstratos de dados;
- especificação de procedimentos que apresentam comportamentos distintos, dependendo dos parâmetros instanciados.

Em seguida, introduzimos os conceitos de extensões, extensões conservativas, extensões expansivas e interpretações para a lógica de *defaults*, as quais denominamos, respectivamente, de extensões monotônicas restritas, extensões conservativas restritas, extensões expansivas restritas e interpretações *default*. Esses conceitos são definidos em termos de funções entre os conjuntos de extensões *default* para as teorias com *defaults*, pois precisamos saber de qual extensão *default* determinada informação veio. Na maioria das definições, exigimos que a função fosse bijetiva. No caso das extensões conservativas restritas, um exemplo nos mostrou a necessidade da bijetividade. Já no caso das interpretações *default*, nossa motivação exigia apenas que a função fosse injetiva, ma8 a bijetividade se mostrou necessária para ser possível uma prova para o teorema da modularização. Apesar dessa a itude assumir, implicitamente, que o conjunto de consequências de uma teoria com *defaults* é seu conjunto de extensões *default*, não nos preocupamos, em nossas: definições, com essa,noção da consequência, mas somente com a relação entre as extensões *default*.

Depois, estudamos como compor esses conceitos no passo canônico, nos preocupando não só com as questões técnicas, mas também com a contribuição desses conceitos na compreensão do processo de desenvolvimento de programas por refinamentos sucessivos. No fado técnico, tentamos provar o teorema da modularização nesse contexto. Através de várias aplicações do teorema da modularização para a lógica clássica, obtemos um conjunto de teorias de primeira ordem que fazem o papel da teoria com *defaults* que o teorema menciona. Entretanto, mostramos que essas teorias de primeira ordem formam uma teoria com *defaults* se forem finitamente axiomatizáveis. Corno isso não é sempre verdade, não podemos enunciar o teorema para o caso geral. Mostramos, no entanto, que, sob certas restrições, o teorema vale, sendo que essas restrições não estão tão distantes da realidade.

Finalmente, estudamos técnicas de estender e de interpretar uma teoria com *defaults*, de modo que o resultado seja, respectivamente, uma extensão conservativa restrita e uma interpretação *default*. Baseamos nossas provas em uma caracterização alternativa de extensões *default* para uma teoria com *defaults* arbitrários sem pré-requisitos. Fizemos isso, pois precisávamos entender uma extensão *default* em termos dos *defaults* aplicáveis a um conjunto de fórmulas, o que não era possível com as caracterizações existentes. Mostramos, também, que, se as teorias com *defaults* são estendidas e interpretadas segundo essas técnicas, o teorema da modularização é provado.

Percebemos, então, que a teoria desenvolvida satisfaz a motivação inicial, já que nossos exemplos exigiam somente *defaults* sem pré-requisitos. *Além* disso, no capítulo V, mostramos que podemos resumir várias teorias de primeira ordem finitamente axiomatizáveis que niio estão contidas umas nas outras em uma teoria com *defaults* semi-normais sem pré-requisitos. Dessa forma, podemos afirmar que! em relação à especificação formal de programas, não se necessita mais do que *defaults* semi-normais sem pré-requisitos com apenas uma justificativa.

Achamos que o trabalho deve ser expandido de várias maneiras:

- É preciso se aprofundar mais na aplicação da lógica de *defaults* como linguagem de especificação de programas. Com certeza, outras aplicações existem e podem exigir novas investigações teóricas. Para casos práticos, é necessária uma ampliação do estudo para uma lógica de *defaults* polisortida;
- É interessante se aprofundar o estudo da relação de nossas definições com as noções de consequência para a lógica de *defaults*;
- Precisamos de uma prova completa para o teorema da modularização para o caso geral. Parece-nos que, quando provamos, no capítulo V, o teorema da modularização restrito às técnicas de construção de extensões conservativas restritas e interpretações *default*, não haveria problema se a função associada à definição de interpretação *default* fosse apenas injetiva. Por isso, talvez essa prova para o caso geral só exija que essa função seja injetiva;
- É necessário se desenvolver novas maneiras de se obter interpretações *default* e extensões conservativas restritas, no caso em que os *defaults* das teorias com *defaults* em questão tenham uma relação menos restritiva do que nos casos tratados nessa tese.

Apêndice

Nesse apêndice, apresentamos uma prova alternativa para a proposição V.9 que não se restringe a *defaults* sem pré-requisitos.

Antes, apresentamos uma definição, dois lemas e uma proposição que usaremos na prova da proposição.

Sejam G e H conjuntos de fórmulas de primeira ordem.

Lema A.1 $Th(G+H) = Th(Th(G)+H))$

Prova:

1. $Th(Th(G)+H) \subseteq Th(G+H)$
   Sabemos que $Th(G) \subseteq Th(G+H)$ e que
   $H \subseteq Th(G+H)$
   Logo, $Th(G)+H \subseteq Th(G+H) \Rightarrow Th(Th(G)+H) \subseteq Th(Th(G+H))$.
   Portanto, $Th(Th(G)+H) \subseteq Th(G+H)$.
2. $Th(G+H) \subseteq Th(Th(G)+H)$
   Seja $F \in Th(G+H) \Rightarrow G+H \models F$. (*)
   Seja $\mathcal{A}$ uma estrutura, tal que $\mathcal{A} \models Th(G)+H$
   Logo, $\left\{ \begin{array}{lll} \mathcal{A} & \models & Th(G) \Rightarrow \mathcal{A} \models G \text{ e} \\ \mathcal{A} & \models & H \end{array} \right.$
   Portanto, $\mathcal{A} \models G+H$.
   Então, por (*): $\mathcal{A} \models F$.
   Como para toda estrutura $\mathcal{A}$, tal que $\mathcal{A} \models Th(G)+H$, ocorre que $\mathcal{A} \models F$, então $Th(G)+H \models F$.
   Logo, $F \in Th(Th(G)+H)$
   Portanto, $Th(G+H) \subseteq Th(Th(G)+H)$.
3. Por (1) e (2), $Th(G+H) = Th(Th(G)+H)$.

C.Q.D.

Sejam P e Q teorias de primeira ordem. Seja D um conjunto de *defaults*.

Definição A.1 (Disparos baseados em P)

$$D(P,Q) = \left\{ \frac{\alpha:\beta_1,\ldots,\beta_m}{w} \in D / \alpha \in P \text{ e } \neg\beta_1,\ldots,\neg\beta_m \notin Q \right\}.$$

Lema A.2 *Sejam W um conjunto de fórmulas* fechadas de *primeira ordem, E uma teoria de primeira ordena tal que $E = \bigcup_{i=0}^{\infty} E_i$, onde*

$$\begin{aligned} E_0 &= W \\ E_{i+1} &= Th(E_i) \cup CONSEQUENTES(D(E_i, E)) \end{aligned}$$

Seja F_i da seguinte forma:

$$\begin{aligned} F_0 &= Th(W) \\ F_{i+1} &= Th(F_i) \cup CONSEQUENTES(D(Th(F_i), E)) \end{aligned}$$

Afirmamos que $\bigcup_{i=0}^{\infty} E_i = \bigcup_{i=0}^{\infty} F_i$.

Prova:

1. $\bigcup_{i=0}^{\infty} E_i \subseteq \bigcup_{i=0}^{\infty} F_i$

   Basta provarmos que $E_i \subseteq F_i, \forall i \geq 0$.

   Por indução:

   i=0: $E_0 = W \subseteq Th(W) = F_0$. Logo, $E_0 \subseteq F_0$;

   h.i.: $E_i \subseteq F_i$. Portanto, $Th(E_i) \subseteq Th(F_i)$;

   p/ i+1: Tome um $F \in E_{i+1}$. Logo, ou:

   (a) $F \in Th(E_i)$. Por h.i., $F \in Th(F_i)$; ou:

   (b) $F \in$ CONSEQUENTES$(D(E_i, E))$.
   Seja $d = \frac{\alpha:\beta_1,\ldots,\beta_m}{w} \in D(E_i, E)$. Logo, $\alpha \in E_i$. Portanto, $\alpha \in Th(E_i)$.
   Por h.i., $\alpha \in Th(F_i)$.
   Dessa forma, $d \in D(Th(F_i), E)$. Então,

   $$F \in \text{CONSEQUENTES}(D(Th(F_i), E)).$$

   (c) Por (a) e (b), $F \in F_{i+1}$. Então, $E_{i+1} \subseteq F_{i+1}$.

   Portanto, $\bigcup_{i=0}^{\infty} E_i \subseteq \bigcup_{i=0}^{\infty} F_i$.

2. $\bigcup_{i=0}^{\infty} F_i \subseteq \bigcup_{i=0}^{\infty} E_i$

   Basta provarmos que $F_i \subseteq \bigcup_{i=0}^{\infty} E_i, \forall i \geq 0$.

   Por indução:

   i=0: $F_0 = Th(W) \subseteq E1$. Logo, $F_0 \subseteq \bigcup_{i=0}^{\infty} E_i$;

   h.i.: $F_i \subseteq \bigcup_{i=0}^{\infty} E_i$;

   p/ i+1: Seja $F \in F_{i+1}$. Logo, ou:

   (a) $F \in Th(F_i)$. Por h.i., $F \in Th(\bigcup_{i=0}^{\infty} E_i)$. Como $\bigcup_{i=0}^{\infty} E_i$ é teoria de primeira ordem, $\bigcup_{i=0}^{\infty} E_i = Th(\bigcup_{i=0}^{\infty} E_i)$. Logo, $F \in \bigcup_{i=0}^{\infty} E_i$; ou:

   (b) $F \in$ CONSEQUENTES$(D(Th(F_i), E))$.
   Seja $d = \frac{\alpha:\beta_1,\ldots,\beta_m}{\omega} \in D(Th(F_i), E)$. Logo, $\alpha \in Th(F_i)$ e como discutido em (a), $\alpha \in \bigcup_{i=0}^{\infty} E_i$. Portanto, p/ algum $0 \leq j \leq \infty$, $\alpha \in E_j$.
   Portanto, $d \in D(E_j, E)$. Logo,

   $$F \in \text{CONSEQUENTES}(D(E_j, E)).$$

   Dessa forma, $F \in E_{j+1}$. Então, $F \in \bigcup_{i=0}^{\infty} E_i$.

   (c) Por (a) e (b), vemos que $F_{i+1} \subseteq \bigcup_{i=0}^{\infty} E_i$.

   Portanto, $\bigcup_{i=0}^{\infty} F_i \subseteq \bigcup_{i=0}^{\infty} E_i$.

3. Por 1 e 2, $\bigcup_{i=0}^{\infty} E_i = \bigcup_{i=0}^{\infty} F_i$.

C.Q.D.

Proposição A.1 *Observe o esquema abaixo:*

$$\begin{array}{ccc} W & \xrightarrow{D(W,E)} & E \\ /\wedge & & /\wedge \\ W' & \xrightarrow{D(W',E')} & E' \end{array}$$

W, E são teorias de primeira ordem consistentes sobre uma linguagem L, e W' e E' são teorias de primeira ordem consistentes sobre uma linguagem L', com $L \subseteq L'$. D é um conjunto de defaults*, cujos pré-requisitos, justificativas e consequentes estão em L.*

$$\textit{Se } W \leq W' \textit{ e } E \leq E', \textit{ então } D(W,E) = D(W',E').$$

Prova:

1. $D(W', E') \subseteq D(W, E)$

   Seja $d = \frac{\alpha:\beta_1,\ldots,\beta_m}{\omega} \in D(W', E')$. Por definição:

(a) $\alpha \in W'$. Como $\alpha \in L$ e $W \leq W'$, então, $\alpha \in W$;

(b) $\neg\beta_1, \ldots, \neg\beta_m \notin E'$. Como $E \subseteq E'$ e E é consistente, então $\neg\beta_1, \ldots, \neg\beta_m \notin E$.

Por (a) e (b), temos que $d \in D(W, E)$. Portanto, $D(W', E') \subseteq D(W, E)$.

2. $D(W, E) \subseteq D(W', E')$

Seja $d = \frac{\alpha:\beta_1,\ldots,\beta_m}{\omega} \in D(W, E)$. Por definição:

(a) $\alpha \in W$. Como $W \subseteq W'$, então $\alpha \in W'$;

(b) $\neg\beta_1, \ldots, \neg\beta_m \notin E$. Se $\neg\beta_1, \ldots, \neg\beta_m \in E'$, como $E \leq E'$ e $\neg\beta_1, ..., \neg\beta_m \in L1$, então $\neg\beta_1, ..., \neg\beta_m \in E$, o que é um absurdo! Portanto, $\neg\beta_1, ..., \neg\beta_m \notin E'$.

Por (a) e por (b), $d \in D(W', E')$. Portanto, $D(W, E) \subseteq D(W', E')$.

3. Por (1) e por (2) $D(W, E) = D(W', E')$

C.Q.D

Vamos provar que $\Delta 1 = (D, W1) \leq_D \Delta 2 = (D, W1 + A)$, onde $\Delta 1$ e $\Delta 2$ são teorias com *defaults* fechados e $Th(W1) \leq Th(W1 + A)$, onde A é uma fórmula fechada. Se $W1$ for inconsistente, REITER [7] mostrou que $\Delta 1$ e $\Delta 2$ possuem, cada uma, uma única extensão *default* inconsistente, que chamamos de $E1$ e $E2$, respectivamente. Obviamente $E1 \leq E2$. Logo, $\Delta 1 \leq_D \Delta 2$. Assim, consideraremos o caso em que $W1$ e $W1 + A$ são consistentes.

Consideremos, agora, a seguinte situação. Sejam $\Delta 1 = (D, W1)$ e $\Delta 2 = (D, W2)$ teorias com *defaults* fechados sobre linguagens L1 e L2, respectivamente, sendo que $L1 \subseteq L2$. $W1$ e $W2$ são apresentações de teorias consistentes, $W2 = W1 + A$, onde A é um novo axioma, e $Th(W1) \leq Th(W2)$. Seja E1 uma extensão *default* para $\Delta 1$.

Pelo teorema III.2, $E1 = Th(W1 \cup \text{CONSEQUENTES}(GD(E1, \Delta 1)))$.

Seja $E2 = Th(W1 + A + \text{CONSEQUENTES}(GD(E1, \Delta 1)))$. Pela proposição V.4, temos que $E1 \leq E2$.

Pelo lema A.2 e usando nossa notação, $E1 = \bigcup_{i=0}^{\infty} E_i^1$, onde:

$$\begin{aligned} E_0^1 &= Th(W1) \\ E_{i+1}^1 &= Th(E_i^1) \cup \text{CONSEQUENTES}(D(Th(E_i^1), E1)) \end{aligned}$$

Considere E_i^2 do seguinte modo:

$$\begin{aligned} E_0^2 &= Th(W2) = Th(W1 + A) \\ E_{i+1}^2 &= Th(E_i^2) \cup \text{CONSEQUENTES}(D(Th(E_i^2), E2)) \end{aligned}$$

Vamos provar que $E2 = \bigcup_{i=0}^{\infty} E_i^2$, mostrando que $E2$ é uma extensão *default* para $\Delta 2$.

Lema A.3 $Th(E_i^1) \leq Th(E_i^2)$ e $D(Th(E_i^1), E1) = D(Th(E_i^2), E2), \forall i \geq 0$.

Prova:

Por indução:

i=0: $Th(E_0^2) = Th(Th(W1 + A)) = Th(W1 + A) = Th(W2) \geq Th(W1) = Th(Th(W1)) = Th(E_0^1)$. Portanto, $Th(E_0^1) \leq Th(E_0^2)$.

Acrescentando a isso o fato de $E1 \leq E2$, pela proposição A.1:

$$D(Th(E_0^1), E1) = D(Th(E_0^2), E2);$$

h.i.: $Th(E_i^1) \leq Th(E_i^2)$ e $D(Th(E_i^1), E1) = D(Th(E_i^2), E2)$;

p/ i+1: Seja $F \in Th(E_{i+1}^2)$, onde $F \in L1$.

Sabemos que

$$Th(E_{i+1}^2) = Th[Th(E_i^2) + \text{CONSEQUENTES}(D(Th(E_i^2), E2))].$$

Logo,

$$Th(E_i^2) + \text{CONSEQUENTES}(D(Th(E_i^2), E2)) \models F.$$

Considere $\{w1, \ldots, wn\} \in \text{CONSEQUENTES}(D(Th(E_i^2), E2))$. Por compacidade:

$Th(E_i^2) + \{w1, ..., wn\} \models F$ (s.p.gen.)

$Th(E_i^2) \models w1 \wedge \ldots \wedge wn \rightarrow F$.

Como $w1 \wedge \ldots \wedge wn \rightarrow F \in L1$ e, por h.i., $Th(E_i^1) \leq Th(E_i^2)$,

$Th(E_i^1) \models w1 \wedge \ldots \wedge wn \rightarrow F$.

Como, por h.i., $D(Th(E_i^1), E1) = D(Th(E_i^2), E2)$, então

$\{w1, \ldots, wn\} \in \text{CONSEQUENTES}(D(Th(E_i^1), E1))$.

Logo, $Th(E_i^1) + \text{CONSEQUENTES}(D(Th(E_i^1), E1)) \models F$.

$\Rightarrow F \in Th[Th(E_i^1) + \text{CONSEQUENTES}(D(Th(E_i^1), E1))]$.

$\Rightarrow F \in Th(E_{i+1}^1)$

$\Rightarrow Th(E_{i+1}^1) \leq Th(E_{i+1}^2)$.

Além disso, como $E1 \leq E2$, pela proposição A.1: $D(Th(E_{i+1}^1), E1) = D(Th(E_{i+1}^2), E2)$.

C.Q.D.

Lema A.4 $Th(E_i^2) = Th(E_i^1 + A)$

Prova:

Por indução:

i=0: $E_0^1 = Th(W1)$. Logo,

$$Th(E_0^1 + A) = Th(Th(W1) + A) = Th(W1 + A) = E_0^2 = Th(E_0^2);$$

h.i.: $Th(E_i^2) = Th(E_i^1 + A)$;

p/ i + 1:

$$Th(E_{i+1}^2) = Th[Th(E_i^2) + \text{CONSEQUENTES}(D(Th(E_i^2), E2))]$$

Por h.i. e pelo lema A.3:

$$\begin{aligned}
Th(E_{i+1}^2) &= Th[Th(E_i^1 + A) + \text{CONSEQUENTES}(D(Th(E_i^1), E1))] \\
&= Th[E_i^1 + A + \text{CONSEQUENTES}(D(Th(E_i^1), E1))] \\
&= Th[E_i^1 + \text{CONSEQUENTES}(D(Th(E_i^1), E1)) + A] \\
&= Th[Th(E_i^1 + \text{CONSEQUENTES}(D(Th(E_i^1), E1))) + A] \\
&= Th[Th(Th(E_i^1) + \text{CONSEQUENTES}(D(Th(E_i^1), E1))) + A] \\
&= Th[Th(E_{i+1}^1) + A] \\
&= Th[E_{i+1}^1 + A].
\end{aligned}$$

C.Q.D

Proposição A.2 *E2 é extensão* default *para* $\Delta 2$.

Prova:

1. $\bigcup_{i=0}^{\infty} E_i^2 = \bigcup_{i=0}^{\infty} Th(E_i^2)$, pois
   (a) $\bigcup_{i=0}^{\infty} E_i^2 \subseteq \bigcup_{i=0}^{\infty} Th(E_i^2)$ (monotonicidade);
   (b) $\bigcup_{i=0}^{\infty} Th(E_i^2) \subseteq \bigcup_{i=0}^{\infty} E_i^2$, pois $Th(E_i^2) \subseteq E_{i+1}^2$, p/ $\forall i \geq 0$.
2. $E2 = \bigcup_{i=0}^{\infty} E_i^2$
   Por definição, $E2 = Th(E1 + A) = Th(\bigcup_{i=0}^{\infty} E_i^1 + A)$.
   Por sua vez, $\bigcup_{i=0}^{\infty} E_i^2 \overset{\text{por1}}{=} \bigcup_{i=0}^{\infty} Th(E_i^2) \overset{\text{Lema A.4}}{=} \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$.
   Portanto, devemos provar: $Th(\bigcup_{i=0}^{\infty} E_i^1 + A) = \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$.

(a) $\bigcup_{i=0}^{\infty} Th(E_i^1 + A)$ é teoria de primeira ordem.

Basta provarmos que $\bigcup_{i=0}^{\infty} Th(E_i^1 + A) = Th[\bigcup_{i=0}^{\infty} Th(E_i^1 + A)]$.

i. $\bigcup_{i=0}^{\infty} Th(E_i^1 + A) \subseteq Th[\bigcup_{i=0}^{\infty} Th(E_i^1 + A)]$ (monotonicidade)

ii. $Th[\bigcup_{i=0}^{\infty} Th(E_i^1 + A)] \subseteq \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$

Sabemos que $Th(E_i^1) \subseteq Th(E_{i+1}^1)$, para $\forall i \geq 0$, pois $Th(E_{i+1}^1) = Th(Th(E_i^1) + \text{CONSEQUENTES}(D(Th(E_i^1), E1)))$.

Portanto, $\bigcup_{i=0}^{j} Th(E_i^1) \subseteq Th(E_{j+1}^1)$, para cada j finito (*)

Seja $F \in Th[\bigcup_{i=0}^{\infty} Th(E_i^1 + A)]$

Logo, $\bigcup_{i=0}^{\infty} Th(E_i^1 + A) \models F$.

Por compacidade, existe um j finito tal que $\bigcup_{i=0}^{j} Th(E_i^1 + A) \models F$.

Por (*) e por monotonicidade:

$$Th(E_{j+1}^1 + A) \models F \Rightarrow F \in \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$$

Portanto, $Th[\bigcup_{i=0}^{\infty} Th(E_i^1 + A)] \subseteq \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$

Por (i) e (ii), vemos que $\bigcup_{i=0}^{\infty} Th(E_i^1 + A) = Th[\bigcup_{i=0}^{\infty} Th(E_i^1 + A)]$.

Logo, $\bigcup_{i=0}^{\infty} Th(E_i^1 + A)$ é teoria!

(b) $Th(\bigcup_{i=0}^{\infty}(E_i^1) + A) \subseteq \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$

Seja $F \in Th(\bigcup_{i=0}^{\infty}(E_i^1) + A) \Rightarrow \bigcup_{i=0}^{\infty}(E_i^1) + A \models F$ (**)

Seja $\mathcal{A}$ uma estrutura, tal que $\mathcal{A} \models \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$.

Logo, $\mathcal{A} \models Th(E_i^1 + A), \forall i \geq 0 \Rightarrow \mathcal{A} \models E_i^1 + A, \forall i \geq 0$

$$\Rightarrow \begin{cases} \mathcal{A} & \models & E_i^1, \forall i \geq 0 \Rightarrow \mathcal{A} \models \bigcup_{i=0}^{\infty} E_i^1 \\ \mathcal{A} & \models & A \end{cases}$$

Portanto, $\mathcal{A} \models \bigcup_{i=0}^{\infty} E_i^1 + A$,

Então, por (**): $\mathcal{A} \models F$.

Como para toda estrutura $\mathcal{A}$, tal que $\mathcal{A} \models \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$, ocorre que $\mathcal{A} \models F$, então $\bigcup_{i=0}^{\infty} Th(E_i^1 + A) \models F$.

Logo. $F \in Th[\bigcup_{i=0}^{\infty} Th(E_i^1 + A)] \overset{\text{por a}}{=} \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$

Portanto, $Th(\bigcup_{i=0}^{\infty}(E_i^1) + A) \subseteq \bigcup_{i=0}^{\infty} Th(E_i^1 + A)$.

(C) $\bigcup_{i=0}^{\infty} Th(E_i^1 + A) \subseteq Th(\bigcup_{i=0}^{\infty}(E_i^1) + A)$

Seja $F \in \bigcup_{i=0}^{\infty} Th(E_i^1 + A) \Rightarrow F \in Th(E_j^1 + A)$, p/ algum $j \geq 0$

$\Rightarrow E_j^1 + A \models F$, p/ algum $j \geq 0$ (***)

Seja $\mathcal{A}$ uma estrutura, tal que $\mathcal{A} \models \bigcup_{i=0}^{\infty} E_i^1 + A$

$$\text{Logo, } \begin{cases} \mathcal{A} & \models & \bigcup_{i=0}^{\infty} E_i^1 \Rightarrow \mathcal{A} \models E_i^1, \forall i \geq 0 \Rightarrow \mathcal{A} \models E_j \\ \mathcal{A} & \models & A \end{cases}$$

Portanto. $\mathcal{A} \models E_j^1 + A, \forall i \geq 0$

Então, por (***): $\mathcal{A} \models F$.

Como para, toda estrutura $\mathcal{A}$, tal que $\mathcal{A} \models \bigcup_{i=0}^{\infty} E_i^1 + A$, ocorre que $\mathcal{A} \models F$, então $\bigcup_{i=0}^{\infty} E_i^1 + A \models F$.

Logo, $F \in Th[\bigcup_{i=0}^{\infty} E_i^1 + A]$.

Portanto, $\bigcup_{i=0}^{\infty} Th(E_i^1 + A) \subseteq Th(\bigcup_{i=0}^{\infty}(E_i^1) + A)$.

(d) Por (b) e (C), $\bigcup_{i=0}^{\infty} Th(E_i^1 + A) = Th(\bigcup_{i=0}^{\infty}(E_i^1) + A)$

Portanto, $E2 = \bigcup_{i=0}^{\infty} E_i^2$. Isto é, E2 é extensão *default* para $\Delta 2$.

C.Q.D.

A proposição A.2 afirma que, ao acrescentarmos o axioma A à uma extensão *default* para $\Delta 1$, as consequências dessa união formam uma extensão *default* para $\Delta 2$. Isso define uma função $f : \mathcal{E}1 \mapsto \mathcal{E}2$, onde para cada par $(E1, E2)$ de f, E2 é uma extensão conservativa clássica de E1. Com a mesma argumentação usada na proposição V.6, provamos que f é injetiva.

Consideremos, agora, a seguinte situação: sejam $\Delta 1$ e $\Delta 2$ os mesmos da discussão anterior. Seja E2 uma extensão *default* para $\Delta 2$. Pelo teorema III.2, $E2 = Th(W1 + A + \text{CONSEQUENTES}(GD(E2, \Delta 2)))$. Pelo lema A.2 e usando nossa notação, $E2 = \bigcup_{i=0}^{\infty} E_i^2$, onde

$$
\begin{aligned}
E_0^2 &= Th(W1 + A) \\
E_{i+1}^2 &= Th(E_i^2) + \text{CONSEQUENTES}(D(Th(E_i^2), E2))
\end{aligned}
$$

Seja $E1 = E2|_{L1}$

Considere E_i^1 do seguinte modo:

$$
\begin{aligned}
E_0^1 &= Th(W1) \\
E_{i+1}^1 &= Th(E_i^1) \cup \text{CONSEQUENTES}(D(Th(E_i^1), E1))
\end{aligned}
$$

Lema A.5 $E1 \leq E2$

Prova:

Como $E1 = E2|_{L1}$, obviamente $E1 \leq$ E?, .

C.Q.D

Lema A.6 $Th(E_i^1) = Th(E_i^2)|_{L1}$

Prova:

Por indução:

i=0: $Th(E_0^2)|_{L1} = Th(Th(W1 + A))|_{L1} = Th(W1 + A)|_{L1}$

Como $Th(W1) \leq Th(W1 + A)$:

$$Th(W1 + A)|_{L1} = Th(W1) = Th(Th(W1)) = Th(E_0^1);$$

h.i.: $Th(E_i^2)|_{L1} = Th(E_i^1)$;

p/ i+1:

Por h.i., $Th(E_i^2)|_{L1} = Th(E_i^1) \Rightarrow Th(E_i^1) \leq Th(E_i^2)$.

Além disso, pelo lema A.5, $E1 \leq E2$. Logo, pela proposição A.1: $D(Th(E_i^1), E1) = D(Th(E_i^2), E2)$.

$$\begin{aligned}
Th(E_{i+1}^2)|_{L1} &= Th[Th(E_i^2) + \text{CONSEQUENTES}(D(Th(E_i^2), E2))]|_{L1} \\
&= Th[Th(E_i^2) + \text{CONSEQUENTES}(D(Th(E_i^1), E1))]|_{L1} \\
&= Th[Th(E_i^2)|_{L1} + \text{CONSEQUENTES}(D(Th(E_i^1), E1))] \\
&\overset{\text{por h.i.}}{=} Th[Th(E_i^1) + \text{CONSEQUENTES}(D(Th(E_i^1), E1))] \\
&= Th(E_{i+1}^1)
\end{aligned}$$

C.Q.D.

Corolário A.1 $Th(E_i^1) \leq Th(E_i^2)$

Prova: Óbvia.

Proposição A.3 *E1 é extensão* default *para* $\Delta 1$.

Prova:

Devemos provar que $E1 = \bigcup_{i=0}^{\infty} E_i^1$

1. $\bigcup_{i=0}^{\infty} E_i^1 \subseteq E1$

   Basta provarmos que $E_i^1 \subseteq E1$, p/ $\forall i \geq 0$.

   Por indução:

   i=0:

$$\begin{aligned}
E_0^1 &= Th(W1) \\
&\subseteq Th(W1 + A + \text{CONSEQUENTES}(GD(E2, \Delta 2)))|_{L1} \\
&= E1 \Rightarrow E_0^1 \subseteq E1;
\end{aligned}$$

   h.i.: $E_i^1 \subseteq E1$.

   p/ i+1: Seja $F \in E_{i+1}^1$. Logo, ou:

   (a) $F \in Th(E_i^1)$. Por h.i., $F \in E1$;

(b) $F \in$ CONSEQUENTES$(D(Th(E_i^1), E1))$.
Pelo corolário A.1, $Th(E_i^1) \leq Th(E_i^2)$ e pelo lema A.5, $E1 \leq E2$. Logo, pela proposição A.1:
$D(Th(E_i^1), E1) = D(Th(E_i^2), E2)$
$\Rightarrow F \in$ CONSEQUENTES$(D(Th(E_i^2), E2)) \Rightarrow F \in E_{i+1}^2 \Rightarrow F \in E2$.
Como, pelo lema A.5 $E1 \leq E2$ e como $F \in L1 \Rightarrow F \in E1$.

Logo, $E_i^1 \subseteq E1, \forall i \geq 0 \Rightarrow \bigcup_{i=0}^{\infty} E_i^1 \subseteq E1$.

a, $E1 \subseteq \bigcup_{i=0}^{\infty} E_i^1$

Seja F $\in E1 \Rightarrow E1 \models$ F.

Como $E1 \subseteq$ E2 (implícito no lema A.5), $E2 \models$ F.

Portanto, para algum i, $E_i^2 \models F \Rightarrow F \in Th(E_i^2)$.

Pelo corolário A.1, $Th(E_i^1) \leq Th(E_i^2)$. Como $F \in L1$, $F \in Th(E_i^1)$. Logo, $F \in E_{i+1}^1 \Rightarrow E1 \subseteq \bigcup_{i=0}^{\infty} E_i^1$.

3. Por (1) e (2), $E1 = \bigcup_{i=0}^{\infty} E_i^1$.

C.Q.D.

A proposição A.3 afirma que, ao restringirmos uma extensão *default* para $\Delta 2$ à linguagem de $\Delta 1$, obtemos uma extensão *default* para $\Delta 1$. Isso define uma função $g : \mathcal{E}2 \mapsto \mathcal{E}1$, onde para cada par $(E2, E1)$ de g, E2 é extensão consevativa clássica da E1.

Como E2 é extensão *default* para $\Delta 2$:

$$E2 = Th(W1 + A + \text{CONSEQUENTES}(GD(E2, \Delta 2))).$$

Como uma extensão *default* nunca está dentro da outra, sempre haverá algum elemento que diferenciará uma extensão de outra. Pela estrutura de E2, percebemos que esse elemento tem que estar nos consequentes dos *defaults* geradores da extensão *default*. Como esses consequentes estão na linguagem de $\Delta 1$, esses elementos diferenciadores também estarão na restrição de E2 à L1. Portanto, g é injetiva.

Proposição A.4 *Sejam $\Delta 1 = (D, W1)$ e $\Delta 2 = (D, W2)$ teorias com* defaults *sobre linguagens L1 e L2, respectivamente, onde $W2 = W1 + A$. Se $Th(W1) \leq Th(W2)$, então $\Delta 1 \leq_D \Delta 2$.*

Prova: Equivalente a prova da proposição V.9, considerando que a restrição de $Th(E1 + A)$ R linguagem de $E1$ é o próprio $E1$.

Referências Bibliográficas

[1] SHOENFIELD, J. E., *Mathematical Logic*, Addinson Wesley Publishing Company, 1967;

[2] ENDERTON, *H*. F., *A Mathematical Introduction to Logic*, Academic Press, 1973;

[3] TURSKI, W. M., MAIBAUM, T. S., *The Specification of Computer Programs*, Addison-Wesley Publishing Company, 1987;

[4] VELOSO, P. A. S., VELOSO, *S. R*. M., 'On Extensions by Function Symbols: Conservativeness and Comparison', Relatório Técnico COPPE/UFRJ, Univ. Federal do Rio de Janeiro, Programa de Engenharia de Sistemas e Computação, 1990;

[5] VELOSO, P. A. S., VELOSO, S. R. M., 'Some Remarks on Conservative Extensions: A Socratic Dialogue', *Bulletim of EATCS (European Association for Theoretical Computer Science)*, 43, 189-198, fevereiro 1991;

[6] VELOSO, P. A. S., *Estruturação e Verificação de Programas com Tipos de Dados*, Edt. Edgar Blutcher Ltda., 1987;

[7] REITER, R., 'A Logic for Default Reasoning', *Artificial Intelligence*, 13, 81-132, 1980;

[8] LUKASZEWICZ, W., 'Two Results on Default Logic', *Proceedings* 9^{th} *International Joint Conference on Artificial Intelligence*, Los Angeles, CA, 459-461, 1985;

[9] ETHERINGTON, D. W., *Reasoning with Incomplete Information*, (Research Notes in Artificial Intelligence), Pittman Publishing, 1988;

[10] REITER, R., CRISCOULO, G., 'On Interacting Defaults', *Proceedings* 7^{th} *International Joint Conference on Artificial Intelligence*, 94-100, 1981;

[11] LUKASZEWICZ, W., *Non-monotonic Reasoning: Formalization of Commonsense Reasoning*, Ellis Horwood series in artificial intelligence, 1990;

[12] ETHERINGTON, D. W., 'A Semantics for Default Logic', *Proceedings* 10^{th} *International Joint Conference an Artificial Intelligence*, Milan, Italy, 495-498, i987;

[13] GUERREIRO, R.A.T., CASANOVA, M.A., 'An Alternative Semantics for Default Logic', *Preprints of the Third International Workshop on Nonmonotonic Reasoning*, South Lake Tahoe, California, 141-157, 1990;

[14] GUERREIRO, R.A.T., SILVA, A., CASANOVA, M.A., 'Raciocínio Matemático em Lógica de Defaults', *Revista Brasileira de Computação*, vol. 5, n. 3, 27-36, 1990;

[15] LUKASZEWICZ, W., 'Considerations on Default Logic', *Proceedings AAAI Workshop on Nonmonotonic Reasoning*, New Paltz, NY, 165-193, Outubro 1984;

[16] MAIBAUM, T. S., TURSKI, W. M., 'On What Exactly Is Going On When Software Is Developed Step-by-step', *Proceedings 7th International Conferenc on Software Engineering*, 528-533, 1984;

[17] POOLE, D., 'A Logical Framework for Default Reasoning', *Artificial Intelligence*, 36, 27-48, 1988.