065

UM MÉTODO DE DEDUÇÃO NÃO-MONOTÔNICO E PARAMETRIZADO BASEADO EM ELIMINAÇÃO DE MODELOS

Alexandre P. de Carvalho[1,2], Marco A. Casanova[1] e Sheila R. M. Veloso[2]

[1]Centro Científico Rio - IBM Brasil
Caixa Postal 4624 - 20001, Rio de Janeiro, RJ

[2]Universidade Federal do Rio de Janeiro
Programa de Engenharia de Sistemas e Computação - COPPE
Caixa Postal 68511 - 21945, Rio de Janeiro, RJ

RESUMO - Este trabalho apresenta os fundamentos teóricos de uma família de sistemas não-monotônicos para programação em lógica. Estes sistemas definem diferentes implementações da Hipótese do Mundo Fechado no contexto de cláusulas genéricas. Demonstra-se ainda que alguns destes sistemas são corretos e completos.

PALAVRAS-CHAVE - Método de Dedução Não-Monotônico, Hipótese do Mundo Fechado, Cláusulas Genéricas, Eliminação de Modelos, Lógica de Defaults, Programação em Lógica.

1. INTRODUÇÃO

A Hipótese do Mundo Fechado (HMF) [LLOY87,REIT78] permite que informações negativas sejam deduzidas a partir de um conjunto de cláusulas definidas: se um literal básico positivo L não é conseqüência lógica de um conjunto de cláusulas definidas Q, então deduza $\neg$L a partir de Q. Em se tratando de cláusulas genéricas, a HMF pode ser estendida a fim de permitir a inferência tanto de literais básicos negativos, assim como de literais básicos positivos. Entretanto, esta extensão deve ser cuidadosamente definida de forma a não permitir a inferência de um literal e seu complemento simultaneamente.

Neste trabalho então propomos um método de dedução não-monotônico, baseado em eliminação de modelos, que implementa a HMF no contexto de cláusulas genéricas. Nesta abordagem, permitiremos que sejam indicados os símbolos predicativos sobre os quais deseja-se aplicar a HMF.

A nossa proposta resulta no Método da Eliminação de Modelos Fraca com Geração de Lemas por Defaults ou, abreviadamente, EMF/GLD. Este método singulariza-se por um procedimento de geração de lemas durante o processo de dedução. Um lema L é um literal básico cuja adição à base de conhecimento justifica-se por um default da forma (:L/L). Deste modo, associado a uma base de conhecimento, existe um conjunto de extensões, no sentido da Lógica de Defaults.

A geração de um lema L exige a execução de um teste de consistência de L em relação à base de conhecimento acrescida dos lemas até então gerados. Conforme constataremos no decorrer deste trabalho, não existe, para todas as situações, uma forma única e ideal de proceder a este teste de consistência. Este fato nos levou a parametrizar o teste de consistência dos lemas gerados: a forma deste teste e o momento ao longo da dedução no qual é realizado são definidos através de duas funções parciais que funcionam como parâmetros. Desta forma, o teste de consistência pode ser adequadamente definido, de acordo com a aplicação apresentada. O método EMF/GLD define uma família de sistemas não-monotônicos para programação em lógica na medida em que cada parametrização do método equivale a um sistema não-monotônico.

Estruturamos este trabalho da seguinte forma. A Seção 2 apresenta uma revisão da Lógica de Defaults e do Método da Eliminação de Modelos Fraca. A Seção 3 define o método EMF/GLD. A Seção 4 ilustra o método com a definição de duas parametrizações e analisa o funcionamento destas. A Seção 5 examina os principais resultados obtidos acerca do método proposto e, encerrando, a Seção 6 apresenta as conclusões. (Todos os detalhes da proposta apresentada neste trabalho podem ser encontados em [CARV90].)

2. A LÓGICA DE DEFAULTS E O MÉTODO DA ELIMINAÇÃO DE MODELOS FRACA

Esta seção apresenta uma revisão dos conceitos básicos da Lógica de Defaults na qual está fundamentado o componente não-monotônico do método de dedução a ser proposto. Além disto, examina o Método da Eliminação de Modelos Fraca, abreviadamente EMF, que constitui a base para a definição do novo método.

2.1. A Lógica de Defaults

A Lógica de Defaults [REIT80] representa uma das alternativas utilizadas para formalizar o raciocínio não-monotônico. Permite estender a informação contida em uma teoria de primeira ordem através de regras de inferência especiais, denominadas defaults. Intuitivamente, os defaults expressam regras prevendo exceções, retratando o conhecimento impreciso ou incompleto denotado por expressões como "Normalmente" ou "Tipicamente".

Definição 1: Default.

(a) Um *default* sobre um alfabeto de primeira ordem $\mathcal{A}$ é uma expressão da forma $(A{:}B_1,B_2,\ldots,B_n/C)$, onde $A,B_1,B_2,\ldots,B_n,C$, $n \geq 1$, são fórmulas sobre $\mathcal{A}$.

(b) Um default da forma $(A{:}B_1,B_2,\ldots,B_n/C)$ é *fechado* se e somente se as fórmulas $A,B_1,B_2,\ldots,B_n,C$ são fechadas.

Um exemplo clássico da literatura é o default Pássaro(x):Voa(x)/Voa(x) que representa a sentença: "Se x é um pássaro e se é consistente afirmar que x voa, então conclua que x voa", ou simplesmente, "Tipicamente, pássaros voam".

Definição 2: Teoria com Defaults.

(a) Uma *teoria com defaults* sobre um alfabeto de primeira ordem $\mathcal{A}$ é um par (D,S), onde D é um conjunto de defaults sobre $\mathcal{A}$ e S é um conjunto de sentenças sobre $\mathcal{A}$.

(b) Uma teoria com defaults (D,S) é *fechada* se e somente se todo default em D é fechado.

A definição a seguir trata das extensões de uma teoria com defaults fechada que, intuitivamente, são os conjuntos de sentenças de primeira ordem que podem, individualmente, ser aceitos como interpretações consistentes desta teoria.

A notação Th(F) representa o conjunto de todas as conseqüências lógicas em primeira ordem do conjunto F de sentenças sobre um alfabeto (de primeira ordem) $\mathcal{A}$ ou, mais especificamente, $Th(F) = \{w \mid w \text{ é uma sentença sobre } \mathcal{A} \text{ e } F \models w\}$.

Definição 3: Extensão de uma Teoria com Defaults Fechada.

Seja A um alfabeto de primeira ordem.
Seja $\Delta=(D,S)$ uma teoria com defaults fechada sobre A.
Seja F um conjunto de sentenças sobre A.
Seja $\Gamma(F)$ o menor conjunto com as seguintes propriedades:

(i) $S \subseteq \Gamma(F)$;

(ii) $Th(\Gamma(F)) = \Gamma(F)$;

(iii) Se $(A:B_1,B_2,\ldots,B_n/C) \in D$, $A \in \Gamma(F)$ e se, para todo i, $1 \leq i \leq n$, $\neg B_i \notin F$, então $C \in \Gamma(F)$.

Um conjunto E de sentenças sobre A é uma *extensão* de Δ se e somente se $\Gamma(E)=E$, ou seja, se e somente se E é um ponto fixo do operador Γ.

2.2. O Método da Eliminação de Modelos Fraca

O Método da Eliminação de Modelos Fraca, ou EMF, apresentado em [CASA87,LOVE68,LOVE69,LOVE78], foi adotado como base do método proposto neste trabalho devido a algumas características especiais. O método EMF é linear de entrada, não utiliza fatoração e, apesar destas características, é refutacionalmente correto e completo.

O método EMF trabalha com seqüências de literais e literais resolvidos, chamadas cadeias. Os literais resolvidos, ou R-literais, são marcados com colchetes. Por este motivo, o alfabeto de primeira ordem em questão deve conter os colchetes esquerdo e direito, "[" e "]".

Definição 4: Cadeia.

Seja A um alfabeto de primeira ordem.

(a) Um *literal resolvido*, ou um *R-literal*, sobre A é uma expressão da forma [L], onde L é um literal sobre A.

(b) Um *elemento* sobre A é um literal ou um R-literal sobre A.

(c) Uma *cadeia* sobre A é ou uma seqüência não vazia de elementos sobre A ou a *cadeia vazia*, denotada por "$\square$".

(d) Uma cadeia é *elementar* se e somente se é a cadeia vazia ou é uma cadeia sem ocorrências de R-literais.

Uma cadeia C é satisfatível se e somente se o fecho universal da disjunção dos literais de C for satisfatível. Por convenção, a cadeia vazia é sempre insatisfatível. Note que os R-literais não influenciam a satisfatibilidade de uma cadeia.

O sistema formal do método EMF contém duas regras de inferência: a extensão plena e a redução plena. Estas são combinações de outras três: extensão, redução e contração, a seguir definidas.

No que se segue, entendemos por uma renomeação para uma cadeia B em presença de uma cadeia A, uma substituição β tal que A e $B\beta$ não possuem variáveis em comum. Usaremos C_1C_2 para denotar a concatenação de duas cadeias C_1 e C_2. A notação |L| representa a fórmula atômica F, onde L é o literal F ou o literal $\neg$F. Além disto, a abreviação u.m.g. significa unificador mais geral cuja definição pode ser encontrada em [CASA87].

Definição 5: Extensão, Redução, Contração, Extensão Plena e Redução Plena.

Sejam A′, A″ e C cadeias e β uma renomeação para A″ em presença de A′.
Seja L′ o elemento mais à esquerda de A′ e suponha que L′ seja um literal.

(a) Uma cadeia A é uma *extensão* de A′ por A″ se e somente se existe um literal L″ em A″ e uma substituição θ tais que:

(i) L′ e L″ têm sinais opostos e θ é um u.m.g. de $\{|L'|,|L''\beta|\}$;

(ii) $A = B''B'$, onde B″ é a cadeia $A''\beta\theta$ com o literal $L''\beta\theta$ removido e B′ é a cadeia $A'\theta$ com o literal $L'\theta$ transformado em um R-literal.

(b) Uma cadeia A é uma *redução* de A′ se e somente se existe um R-literal M′ em A′ e uma substituição θ tais que:

(i) L′ e M′ têm sinais opostos e θ é um u.m.g. de $\{|L'|,|M'|\}$;

(ii) A é a cadeia $A'\theta$ com o literal $L'\theta$ removido.

(c) Uma cadeia A é a *contração* de C se e somente se A é obtida removendo-se repetidamente o elemento mais à esquerda de C enquanto este for um R-literal.

(d) Uma cadeia A é uma *extensão plena* de A′ por A″ se e somente se A é a contração de uma extensão de A′ por A″.

(e) Uma cadeia A é uma *redução plena* de A′ se e somente se A é a contração de uma redução de A′.

Definição 6: Sistema Formal da Eliminação de Modelos Fraca.

O *Sistema Formal da Eliminação de Modelos Fraca*, ou *S-EMF*, consiste de:

-Classe de Linguagens: Linguagens das Cadeias.

-Axiomas: Nenhum.

-Regras de Inferência: Sejam A′ e A″ cadeias.

(i) Extensão Plena (EX):
Se A é uma extensão plena de A′ por A″, então derive A de A′ e A″;

(ii) Redução Plena (RD):
Se A é uma redução plena de A′, então derive A de A′.

Definição 7: EMF-Dedução e EMF-Refutação.

Seja Q um conjunto de cadeias elementares e B uma cadeia em Q.

(a) Uma *EMF-dedução* de uma cadeia C a partir de Q, iniciando-se em B, é uma seqüência $(C_1,C_2,\ldots,C_n)$, $n \geq 1$, de cadeias tal que:

(i) $C_1 = B$ e $C_n = C$;

(ii) para todo i, $1 < i \leq n$, C_i é uma redução plena de C_{i-1} ou uma extensão plena de C_{i-1} por uma cadeia em Q.

(b) Uma *EMF-refutação* a partir de Q é uma EMF-dedução da cadeia vazia a partir de Q.

Definição 8: Método da Eliminação de Modelos Fraca.

O *Método da Eliminação de Modelos Fraca*, ou *EMF*, consiste do par (S-EMF,EMF-D), onde S-EMF é o Sistema Formal da Eliminação de Modelos Fraca e EMF-D é o conjunto das EMF-deduções.

Conforme mencionado anteriormente, o método EMF é refutacionalmente correto e completo. Para uma demonstração veja [LOVE69].

3. O MÉTODO EMF/GLD

Esta seção apresenta a proposta deste trabalho: o Método da Eliminação de Modelos Fraca com Geração de Lemas por Defaults, ou EMF/GLD. Antes da definição do método, examinaremos a sintaxe e a semântica da forma de representação de conhecimento com a qual o método trabalha, denominada Teoria com Cadeias.

3.1. Teoria com Cadeias

3.1.1. Sintaxe

O método EMF/GLD trabalha não somente com um conjunto de cadeias elementares, como o método EMF, mas também utiliza informações embutidas em dois conjuntos de símbolos predicativos.

Definição 9: Teoria com Cadeias.

Seja A um alfabeto de primeira ordem.
Uma *teoria com cadeias* sobre A é uma tripla, $T=(Q,P_p,P_n)$, onde:

(i) Q é um conjunto satisfatível de cadeias elementares sobre A, chamado *conjunto de cadeias* de T;

(ii) P_p é um conjunto de símbolos predicativos de A, chamado *conjunto de símbolos predicativos positivos* de T;

(iii) P_n é um conjunto de símbolos predicativos de A, chamado *conjunto de símbolos predicativos negativos* de T.

Exemplo 1: Teoria com Cadeias.

$T=(Q,P_p,P_n)$ é uma teoria com cadeias, onde:

(i) $Q=\{\neg p(x)q(x), p(a), r(b)\}$;

(ii) $P_p=\{p,q\}$;

(iii) $P_n=\{q\}$.

A função dos conjuntos P_p e P_n é permitir que sejam indicados os símbolos predicativos sobre os quais deseja-se aplicar a hipótese do mundo fechado. A partir da teoria com cadeias T do exemplo anterior, poderíamos inferir "$\neg q(b)$", dado que "$q(b)$" não é conseqüência lógica de Q e $q \in P_n$. Entretanto, não poderíamos inferir "$\neg r(a)$" pois, apesar de "$r(a)$" não ser conseqüência lógica de Q, $r \notin P_n$.

3.1.2. Semântica

A semântica das teorias com cadeias caracteriza-se por um conjunto de extensões no sentido da Lógica de Defaults. Estas extensões são definidas por defaults da forma $(:L/L)$, onde L é um literal básico cujo símbolo predicativo pertence a P_p, se L for positivo, ou a P_n, se L for negativo. Portanto, existe uma teoria com defaults associada a cada teoria com cadeias. Estes defaults justificam o procedimento de geração de lemas no processo de dedução. Na realidade, a geração de um lema L equivale à utilização de um default da forma $(:L/L)$. A seguir, apresentaremos a formalização destes conceitos.

Definição 10: Mapeamento de uma Teoria com Cadeias em uma Teoria com Defaults.

Seja A um alfabeto de primeira ordem.
Seja $T=(Q,P_p,P_n)$ uma teoria com cadeias sobre A.
O *mapeamento de* T *em uma teoria com defaults*, denotado por $\delta(T)$, é definido como $\delta(T)=(D_p \cup D_n,Q)$, onde:

$D_p=\{\ :q(\bar{t})/q(\bar{t})\ |\ q \in P_p$ e $\bar{t}=t_1,t_2,...,t_m$, onde t_i, $1 \leq i \leq m$, são termos básicos sobre A e m é a aridade de q $\}$ e

$D_n=\{\ :\neg q(\bar{t})/\neg q(\bar{t})\ |\ q \in P_n$ e $\bar{t}=t_1,t_2,...,t_m$, onde t_i, $1 \leq i \leq m$, são termos básicos sobre A e m é a aridade de q $\}$.

Conforme definido, uma teoria com defaults é formada por um conjunto de defaults e um conjunto de sentenças. Na definição acima, o conjunto de cadeias Q representa, sem nenhuma restrição, o conjunto de sentenças da teoria com defaults $\delta(T)$, dado que uma cadeia $L_1L_2...L_n$ equivale à sentença $\forall[L_1 \vee L_2 \vee ... \vee L_n]$.

Observe que, para qualquer teoria com cadeias T, $\delta(T)$ é fechada.

Exemplo 2: Mapeamento de uma Teoria com Cadeias em uma Teoria com Defaults.

Seja $T=(Q,P_p,P_n)$ a teoria com cadeias do Exemplo 1. Então $\delta(T)=(D,S)$, onde (considere "a" e "b" as únicas constantes do alfabeto em questão):

(i) $D=\{:p(a)/p(a),\ :p(b)/p(b),\ :q(a)/q(a),\ :q(b)/q(b),\ :\neg q(a)/\neg q(a),\ :\neg q(b)/\neg q(b)\}$;

(ii) $S=\{\neg p(x)q(x),\ p(a),\ r(b)\}$.

Definição 11: Extensão de uma Teoria com Cadeias.

Um conjunto de sentenças E é uma *extensão* de uma teoria com cadeias T se e somente se E é uma extensão de $\delta(T)$.

Exemplo 3: Extensão de uma Teoria com Cadeias.

Seja $T=(Q,P_p,P_n)$ a teoria com cadeias do Exemplo 1.
Então os seguintes conjuntos são extensões de T:

(a) $E_1 = Th(Q \cup \{p(b)\})$

(b) $E_2 = Th(Q \cup \{\neg q(b)\})$

Um novo conceito de conseqüência lógica será definido para relacionar sentenças às teorias com cadeias.

Definição 12: Conseqüência Lógica Fraca de uma Teoria com Cadeias.

Seja T uma teoria com cadeias e F uma sentença.
F é *conseqüência lógica fraca* de T se e somente se F pertence a alguma extensão de T.

3.2. Definição do Método EMF/GLD

Esta subseção define o Método da Eliminação de Modelos Fraca com Geração de Lemas por Defaults, ou EMF/GLD. O desenvolvimento deste método de dedução objetiva um procedimento capaz de determinar, de forma correta e completa, se uma sentença é conseqüência lógica fraca de uma teoria com cadeias.

O Sistema Formal da Eliminação de Modelos Fraca com Geração de Lemas por Defaults, ou S-GLD, além das regras do sistema formal S-EMF, possui a regra da contração por defaults plena. Dada uma teoria com cadeias $T=(Q,P_p,P_n)$, esta regra consiste, basicamente, em eliminar o primeiro literal L da cadeia em questão, onde o símbolo predicativo de L pertence a P_p e L é negativo ou pertence a P_n e L é positivo. Conforme veremos mais adiante, o método EMF/GLD prevê, mediante a aplicação desta regra, a adição do lema $\neg L$ ao conjunto de lemas gerados. Seguem as definições da regra e do sistema formal novos.

Definição 13: Contração por Defaults Plena.

Seja (P_p,P_n) um par de conjuntos de símbolos predicativos.
Seja A′ uma cadeia da forma "$L_1L_2...L_n$", $n \geq 1$.
A cadeia A é a *contração por defaults plena* de A′ em presença de (P_p,P_n) se e somente se A é a contração da cadeia "$L_2...L_n$" e L_1 é positivo e seu símbolo predicativo pertence a P_n ou é negativo e seu símbolo predicativo pertence a P_p.

Exemplo 4: Contração por Defaults Plena.

Sejam $P_p=\{q\}$ e $P_n=\{p\}$ conjuntos de símbolos predicativos.

(a) A cadeia "$\square$" é a contração por defaults plena da cadeia "$p(a)[\neg s(a)][\neg s(b)]$" em presença de (P_p,P_n).

(b) A cadeia "p(a)" é a contração por defaults plena da cadeia "$\neg q(a)p(a)$" em presença de (P_p,P_n).

Definição 14: Sistema Formal da Eliminação de Modelos Fraca com Geração de Lemas por Defaults.

O *Sistema Formal da Eliminação de Modelos Fraca com Geração de Lemas por Defaults*, ou *S-GLD*, consiste de:

-Classe de Linguagens: Conjunto das Teorias com Cadeias.

-Axiomas: Nenhum.

-Regras de Inferência: Sejam A′ e A″ cadeias.

(i) Extensão Plena (EX):
Se A é uma extensão plena de A′ por A″, então derive A de A′ e A″;

(ii) Redução Plena (RD):
Se A é uma redução plena de A′, então derive A de A′;

(iii) Contração por Defaults Plena (CDP):
Seja (P_p,P_n) um par de conjuntos de símbolos predicativos.
Se A é a contração por defaults plena de A′ em presença de (P_p,P_n), então derive A de A′ e (P_p,P_n).

O método EMF/GLD baseia-se neste sistema formal e caracteriza-se pelo procedimento de geração de lemas amarrado à definição de dedução. Conforme anunciado anteriormente, o teste de consistência dos lemas gerados serão parametrizados.

Cada passo da dedução parametrizada consiste em utilizar uma das regras de inferência do sistema formal S-GLD e, em seguida, aplicar um teste de aceitação local ao resultado da utilização da regra. Ao término da dedução, o resultado é submetido a um teste de aceitação final.

Estes testes de aceitação local e final são funções parciais que funcionam como parâmetros, cujo objetivo é definir a forma do teste de consistência dos lemas gerados, assim como o momento ao longo da dedução no qual deve ser realizado. A seguir, definiremos estes testes de aceitação. Cabe ressaltar, entretanto, que as definições seguintes não apresentam os testes de consistência, mas contêm apenas as condições necessárias para que uma função parcial seja um teste de aceitação local ou final. Na Seção 4, apresentaremos duas combinações de testes de aceitação local e final que formam exemplos de parametrizações e definem, então, os testes de consistência. No que se segue, representaremos o conjunto das partes de um conjunto C por $\mathcal{P}(C)$.

Definição 15: Teste de Aceitação Local.

Seja A um alfabeto de primeira ordem.
Seja C o conjunto das cadeias sobre A, L o conjunto de literais sobre A, Lb o conjunto de literais básicos sobre A e B o conjunto das substituições formadas por variáveis de A e por termos sobre A.
Um *teste de aceitação local* é uma função parcial $\varphi_L: C \times \mathcal{P}(Lb) \times \mathcal{P}(L) \times B \times \mathcal{P}(L) \times \mathcal{P}(C) \rightarrow C \times \mathcal{P}(Lb) \times \mathcal{P}(L)$ tal que, se $\varphi_L(C',S',N',\theta,L,Q)=(C,S,N)$, então existe uma substituição $\beta \in B$ tal que:

(i) $C=C'\beta$;

(ii) $S' \subseteq S$;

(iii) $S \cup N = S' \cup N'\theta\beta \cup L\beta$.

Definição 16: Teste de Aceitação Final.

Seja A um alfabeto de primeira ordem.
Seja C o conjunto das cadeias sobre A, L o conjunto de literais sobre A, Lb o conjunto de literais básicos sobre A e Bb o conjunto das substituições formadas por variáveis de A e por termos básicos sobre A.
Um *teste de aceitação final* é uma função parcial $\varphi_F: C \times \mathcal{P}(Lb) \times \mathcal{P}(L) \times \mathcal{P}(C) \rightarrow C \times \mathcal{P}(Lb)$ tal que, se $\varphi_F(C',S',N,Q)=(C,S)$, então existe uma substituição $\beta \in Bb$ tal que:

(i) $C=C'\beta$;

(ii) $S=S' \cup N\beta$.

O teste de consistência de um lema só pode ser executado se este for básico dado que, de acordo com a semântica das teorias com cadeias, os defaults que justificam a geração de lemas são da forma (:B/B), onde B é um literal básico. Porém, o método EMF/GLD permite, dependendo da parametrização, que lemas não básicos sejam gerados e que o teste de consistência não seja feito no momento da geração do lema. Portanto, se em dado momento a consistência de um lema não básico tiver que ser testada, então este lema deve, de alguma forma, ser instanciado. Esta necessidade de instanciação nos fez prever a substituição β nas definições de teste de aceitação local e final. A substituição vazia poderá ser adotada sempre que a parametrização em questão não considerar tais instanciações.

Estas instanciações forçadas, diante da necessidade de testar um lema, consistem em substituir as variáveis do lema por constantes não pertencentes ao alfabeto de primeira ordem adotado, às quais chamaremos constantes de tipicidade. Estas constantes agem como as constantes de Skolem introduzidas pelo processo de Skolemização da fórmula $\exists[L_1 \wedge L_2 \wedge \ldots \wedge L_n]$, onde $L_1, L_2, \ldots, L_n$ são os lemas a serem testados, dado que o teste de consistência a ser feito deve verificar se $\exists[L_1 \wedge L_2 \wedge \ldots \wedge L_n]$ é consistente em relação ao conjunto de cadeias acrescido dos lemas testados. Devemos, então, considerar o alfabeto de primeira ordem acrescido das constantes de tipicidade. Os fundamentos da utilização das constantes de tipicidade, também chamadas "elementos típicos", podem ser encontrados em [GUER89,SILV88].

A definição de dedução neste novo método deve, de alguma forma, guardar os lemas gerados para que possam ser utilizados em derivações posteriores. Deve, ainda, possuir uma estrutura que mantenha separados os lemas testados e os não testados. Portanto, formalizaremos a dedução como uma seqüência de triplas formadas por cadeias e dois conjuntos de lemas, de modo que a tripla derivada contenha a cadeia derivada, o conjunto de lemas testados e o conjunto de lemas não testados até o momento.

No que se segue, representaremos a substituição vazia por ε. Além disto, uma *representação em cadeias* de uma fórmula F, denotada por CD(F), é um conjunto de cadeias elementares tal que F é satisfatível se e somente se CD(F) é satisfatível. Observe que um conjunto de cadeias elementares pode ser considerado, sem nenhuma restrição, como um conjunto de cláusulas. Desta forma, segundo o Algoritmo de Representação Clausal apresentado em [CASA87], toda fórmula possui uma representação clausal e, conseqüentemente, uma representação em cadeias.

Definição 17: EMF/GLD-Dedução.

Seja $T=(Q,P_p,P_n)$ uma teoria com cadeias e F uma sentença.
Seja B uma cadeia em $CD(\neg F)$ e C uma cadeia qualquer.
Seja φ_L um teste de aceitação local e φ_F um teste de aceitação final.
Uma *EMF/GLD-dedução* de C a partir de T e $CD(\neg F)$, iniciando-se em B, com testes φ_L e φ_F é uma seqüência de triplas $((C_1,S_1,N_1),(C_2,S_2,N_2),\ldots,(C_n,S_n,N_n))$, $n \geq 1$, onde C_i, $1 \leq i \leq n$, é uma cadeia, S_i é um conjunto de literais básicos e N_i é um conjunto de literais, chamados, respectivamente, *conjunto de lemas testados* e *conjunto de lemas não testados*, tais que:

(i) $C_1=B$, $S_1=\emptyset$ e $N_1=\emptyset$;

(ii) $(C,S)=\varphi_F(C_n,S_n,N_n,Q)$ tal que S é um conjunto de literais básicos chamado *conjunto final de lemas*;

(iii) para todo i, $1<i\leq n$, a tripla (C_i,S_i,N_i) é derivada de $(C_{i-1},S_{i-1},N_{i-1})$ através da utilização de uma das regras de inferência do sistema formal S-GLD e da aplicação de φ_L conforme indicado abaixo:

- $(C_i,S_i,N_i)=\varphi_L(C_i',S_{i-1},N_{i-1},\theta_{i-1},\emptyset,Q)$, onde
  C_i' é uma extensão plena de C_{i-1} por uma cadeia A pertencente ao conjunto $Q \cup CD(\neg F) \cup S_{i-1} \cup N_{i-1}$ com substituição θ_{i-1}, tal que se $A \in N_{i-1}$, então C_{i-1} e A não possuem variáveis em comum (*), ou
- $(C_i,S_i,N_i)=\varphi_L(C_i',S_{i-1},N_{i-1},\theta_{i-1},\emptyset,Q)$, onde
  C_i' é uma redução plena de C_{i-1} com substituição θ_{i-1}, ou
- $(C_i,S_i,N_i)=\varphi_L(C_i',S_{i-1},N_{i-1},\theta_{i-1},\{\neg L\},Q)$, onde
  C_i' é a contração por defaults plena de C_{i-1} em presença de (P_p,P_n), L é o literal mais à esquerda de C_{i-1} e $\theta_{i-1}=\varepsilon$. "$\neg L$" é o *lema* gerado.

(*) Esta restrição torna desnecessário um controle de renomeações das variáveis dos lemas. Note que isto não causará danos ao método dado que, se $A \in (S_{i-1} \cup N_{i-1})$, então existe a opção de obter C_i' através da regra da contração por defaults plena. O resultado é equivalente.

Observe que a geração de um lema "$\neg L$" está associada à eliminação do literal "L" da cadeia em questão, ou melhor, à utilização da regra da contração por defaults plena.

Uma EMF/GLD-dedução (indicada pela seqüência de triplas entre colchetes) poderia ser esquematizada da seguinte forma (RI indica a aplicação de uma regra de inferência):

$$[C_1,S_1,N_1]\rightarrow RI\rightarrow(C_1',S_1,N_1)\rightarrow\varphi_L\rightarrow[C_2,S_2,N_2]\rightarrow RI\rightarrow(C_2',S_2,N_2)\rightarrow\varphi_L\rightarrow\ldots$$

$$\ldots\rightarrow\varphi_L\rightarrow[C_i,S_i,N_i]\rightarrow RI\rightarrow(C_i',S_i,N_i)\rightarrow\varphi_L\rightarrow\ldots\rightarrow\varphi_L\rightarrow[C_n,S_n,N_n]\rightarrow\varphi_F\rightarrow\langle C,S\rangle.$$

Definição 18: EMF/GLD-Refutação e EMF/GLD-Prova.

Seja T uma teoria com cadeias e F uma sentença.
Seja φ_L um teste de aceitação local e φ_F um teste de aceitação final.

(a) Uma *EMF/GLD-refutação* a partir de T e CD($\neg$F) com testes φ_L e φ_F é uma EMF/GLD-dedução da cadeia vazia a partir de T e CD($\neg$F) com testes φ_L e φ_F.

(b) Uma *EMF/GLD-prova* de F a partir de T com testes φ_L e φ_F é uma EMF/GLD-refutação a partir de T e CD($\neg$F) com testes φ_L e φ_F.

Definição 19: Método da Eliminação de Modelos Fraca com Geração de Lemas por Defaults.

O *Método da Eliminação de Modelos Fraca com Geração de Lemas por Defaults*, ou *EMF/GLD*, consiste do par (S-GLD,GLD-D), onde S-GLD é o Sistema Formal da Eliminação de Modelos com Geração de Lemas por Defaults e GLD-D é o conjunto das EMF/GLD-deduções.

4. EXEMPLOS DE PARAMETRIZAÇÕES

A fim de ilustrar o funcionamento do método EMF/GLD, apresentaremos a seguir duas parametrizações do método, ou seja, duas combinações de testes de aceitação local e final. Além disto, avaliaremos estas parametrizações com alguns exemplos. Outras parametrizações podem ser encontradas em [CARV90].

Exemplo 5:

Na parametrização apresentada neste exemplo, os lemas básicos são testados assim que gerados e os não básicos assim que instanciados por substituições relativas às regras de extensão plena e redução plena utilizadas durante a dedução. O teste de aceitação final, neste exemplo, instancia com constantes de tipicidade e testa os lemas que ainda tenham variáveis livres ao final da dedução.

Seja φ_{L1} o teste de aceitação local, chamado Teste Local Básico, definido por $\varphi_{L1}(C',S',N',\theta,L,Q)=(C,S,N)$ tal que:

(i) $C=C'$;

(ii) $S=S'\cup B'$, onde B' é o subconjunto dos literais básicos de $N'\theta\cup L$ e $Q\cup S'\cup B'$ é satisfatível;

(iii) $N=(N'\theta\cup L)-B'$.

Seja φ_{F1} o teste de aceitação final, chamado Teste Final com Instanciação, definido por $\varphi_{F1}(C',S',N,Q)=(C,S)$ tal que:

(i) $C=C'\beta$;

(ii) $S=S'\cup N\beta$, onde $N\beta$ é básico e $Q\cup S'\cup N\beta$ é satisfatível.

(iii) $\beta=\{x_1/e_1,x_2/e_2,\ldots,x_n/e_n\}$, $n\geq 0$, onde $x_1,x_2,\ldots,x_n$ são as variáveis dos literais de N e $e_1,e_2,\ldots,e_n$ são constantes de tipicidade incluídas no alfabeto de primeira ordem adotado;

O objetivo de verificar, no teste de aceitação local (ou no teste de aceitação final), a satisfatibilidade de QUS'UB' (ou QUS'UNβ) é assegurar que cada um dos lemas em B' (ou Nβ) é consistente em relação a QUS'. Lembre-se que testar a satisfatibilidade de um conjunto de cadeias é um procedimento não decidível. Porém, em uma implementação, poderíamos substituir este teste, por exemplo, pelo procedimento que consiste em verificar se todas as árvores de EMF-refutação (veja [CASA87]) para QUS'UB' (ou QUS'UNβ), iniciando-se em cada um dos lemas em B' (ou Nβ), são finitas e não possuem ramos de sucesso.

Vejamos, então, alguns exemplos de EMF/GLD-deduções com testes φ_{L1} e φ_{F1}:

(a) Seja T=({p(a), q(b), r(b)}, $\emptyset$, {p}) uma teoria com cadeias.
Seja F=$\exists$x[$\neg$p(x)$\wedge$q(x)$\wedge$r(x)].

A seqüência abaixo, a partir da tripla (4), representa uma EMF/GLD-prova de F a partir de T com testes φ_{L1} e φ_{F1}.

1.	p(a)	.cadeia de T
2.	q(b)	.cadeia de T
3.	r(b)	.cadeia de T
4.	(p(x)$\neg$q(x)$\neg$r(x), $\emptyset$, $\emptyset$)	.cadeia de CD($\neg$F)
5.	($\neg$q(x)$\neg$r(x), $\emptyset$, {$\neg$p(x)})	.CDP 4
6.	($\neg$r(b), {$\neg$p(b)}, $\emptyset$)	.EX 5,2
7.	($\square$, {$\neg$p(b)}, $\emptyset$)	.EX 6,3

Conjunto Final de Lemas: S={$\neg$p(b)}.

Observe que, na obtenção da tripla (6), o Teste Local Básico está definido para ($\neg$r(b),$\emptyset$,{$\neg$p(x)},{x/b},$\emptyset$,{p(a),q(b),r(b)}), pois o lema "$\neg$p(b)" é consistente em relação ao conjunto {p(a),q(b),r(b)}.

(b) Seja T=({$\neg$p(a)$\vee\neg$p(b), q(b)}, {p}, $\emptyset$) uma teoria com cadeias.
Seja F=[p(a)$\wedge$p(b)$\wedge$q(b)].

A seqüência abaixo representa uma tentativa fracassada e exaustiva de se obter uma EMF/GLD-prova de F a partir de T com testes φ_{L1} e φ_{F1}. Realmente, F não é conseqüência lógica fraca de T.

1.	$\neg$p(a)$\neg$p(b)	.cadeia de T
2.	q(b)	.cadeia de T
3.	($\neg$p(a)$\neg$p(b)$\neg$q(b), $\emptyset$, $\emptyset$)	.cadeia de CD($\neg$F)
4.	($\neg$p(b)$\neg$q(b), {p(a)}, $\emptyset$)	.CDP 3

Observe que o Teste Local Básico não está definido para ($\neg$q(b),{p(a)},$\emptyset$,ε,{p(b)},{$\neg$p(a)$\neg$p(b),q(b)}) dado que o lema "p(b)" não é consistente em relação ao conjunto {$\neg$p(a)$\neg$p(b),q(b)}U{p(a)}. Portanto, a dedução é bloqueada.

Este exemplo evidencia a importância da geração de lemas e do armazenamento destes junto ao conjunto de cadeias de T, na medida em que a geração do lema "p(b)" foi corretamente inibida em função da existência do lema "p(a)" e da insatisfatibilidade do conjunto {$\neg$p(a)$\neg$p(b),q(b),p(a),p(b)}.

Exemplo 6:

Este exemplo caracteriza-se pela semelhança com a noção de prova "top-down" para teorias com defaults apresentada em [REIT80] na qual o teste de consistência é deixado para o final. A parametrização apresentada neste exemplo testa todos os lemas ao final da dedução. Antes dos testes, instancia com constantes de tipicidade os lemas que neste momento permanecem abertos.

Seja φ_{L2} o teste de aceitação local, chamado Teste Local Identidade, definido por $\varphi_{L2}(C',S',N',\theta,L,Q)=(C,S,N)$ tal que:

(i) $C=C'$;

(ii) $S=S'$;

(iii) $N=N'\theta \cup L$.

Seja φ_{F2} o Teste Final com Instanciação definido no Exemplo 5.

Vejamos, então, alguns exemplos de EMF/GLD-deduções com testes φ_{L2} e φ_{F2}:

(a) Seja $T=(\{q(a)\}, \{r\}, \{p\})$ uma teoria com cadeias.
Seja $F=\exists x\exists y[\neg p(x)\wedge q(x)\wedge r(y)]$.

A seqüência abaixo, a partir da tripla (2), representa uma EMF/GLD-prova de F a partir de T com testes φ_{L2} e φ_{F2}.

1.	$q(a)$	.cadeia de T
2.	$(p(x)\neg q(x)\neg r(y), \emptyset, \emptyset)$	.cadeia de CD($\neg$F)
3.	$(\neg q(x)\neg r(y), \emptyset, \{\neg p(x)\})$	.CDP 2
4.	$(\neg r(y), \emptyset, \{\neg p(a)\})$	.EX 3,1
5.	$(\square, \emptyset, \{\neg p(a), r(y)\})$	.CDP 4

Conjunto Final de Lemas: $S=\{\neg p(a), r(e_0)\}$.

Observe a utilização da constante de tipicidade "e_0" na instanciação do lema "r(y)", que ao final da dedução possui uma variável. Note ainda que o conjunto $\{q(a),\neg p(a),r(e_0)\}$ é satisfatível.

(b) Seja $T=(\{\neg q(a)r(a),\neg r(a)s(a)\}, \emptyset, \{p\})$ uma teoria com cadeias.
Seja $F=\exists x[\neg p(x)\wedge\neg q(x)]$.

A seqüência abaixo representa uma tentativa fracassada e exaustiva de se obter uma EMF/GLD-prova de F a partir de T com testes φ_{L2} e φ_{F2}. Realmente, F não é conseqüência lógica fraca de T.

1.	$\neg q(a)\ r(a)$	.cadeia de T
2.	$\neg r(a)\ s(a)$	.cadeia de T
3.	$(p(x)\ q(x), \emptyset, \emptyset)$	.cadeia de CD($\neg$F)
4.	$(q(x), \emptyset, \{\neg p(x)\})$	.CDP 3
5.	$(r(a), \emptyset, \{\neg p(a)\})$	.EX 4,1
6.	$(s(a), \emptyset, \{\neg p(a)\})$	.EX 5,2

Note que nenhuma regra pode ser aplicada à cadeia de (6).

Para uma comparação entre as parametrizações apresentadas, observe que se o item (b) do Exemplo 5 fosse submetido à parametrização deste exemplo, a inconsistência do lema "p(a)" gerado só seria constatada ao final da dedução. Portanto, passos de dedução desnecessários seriam executados. Em compensação, se utilizarmos a parametrização do Exemplo 5 no item (b) deste exemplo, executaríamos um teste de consistência desnecessário no momento da instanciação do lema "$\neg$p(x)", dado que, mais adiante, a dedução é bloqueada. Esta comparação estimula a parametrização do teste de consistência na definição de dedução do método, pois tudo leva a crer que não existe uma única e ideal forma de definir o teste de consistência.

5. RESULTADOS

Esta seção apresenta os principais resultados obtidos acerca do método EMF/GLD. As demonstrações encontram-se em [CARV90].

Conforme formalizaremos a seguir, a correção e a completude do metodo EMF/GLD dependem da parametrização adotada.

Definição 20: Correção e Completude do Método EMF/GLD.

Seja φ_L um teste de aceitação local e φ_F um teste de aceitação final.

(a) O método EMF/GLD com parametrização φ_L e φ_F é *correto* se e somente se, para toda teoria com cadeias T e toda sentença F, se existe uma EMF/GLD-prova de F a partir de T com testes φ_L e φ_F, então F é conseqüência lógica fraca de T.

(b) O método EMF/GLD com parametrização φ_L e φ_F é *completo* se e somente se, para toda teoria com cadeias T e toda sentença F, se F é conseqüência lógica fraca de T, então existe uma EMF/GLD-prova de F a partir de T com testes φ_L e φ_F.

Definição 21: Testes de Aceitação Local e Final Coerentes.

Seja φ_L um teste de aceitação local e φ_F um teste de aceitação final.
Os testes φ_L e φ_F são *coerentes* se e somente se, para toda teoria com cadeias $T=(Q,P_p,P_n)$ e toda sentença F, se existe uma EMF/GLD-prova de F a partir de T com conjunto final de lemas S e testes φ_L e φ_F, então $Q \cup S$ é satisfatível.

O resultado a seguir garante que a correção do método EMF/GLD depende apenas da coerência dos testes de aceitação local e final adotados.

Teorema 1: Correção do Método EMF/GLD.

Sejam φ_L e φ_F testes de aceitação local e final coerentes.
Então o método EMF/GLD com parametrização φ_L e φ_F é correto.

Concluiremos, a seguir, que o método EMF/GLD é correto e completo quando adotada a parametrização do Exemplo 5, composta pelo Teste Local Básico e pelo Teste Final com Instanciação.

Observe que, para provar a correção do método, quando utilizada uma determinada parametrização, basta mostrar, segundo o Teorema 1, que os testes de aceitação local e final que a compõem são coerentes.

Teorema 2: Correção do Método EMF/GLD com uma Parametrização

O Teste Local Básico e o Teste Final com Instanciação são testes de aceitação local e final coerentes.

Teorema 3: Completude do Método EMF/GLD com uma Parametrização

Seja φ_{L1} o Teste Local Básico e φ_{F1} o Teste Final com Instanciação.
Então o método EMF/GLD com parametrização φ_{L1} e φ_{F1} é completo.

Cabe ressaltar que a parametrização do Exemplo 6, composta pelo Teste Local Identidade e pelo Teste Final com Instanciação, também torna o método EMF/GLD correto e completo, pois simula o mecanismo de prova para teoria com defaults correto e completo, apresentado em [REIT80].

6. CONCLUSÕES

Dentre as contribuições mais significativas deste trabalho para a área de programação em lógica, apontamos a criação de um método de dedução não-monotônico, baseado em uma nova noção de dedução que incorpora o procedimento de geração de lemas e a parametrização do teste de consistência destes lemas. Esta parametrização permite que a forma de proceder ao teste de consistência e o momento ao longo da dedução no qual é realizado sejam convenientemente definidos. Destacamos ainda a definição de duas parametrizações que tornam o método proposto correto e completo.

A partir destes resultados, indicamos a seguir possíveis direções para trabalhos futuros:

- Antes de qualquer iniciativa no sentido de implementar um sistema de programação em lógica baseado nos conceitos apresentados neste trabalho, sugerimos amarrar ao método proposto um procedimento de computação de respostas.
- A fim de obter conclusões mais fortes a partir do raciocínio não-monotônico, seria interessante poder determinar se uma sentança F é *conseqüência lógica forte* de uma teoria com cadeias T, ou seja, se F pertence a todas as extensões de T. Em [CARV90] definimos uma parametrização que confere ao método EMF/GLD a correção forte, isto é, garante que F é conseqüência lógica forte de T se existe uma EMF/GLD-prova de F a partir de T com a parametrização em questão.
- Pensamos também em estender os resultados obtidos a partir do procedimento de geração de lemas e da parametrização do teste de consistência para a Lógica de Defaults.

7. REFERÊNCIAS BIBLIOGRÁFICAS

[CARV90] A.P.Carvalho, "Um Método de Dedução Não-Monotônico Baseado em Eliminação de Modelos", Tese de Mestrado, Universidade Federal do Rio de Janeiro - COPPE, Brasil, 1990.

[CASA87] M.A.Casanova, F.A.C.Giorno e A.L.Furtado, **Programação em Lógica e a Linguagem Prolog**, Rio de Janeiro, Editora Blücher, Primeira Edição, 1987.

[GUER89] R.A.T.Guerreiro, A.Silva e M.A.Casanova, "Computing Answers in Default Logic", Annals of the IEEE International Workshop on Tools for Artificial Intelligence, USA, 1989.

[LLOY87] J.W.Lloyd, **Foundations of Logic Programming**, Springer-Verlag, Germany, Second Extended Edition, 1987.

[LOVE68] D.W.Loveland, "Mechanical theorem-proving by model elimination", Journal of the ACM, vol. 15, num. 2, 1968, 236-251.

[LOVE69] D.W.Loveland, "A Simplified Format for the Model Elimination Theorem-Proving Procedure", Journal of the ACM, vol. 16, num. 3, 1969, 349-363.

[LOVE78] D.W.Loveland, **Automated Theorem Proving: a Logical Basis**, North-Holland Publishing Company, 1978.

[REIT78] R.Reiter, "On Closed World Databases", **Logic and Databases**, H. Gallaire and J. Minker (eds.), Plenum Press, New York, 1978, 55-76.

[REIT80] R.Reiter, "A Logic for Default Reasoning", Artificial Intelligence, vol. 13, nums. 1-2, 1980, 81-132.

[SILV88] A.Silva, "Fundamentos de Programação em Cláusulas Genéricas e Defaults por Eliminação de Modelos", Tese de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, Brasil, 1988.