Resolução com a lógica de primeira ordem

1 Unificação

A aplicação do algoritmo de resolução à lógica dos predicados exige mais uma noção, que é a unificação.

Para entender o conceito de unificação, precisamos definir a noção de substituição. Uma substituição é um conjunto finito de associações entre variáveis e expressões no qual cada uma é associada a no máximo uma expressão. Eis um exemplo de substituição:

$$\{x/A, y/F(B), z/w\}$$

Nese exemplo, a variável x é associada à constante A, a variável y à estrutura funcional $F(B)$ e a variável z a uma outra variável w.

Suponhamos ϕ um enunciado e σ uma substituição. Escrevemos SUBST(σ, ϕ) a aplicação da substituição σ ao enunciado ϕ. Por exemplo, com $\sigma = \{x/A, y/F(B), z/w\}$ e $\phi = p(x, K, z)$, a aplicação SUBST(σ, ϕ) retorna o enunciado $p(A, K, w)$.

Com essa definição, podemos agora definir a unificação. A idéia da unificação é de achar uma susbstitução que pode tornar semelhantes dois enunciados:

$$\text{UNIFICAR}(p, q) = \sigma, \text{ onde SUBST}(\sigma, p) = \text{SUBST}(\sigma, q)$$

A substituição σ que resulta do procedimento de unificação é chamada o **unificador** dos dois enunciados. Por exemplo, um unificador dos enunciados gosta(João,x) e gosta(y,mae(y)) é a substituição $\{y/\text{João}, x/\text{mae(João)}\}$.

Tem dois detalhes importantes a saber sobre a unificação antes de poder usá-la no algoritmo de resolução. Primeiro, pode ser necessário renomear as variáveis. Por exemplo, os enunciados gosta(x,João) e gosta(Maria,x) deveriam poder ser unificados. A variável x do primeiro enunciado não tem nada a ver com a variável x do segundo enunciado. Por isso, antes de efetuar a unificação, é importante renomear as variáveis que aparecem nos dois enunciados. No último exemplo, depois da renomeação, a unificação vai ser efetuada com os enunciados gosta(x_1,João) e gosta(Maria,x_2) e dar o resultado $\{x_1$/Maria, x_2/João$\}$.

O outro problema pode ser entendido com o seguinte exemplo de unificação, que pode retornar mais de um resultado:

UNIFICAR(gosta(João,x),gosta(y, z)) =
{ y/João, x/z }
ou
{ y/João, x/z, w/Francisco }
ou
{ y/João, x/Joao, z/João }
ou ...

Para o algoritmo de resolução funcionar bem, é importante que a unificação retorne o **unificador mais geral**, isto é, o que vai instanciar uma variável só se é preciso.

2 Algoritmo de resolução

Já sabemos que o algoritmo de resolução é baseado numa única regra de inferência: o princípio de resolução. Como a lógica dos predicados contém variáveis, temos que modificar a regra para poder utilizá-la nessa lógica:

Princípio de resolução:

Suponhamos dois literais p_j e q_k que podem ser unificados, isto é, existe uma substituição σ tal que UNIFICAR(p_j,q_k) = σ. Então:

$$\frac{[p_1, \ldots, p_j, \ldots, p_m] \qquad [q_1, \ldots, \neg q_k, \ldots, q_n]}{SUBST(\sigma, [p_1, \ldots p_{j-1}, p_{j+1} \ldots p_m, q_1, \ldots q_{k-1}, q_{k+1} \ldots q_n])}$$

Para mostrar a resolução, vamos utilizar o seguinte exemplo:

(1) $(\forall x)(\forall y)((\text{cavalo}(x) \wedge \text{cao}(y)) \supset \text{mais-rapido}(x, y))$
(2) $(\exists y)(\text{galgo}(y) \wedge (\forall z)(\text{coelho}(z) \supset \text{mais-rapido}(y, z)))$
(3) $(\forall y)(\text{galgo}(y) \supset \text{cao}(y))$
(4) $(\forall x)(\forall y)(\forall z)((\text{mais-rapido}(x, y) \wedge \text{mais-rapido}(y, z)) \supset \text{mais-rapido}(x, z))$
(5) cavalo(Rodolfo)
(6) coelho(Bobi)

A base de conhecimento, escrita sob forma normal, é a seguinte:

(1) $[\neg\text{cavalo}(v_1), \neg\text{cao}(v_2), \text{mais-rapido}(v_1, v_2)]$
(2) $[\text{galgo}(A)]$
(3) $[\neg\text{coelho}(v_3), \text{mais-rapido}(A, v_3)]$
(4) $[\neg\text{galgo}(v_4), \text{cao}(v_4)]$
(5) $[\neg\text{mais-rapido}(v_5, v_6), \neg\text{mais-rapido}(v_6, v_7), \text{mais-rapido}(v_5, v_7)]$
(6) $[\text{cavalo(Rodolfo)}]$
(7) $[\text{coelho(Bobi)}]$

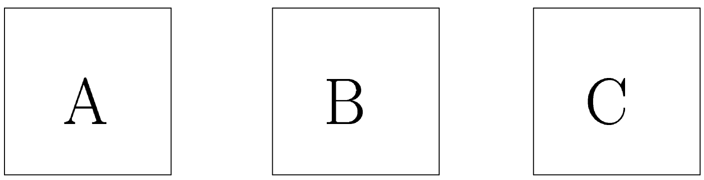
Para provar mais-rapido(Rodolfo, Bobi), acrescentamos a negação desse fato na base de conhecimento:

(8) $[\neg\text{mais-rapido(Rodolfo, Bobi)}]$

Eis as etapas da prova com o algoritmo de resolução:

(9)	$[\text{cao}(A)]$	**(2,4)**
(10)	$[\neg\text{mais-rapido}(\text{Rodolfo}, v_6), \neg\text{mais-rapido}(v_6, \text{Bobi})]$	**(5,8)**
(11)	$[\neg\text{coelho(Bobi)}, \neg\text{mais-rapido(Rodolfo, A)}]$	**(3,10)**
(12)	$[\neg\text{mais-rapido}(\text{Rodolfo}, A)]$	**(7,11)**
(13)	$[\neg\text{cavalo}(v_1), \text{mais-rapido}(v_1, \text{A})]$	**(1,9)**
(14)	$[\neg\text{cavalo(Rodolfo)}]$	**(12,13)**
(15)	$\square$	**(6,14)**

Vamos ver agora mais um exemplo que mostra a vantagem da lógica como representação. Suponhamos um mundo constituído de 3 blocos:



Suponhamos também que sabemos que o bloco A é verde e o bloco C é azul. Sem saber a cor do bloco B, podemos deduzir que existe um bloco verde que está do lado de um bloco que não é verde. Podemos deduzir isso sem conhecer a cor do bloco B. Intuitivamente, o raciocínio é simples. Se B é verde, o fato é verificado, pois ele está do lado do bloco C que não é verde. Se B não é verde, o fato é ainda verificado, pois ele está do lado do bloco A que é verde. Para poder fazer essa dedução, precisamos de uma ferramenta como a lógica dos predicados. Vamos ver agora como representar essa situação em lógica dos predicados.

Primeiro, temos que escolher bem a maneira de representar os fatos. Eis um exemplo de representação que não funciona:

(1) cor(A,Verde)
(2) cor(C,Azul)
(3) $\exists x$ cor(B,x)
(4) adjacente(A,B)
(5) adjacente(B,A)
(6) adjacente(B,C)
(7) adjacente(C,B)

Essa representação não é bastante informativa para deduzir a nossa conclusão, isto é:

(8) $(\exists x)(\exists y)(\text{adjacente}(x, y) \land \neg\text{cor}(x, \text{Verde}) \land \text{cor}(y, \text{Verde}))$

O problema é que esse enunciado não é uma conseqüência lógica da base de conhecimento. Podemos achar uma interpretação que é um modelo da base de conhecimento e que não é um modelo desse fato. Suponhamos, por exemplo, que as constantes Verde e Azul são interpretadas como designando a mesma cor. Teríamos um modelo como o seguinte:

Universo do discurso = $\{$**CorUnica, A1, B1, C1** $\}$
$\text{cor}^I = \{<\mathbf{A1, CorUnica}>, <\mathbf{B1, CorUnica}>, <\mathbf{C1, CorUnica}> \}$
$\text{adjacente}^I = \{<\mathbf{A1,B1}>, <\mathbf{B1, A1}>, <\mathbf{B1, C1}>, <\mathbf{C1, B1}> \}$
$\text{A}^I = \mathbf{A1}$
$\text{B}^I = \mathbf{B1}$
$\text{C}^I = \mathbf{C1}$
$\text{Verde}^I = \mathbf{CorUnica}$
$\text{Azul}^I = \mathbf{CorUnica}$

É claro que essa interpretação não é um modelo do fato (8).

Uma solução a esse problema é de representar também as propriedades que os objetos não têm:

(1)	cor(A,Verde)
(2)	cor(C,Azul)
(3)	$\exists x$ cor(B,x)
(4)	adjacente(A,B)
(5)	adjacente(B,A)
(6)	adjacente(B,C)
(7)	adjacente(C,B)
(8)	$\neg$ cor(A,Azul)
(9)	$\neg$ cor(C,Verde)
(10)	$\neg$ adjacente(A,C)
(11)	$\neg$ adjacente(C,A)

Assim, podemos efetuar a prova. transformando em forma normal, todas as fórmulas ficam iguais, com a exceção da fórmula (3), que será transformada em:

(3') $cor(B, K)$

A forma normal da negação do fato a provar é a seguinte:

(12) $[\neg\text{adjacente}(x, y), \text{cor}(x, \text{Verde}), \neg\text{cor}(y, \text{Verde})]$

As etapas da prova com o algoritmo de resolução são as seguintes:

(13)	$[\text{cor}(B, \text{Verde}), \neg\text{cor}(A, \text{Verde})]$	**(12,5)**
(14)	$[\text{cor}(B, \text{Verde})]$	**(13,1)**
(15)	$[\text{cor}(C, \text{Verde}), \neg\text{cor}(B, \text{Verde})]$	**(12,7)**
(16)	$[\neg\text{cor}(B, \text{Verde})]$	**(15,9)**
(17)	$\square$	**(14,16)**

Exercício 1 Seja Δ a seguinte base de conhecimento:

$(\forall x)(\forall y)(\mathsf{p}(x, y) \supset (\exists z)\mathsf{q}(z))$
$(\forall x)(\forall y)(\mathsf{q}(x) \supset \mathsf{s}(x, y))$
$(\forall x)(\forall y)(\mathsf{s}(x, y) \supset (\mathsf{r}(x) \vee \mathsf{t}(y)))$
$(\forall x)(\mathsf{r}(x) \supset \neg\mathsf{q}(x))$
$\mathsf{p}(\mathsf{Ana}, \mathsf{Paulo})$

a) Prove $\Delta \models \mathsf{t}(\mathsf{Ana}) \wedge \mathsf{t}(\mathsf{Paulo})$.
b) Prove o caso mais geral $\Delta \models \forall x\ \mathsf{t}(x)$

Exercício 2 Considere os seguintes fatos:

João gosta de todo tipo de comida.
Maçãs são comidas.
Frango é comida.
Qualquer coisa que alguém coma e que não cause sua morte é comida.
Paulo come amendoim e ainda está vivo.
Susana come tudo o que Paulo come.

a) Traduza esses fatos em fórmulas em lógica de predicados.
b) Converta as fórmulas lógica em forma clausal.
c) As cláusulas obtidas são cláusulas de Horn?
d) Prove que João gosta de amendoim usando a resolução.
e) Use a resolução para responder à pergunta: "O que Susana come?"

Exercício 3 Seja a seguinte base de conhecimento:

(1) $(\exists x)(\text{paciente}(x) \wedge (\forall y)(\text{medico}(y) \supset \text{gosta}(x, y)))$
(2) $(\forall x)(\text{paciente}(x) \supset (\forall y)(\text{politico}(y) \supset \neg\text{gosta}(x, y)))$

Queremos saber se pode-se deduzir desse base de conhecimento que nenhum médico é político, isto é, se pode-se deduzir o seguinte fato:

(3) $(\forall x)(\text{medico}(x) \supset \neg\text{politico}(x))$

Prove o fato (3), utilizando o algoritmo de resolução.

Exercício 4 Seja o seguinte conjunto de fatos:

(1) $\text{cachorro}(\text{Tina}) \wedge \text{possui}(\text{Joao}, \text{Tina})$
(2) $(\forall x)((\exists y)(\text{cachorro}(y) \wedge \text{possui}(x, y)) \supset \text{gosta-de-bicho}(x))$
(3) $(\forall x)(\text{gosta-de-bicho}(x) \supset (\forall y)(\text{bicho}(y) \supset \neg\text{mata}(x, y)))$
(4) $\text{mata}(\text{Joao}, \text{Toto}) \vee \text{mata}(\text{Titi}, \text{Toto})$
(5) $\text{gato}(\text{Toto})$
(6) $(\forall x)((\text{cachorro}(x) \vee \text{gato}(x)) \supset \text{bicho}(x))$

a) Dê em português, de maneira mais concisa possível, o significado das fórmulas (2) e (3).

b) Traduza essas fórmulas para a forma normal.
c) As claúsulas obtidas em b) são cláusulas de Horn (justifique)?
d) Utilize o algoritmo de resolução para provar que quem matou Toto é Titi.

Exercício 5 Seja uma base de conhecimento que contém somente os seguintes enunciados:

$$(\forall x)((\forall y)(\text{animal}(y) \supset \text{detesta}(x, y)) \supset (\forall z)(\text{humano}(z) \supset \text{detesta}(x, z)))$$
$$(\forall x)(\text{animal}(x) \supset \text{detesta}(\text{Joao}, x))$$
$$\neg\text{detesta}(\text{Joao,Clara})$$

Utilize o algoritmo de resolução para provar o fato ¬humano(Clara). *Cuidado:* O termo $F(x)$ não se unifica com qualquer coisa.
