Sistemas de Representação e Raciocínio – Parte 4

Introdução à Inteligência Artificial

Profa. Josiane

Baseado no material de David Poole, Alan Mackworth e Randy Goebel

Abril/2007

Raciocinando com variáveis

- Uma **instância** de um átomo ou cláusula é obtida pela substituição uniforme de variáveis por termos na cláusula
  - Pela semântica formal, se uma cláusula é verdadeira em ***I*** (uma interpretação), qualquer instância dela também é verdadeira em ***I***
  - Variáveis quantificadas universalmente
- Uma **substituição** é um conjunto finito da forma $\{V_1/t_1, \ldots, V_n/t_n\}$, onde cada V_i é uma variável distinta e cada t_i é um termo
- A **aplicação** de uma substituição $\sigma = \{V_1/t_1, \ldots, V_n/t_n\}$ em um átomo ou cláusula *e*, escrita como $e\sigma$, é a instância de *e* com cada ocorrência da variável V_i trocada por t_i

Aplicações de substituições

- Exemplos:
  - $p(a, X)\ \{X/c\} = p(a, c)$.
  - $p(Y, c)\ \{Y/a\} = p(a, c)$.
  - $p(a, X)\ \{Y/a, Z/x\} = p(a, X)$.
  - $p(X, X, Y, Y, Z)\ \{X/Z, Y/t\} = p(Z, Z, t, t, Z)$.

- Substituições podem ser aplicadas a cláusulas, átomos e termos
  - $p(X, Y) \leftarrow q(a, Z, X, Y, Z)\ \{X/Y, Z/a\} = p(Y, Y) \leftarrow q(a, a, Y, Y, a)$.

Exemplos de aplicação

- As seguintes são substituições:
  - $\sigma_1 = \{X/A, Y/b, Z/C, D/e\}$
  - $\sigma_2 = \{A/X, Y/b, C/Z, D/e\}$
  - $\sigma_3 = \{A/V, X/V, Y/b, C/W, Z/W, D/e\}$

- As seguintes mostram algumas aplicações:
  - $p(A, b, C, D)\sigma_1 = p(A, b, C, e)$
  - $p(X, Y, Z, e)\sigma_1 = p(A, b, C, e)$
  - $p(A, b, C, D)\sigma_2 = p(X, b, Z, e)$
  - $p(X, Y, Z, e)\sigma_2 = p(X, b, Z, e)$
  - $p(A, b, C, D)\sigma_3 = p(V, b, W, e)$
  - $p(X, Y, Z, e)\sigma_3 = p(V, b, W, e)$

Unificadores

- A substituição σ é um **unificador** das expressões e_1 e e_2 se $e_1\sigma = e_2\sigma$
  - $\{X/a, Y/b\}$ é um unificador de $t(a, Y, c)$ e $t(X, b, c)$
    - $t(a, Y, c)\ \{X/a, Y/b\} = t(X, b, c)\ \{X/a, Y/b\} = t(a, b, c)$
- Expressões podem ter muitos unificadores
  - $p(X, Y)$ e $p(Z, Z)$ tem $\{X/b, Y/b, Z/b\}$, $\{X/c, Y/c, Z/c\}$, $\{X/Z, Y/Z\}$
  - Dizemos que $\{X/Z, Y/Z\}$ é o **unificador mais geral** dos três porque faz menos comprometimentos com os valores das variáveis

Unificador mais geral - mgu

- A substituição σ é o **unificador mais geral** (*most general unifier* – **mgu**) de e_1 e e_2 se
  1. σ é um unificador de e_1 e e_2; e
  2. Se a substituição σ' também unifica e_1 e e_2, então $e\sigma'$ é uma instância de $e\sigma$ pata todo átomo e
- Exemplo: $e_1 = p(X, Y)$, $e_2 = p(Z, Z)$, $\sigma = \{X/Z, Y/Z\}$ e $\sigma' = \{X/c , Y/c , Z/c\}$
  1. $p(X, Y) \{X/Z, Y/Z\} = p(Z, Z)$ e $p(Z, Z) \{X/Z, Y/Z\} = p(Z, Z)$
  2. $p(X, Y) \{X/c, Y/c , Z/c\} = p(c, c)$ e $p(Z, Z) \{X/c, Y/c , Z/c\} = p(c, c)$
- $\sigma=\{X/Z, Y/Z\}$ é o mgu porque teremos várias opções para σ'
  - $\{X/a, Y/a , Z/a\}, \{X/b, Y/b , Z/b\}, \{X/K, Y/K , Z/K\}$....

Unificação - Exercício

- Faça as unificações abaixo e mostre quais são os mgus das expressões *p*(*A*, *b*, *C*, *D*) e *p*(*X*, *Y*, *Z*, *e*)
  - $\sigma_1 = \{X/A, Y/b, Z/C, D/e\}$
  - $\sigma_2 = \{A/X, Y/b, C/Z, D/e\}$
  - $\sigma_3 = \{A/V, X/V, Y/b, C/W, Z/W, D/e\}$
  - $\sigma_4 = \{A/a, X/a, Y/b, C/c, Z/c, D/e\}$
  - $\sigma_5 = \{X/A, Y/b, Z/A, C/A, D/e\}$
  - $\sigma_6 = \{X/A, Y/b, Z/C, D/e, W/a\}$
  - $\sigma_7 = \{Y/b, D/e\}$
  - $\sigma_8 = \{X/a, Y/b, Z/c, D/e\}$

Procedimento de prova *bottom-up* com variáveis

- Podemos realizar o procedimento *bottom-up* no conjunto de todas as instâncias fundamentais (sem variáveis) das cláusulas

- Uma instância fundamental da cláusula é obtida pela substituição uniforme das variáveis da cláusula por
  - Constantes que aparecem na BC
  - Ou na query
  - Se não houver constantes teremos que inventar uma

- O procedimento continua sendo correto (*sound*) e completo

Procedimento de prova *bottom-up* com variáveis - Exemplo

- BC:
  - $q(a)$.
  - $q(b)$.
  - $r(a)$.
  - $s(W) \leftarrow r(W)$.
  - $p(X, Y) \leftarrow \mathrm{q}(X) \wedge \mathrm{s}(Y)$.
- Conjunto das instâncias fundamentais:
  - $q(a)$.
  - $q(b)$.
  - $r(a)$.
  - $s(\mathrm{a}) \leftarrow r(\mathrm{a})$.
  - $s(b) \leftarrow r(b)$.
  - $p(a, a) \leftarrow \mathrm{q}(a) \wedge \mathrm{s}(a)$.
  - $p(a, \mathrm{b}) \leftarrow \mathrm{q}(a) \wedge \mathrm{s}(b)$.
  - $p(b, a) \leftarrow \mathrm{q}(b) \wedge \mathrm{s}(a)$.
  - $p(b, b) \leftarrow \mathrm{q}(b) \wedge \mathrm{s}(b)$.

Procedimento de prova *bottom-up* com variáveis - Exemplo

- Conjunto das instâncias fundamentais:
  - $q(a)$.
  - $q(b)$.
  - $r(a)$.
  - $s(\mathrm{a}) \leftarrow r(\mathrm{a})$.
  - $s(b) \leftarrow r(b)$.
  - $p(a, a) \leftarrow \mathrm{q}(a) \wedge \mathrm{s}(a)$.
  - $p(a, \mathrm{b}) \leftarrow \mathrm{q}(a) \wedge \mathrm{s}(b)$.
  - $p(b, a) \leftarrow \mathrm{q}(b) \wedge \mathrm{s}(a)$.
  - $p(b, b) \leftarrow \mathrm{q}(b) \wedge \mathrm{s}(b)$.
- O procedimento de prova irá derivar o conjunto C:
  - $\mathrm{C} = \{r(a)\}$
  - $\mathrm{C} =\{r(a), s(a)\}$
  - $\mathrm{C} =\{r(a), s(a), q(a)\}$
  - $\mathrm{C} =\{r(a), s(a), q(a), p(a,a)\}$
  - $\mathrm{C} =\{r(a), s(a), q(a), p(a,a), q(b)\}$
  - $\mathrm{C} =\{r(a), s(a), q(a), p(a,a), q(b), p(b,a)\}$

Procedimento de prova *bottom-up* com variáveis - Exemplo

- BC:
  - $p(X, Y)$.
  - $g \leftarrow p(W, W)$.
- Inventamos uma constante c e o conjunto das instâncias fundamentais se torna:
  - $p(c, c)$.
  - $g \leftarrow p(c, c)$.
- O procedimento de prova irá derivar o conjunto C:
  - C = $\{p(c, c)\}$
  - C = $\{p(c, c), g\}$

Procedimento de prova *bottom-up* com variáveis – Exercício

- BC:

  imm_west(*r101*, *r103*).
  imm_west(*r103*, *r105*).
  imm_west(*r105*, *r107*).
  imm_west(*r107*, *r109*).
  imm_west(*r109*, *r111*).
  imm_west(*r131*, *r129*).
  imm_west(*r129*, *r127*).
  imm_west(*r127*, *r125*).
  imm_east(*E*, *W*) ← *imm_west*(*W*, *E*).
  next_door(*E*, *W*) ← *imm_east*(*E*, *W*).
  next_door(*W*, *E*) ← *imm_west*(*W*, *E*).
  two_doors_east(E, W) ← *imm_east(E, M)* ∧ *imm_east(M, W).*

- O que podemos derivar da BC com o procedimento *bottom-up*?

Procedimento de prova *top-down* com variáveis

- Uma **cláusula resposta generalizada** tem a forma

  $yes(t_1,...,t_k) \leftarrow a_1 \wedge a_2 \wedge ... \wedge a_m$,

onde $t_1,...,t_k$ são temos e a_1, ..., a_m são átomos.

- A **resolução SDL** desta cláusula resposta generalizada sobre a_i com a cláusula

  $a \leftarrow b_1 \wedge ... \wedge b_p$,

onde a_i e a tem um unificador mais geral θ, é

  $(yes(t_1, ...,t_k) \leftarrow a_1 \wedge ... \wedge a_{i-1} \wedge b_1 \wedge ... \wedge b_p \wedge a_{i+1} \wedge ... \wedge a_m)\theta$.

Para resolver a *query B?* com Variáveis $V_1, \ldots, V_k$:

Faça *ac* = $yes(V_1,\ldots V_k) \leftarrow B$; // a cláusula resposta generalizada

Enquanto *ac* não for uma reposta faça

Suponha que *ac* é $yes(t_1,\ldots t_k) \leftarrow a_1 \wedge a_2 \wedge \ldots \wedge a_m$

Selecione o átomo a_i no corpo de *ac*;

Escolha a clausula $a \leftarrow b_1 \wedge \ldots \wedge b_p$;

Renomeie todas as variáveis em $a \leftarrow b_1 \wedge \ldots \wedge b_p$;

Faça que θ seja o unificador mais geral de a_i e a.

Falhe caso não haja unificação;

Faça *ac* = $(yes(t_1, \ldots, t_k) \leftarrow a_1 \wedge \ldots \wedge a_{i-1} \wedge b_1 \wedge \ldots \wedge b_p \wedge a_{i+1} \wedge \ldots \wedge a_m)\theta$

Fim-enquanto

Procedimento de prova *bottom-up* com variáveis – Exercício

- BC:

  imm_west(r101, r103).

  imm_west(r103, r105).

  imm_west(r105, r107).

  imm_west(r107, r109).

  imm_west(r109, r111).

  imm_west(r131, r129).

  imm_west(r129, r127).

  imm_west(r127, r125).

  imm_east(E, W) ← *imm_west(W, E).*

  next_door(E, W) ← *imm_east(E, W).*

  next_door(W, E) ← *imm_west(W, E).*

  two_doors_east(E, W) ← *imm_east(E, M)* $\wedge$ *imm_east(M, W).*

- O que podemos derivar da BC com o procedimento *bottom-up*?

Procedimento de prova *top-down* com variáveis - Exemplo

- ?*two_doors_east*(R, *r107*).
- Derivação:
  - *yes*(R) ← *two_doors_east*(R, *r107*).
  - Resolve com *two_doors_east*(E_1, W_1) ← *imm_east*(E_1, M_1) ∧ *imm_east*(M_1, W_1).
    - Substituição $\{E_1/R, W_1/r107\}$
  - *yes*(R) ← *imm_east*(R, M_1) ∧ *imm_east*(M_1, *r107*).
  - A cláusula mais à esquerda resolve com *imm_east*(E_2, W_2) ← *imm_west*(W_2, E_2).
    - Substituição $\{E_2/R, W_2/M_1\}$
  - *yes*(R) ← *imm_west*(M_1, R) ∧ *imm_east*(M_1, *r107*).

Procedimento de prova *top-down* com variáveis - Exemplo

- A cláusula mais à esquerda resolve com *imm_west*(*r109, r111*).
  - Substituição {M_1/*r109*, *R*/*r111*}
- *yes*(*r111*) ← *imm_east*(*r109*, *r107*).
- Resolve com *imm_east*(E_3, W_3) ← *imm_west*(W_3, E_3).
  - Substituição {E_3/*r109*, W_3/*r107*}
- *yes*(*r111*) ← *imm_west*(*r107*, *r109*).
- Resolve com *imm_west*(*r107*, *r109*).
  - Substituição { }
- *yes*(*r111*) ←.
- Resposta *R* = *r111.*

Procedimento de prova *top-down* com variáveis - Exemplo

- *BC:*
  - $live(Y) \leftarrow connected\_to(Y, Z) \wedge live(Z)$.
  - $live(outside)$.
  - $connected\_to(w_6, w_5)$.
  - $connected\_to(w_5, outside)$.
- Query: ?$live(A)$.
  - $yes(A) \leftarrow live(A)$.
  - Resolve com $live(Y_1) \leftarrow connected\_to(Y_1, Z_1) \wedge live(Z_1)$.
    - Substituição $\{Y_1/A\}$
  - $yes(A) \leftarrow connected\_to(A, Z_1) \wedge live(Z_1)$.
  - A cláusula mais a esquerda resolve com $connected\_to(w_6, w_5)$.
    - Substituição $\{A/w_6, Z_1/w_5\}$

Procedimento de prova *top-down* com variáveis - Exemplo

- $yes(w_6) \leftarrow live(w_5)$.
- Resolve com $live(Y_2) \leftarrow connected\_to(Y_2, Z_2) \wedge live(Z_2)$.
  - Substituição $\{Y_2/w_5\}$
- $yes(w_6) \leftarrow connected\_to(w_5, Z_2) \wedge live(Z_2)$.
- A cláusula mais a esquerda resolve com $connected\_to(w_5, outside)$.
  - Substituição $\{Z_2/outside\}$
- $yes(w_6) \leftarrow live(outside)$.
- Resolve com $live(outside)$.
  - Substituição { }
- $yes(w_6) \leftarrow .$
- Resposta $A = w_6$.

Símbolos de funções

- Quase sempre queremos nos referir a indivíduos em termos de componentes.
  - Exemplo: 4:55 p.m.. Sentenças em Inglês. Uma lista de alunos.
- Assim, estenderemos a noção de **termo** de tal forma que um termo possa ser **$f(t_1,\ldots,t_n)$ onde *f* é um símbolo de função e os t_i são termos**
- Em uma interpretação com uma atribuição de variáveis, o termo $f(t_1,\ldots,t_n)$ denota um indivíduo do domínio
- Com um símbolo de função e uma constante podemos referenciar um número infinito de indivíduos

Símbolos de funções

- A única forma de usar o símbolo de função é escrever cláusulas que definem relações usando o símbolo de função
- Suponha que gostaríamos de representar horas durante o dia
  - Podemos usar *am*(H, M) para denotar H:M am, onde H é um inteiro entre 1 e 12 e M é um inteiro entre 0 e 59.
  - *am*(*10*, *38*) denota *10:38 am*
  - da mesma forma podemos definir *pm*

Usando símbolos de funções

- Podemos definir a relação *before*(T_1, T_2) que é verdadeira se T_1 ocorre antes de T_2 em um dia
  - *before*(*am*(*H1*, *M1*), *pm*(*H2*, *M2*)).
  - *before*(*am*(*12*, *M1*), *am*(*H2*, *M2*)) $\leftarrow$ H2 < 12.
  - *before*(*am*(*H1*, *M1*), *am*(*H2*, *M2*)) $\leftarrow$ H1 < H2 $\wedge$ H2 < 12.
  - *before*(*am*(*H*, *M1*), *am*(*H*, *M2*)) $\leftarrow$ M1 < M2.
  - *before*(p*m*(*12*, *M1*), p*m*(*H2*, *M2*)) $\leftarrow$ H2 < 12.
  - *before*(p*m*(*H1*, *M1*), p*m*(*H2*, *M2*)) $\leftarrow$ H1 < H2 $\wedge$ H2 < 12.
  - *before*(p*m*(*H*, *M1*), p*m*(*H*, *M2*)) $\leftarrow$ M1 < M2.

Procedimento de prova com símbolos de funções

- A diferença dos procedimentos anteriores é que a classe de termos é expandida com os símbolos de funções
- Com símbolos de funções existem infinitamente muitos termos
- Procedimento de prova *bottom-up*
  - Temos que assegurar que o critério de seleção para selecionar as cláusulas seja imparcial
  - Um **critério de seleção imparcial** (*fair*) é aquele que qualquer cláusula que está disponível para ser escolhida será escolhida em algum momento
  - É completo somente se o critério de seleção for imparcial

Procedimento de prova com símbolos de funções

- Procedimento de prova *top-down*
  - Usa o unificador mais geral
  - Temos que tomar cuidado com: a variável *X* não unifica com o termo *t* no qual *X* ocorre, e ele não é o próprio *X*
    - Suponha que o símbolo de função *s* denota a função sucessor, onde *s*(*n*) é o número depois de n
    - Nesta interpretação não existe nenhum nenhum número X que é igual ao seu sucessor, não faz sentido unifica {X/s(X)}
  - Se isso for permitido o procedimento de prova torna-se incorreto (*unsound*)

Procedimento de prova *top-down* com símbolos de funções

- Considere a cláusula:
  - $append(c(A, X), Y, c(A, Z)) \leftarrow append(X, Y, Z))$.
  - $append(nil, Z, Z)$.
- Considere a query:
  - $?append(\mathrm{F}, c(L, nil), c(l, c(i, c(s, c(t, \mathrm{nil})))))$.
  - $\mathrm{yes}(F, L) \leftarrow append(\mathrm{F}, c(L, nil), c(l, c(i, c(s, c(t, \mathrm{nil})))))$.
  - Resolve com $append(c(A_1, X_1), Y_1, c(A_1, Z_1)) \leftarrow append(X_1, Y_1, Z_1))$.
    - Substituição $\{\mathrm{F}/c(l, X_1), Y_1/c(L, nil), \mathrm{A}_1/l, Z_1/c(i, c(s, c(t, \mathrm{nil})))\}$
  - $\mathrm{yes}(c(l, X_1), L) \leftarrow append(\mathrm{X}_1, c(L, nil), c(i, c(s, c(t, \mathrm{nil}))))$.
  - Resolve com $append(c(A_2, X_2), Y_2, c(A_2, Z_2)) \leftarrow append(X_2, Y_2, Z_2))$.
    - Substituição $\{\mathrm{X}_1/c(i, X_2), Y_2/c(L, nil), \mathrm{A}_2/i, Z_2/c(s, c(t, \mathrm{nil}))\}$
  - $\mathrm{yes}(c(l,c(i, X_2)), L) \leftarrow append(\mathrm{X}_2, c(L, nil), c(s, c(t, \mathrm{nil})))$.

Procedimento de prova *top-down* com símbolos de funções

- yes($c(l,c(i, X_2)), L) \leftarrow append(X_2, c(L, nil), c(s, c(t, \text{nil})))$.
  - Resolve com $append(c(A_3, X_3), Y_3, c(A_3, Z_3)) \leftarrow append(X_3, Y_3, Z_3))$.
    - Substituição $\{X_2/c(s, X_3), Y_3/c(L, nil), A_3/s, Z_3/ c(t, \text{nil}))\}$
- yes($c(l,c(i, c(s, X_3))), L) \leftarrow append(X_3, c(L, nil), c(t, \text{nil}))$.
  - Escolhendo a primeira cláusula para resolver
  - $append(c(A_4, X_4), Y_4, c(A_4, Z_4)) \leftarrow append(X_4, Y_4, Z_4))$.
    - Substituição $\{X_3/c(t, X_4), Y_4/c(L, nil), A_4/t, Z_4/\text{nil}\}$
- yes($c(l,c(i, c(s, c(t, X_4)))), L) \leftarrow append(X_3, c(L, nil), \text{nil})$.
- Não existe mais nenhuma cláusula na qual a cabeça unifica com $append(X_3, c(L, nil), \text{nil})$ e a prova falha

Procedimento de prova *top-down* com símbolos de funções

- voltando....
- yes($c(l, c(i, c(s, X_3))), L$) ← $append(X_3, c(L, nil), c(t, \text{nil}))$.
- E escolhendo a segunda cláusula: $append(nil, Z_5, Z_5)$).
  - Substituição {X_3/nil), $Z_5/c(t, nil)$, L/t}
- yes($c(l, c(i, c(s, nil))), t$) ← .
- F = $c(l, c(i, c(s, nil)))$ e L = t

Listas

- Uma lista é uma seqüência ordenada de elementos.
- Vamos usar a constante *nil* para denotar a lista vazia e a função *cons*(*H*,*T*) para denotar uma lista com o primeiro elemento *H* e o resto-da-lista *T* (esta função não é pré-definida)
- A lista contendo david, alan e randy é:
  - *cons*(*david*,*cons*(*alan*,(*cons*(*randy*,*nil*)))
- *append*(*X*,*Y*, *Z*) é verdadeiro se a lista *Z* contiver os elementos de *X* seguidos pelos elementos de *Y*
  - *append*(*nil*, *Z*, *Z*).
  - *append*(*cons*(*A*,*X*),*Y*, *cons*(*A*,*Z*)) ← *append*(*X*,*Y*,*Z*)