

Lógica Computacional

- Resolução na Lógica de Predicados
- Forma Clausal de FBFs
- Skolemização
- Unificação e Unificadores
- Algoritmo Martelli-Montanari

Resolução na Lógica de Predicados

- O método de resolução pode ser generalizado da Lógica Proposicional para a Lógica de Predicados, mas para esse efeito é necessário algumas adaptações tendo em conta a quantificação das fórmulas em causa.
- Mais especificamente, mantêm-se a noção de que uma demonstração é uma refutação, ou seja a dedução da cláusula vazia através da regra de resolução, mas é necessário:
 - a) Obter cláusulas a partir de fórmulas quantificadas
 - b) Eliminar os quantificadores existenciais
 - c) Generalizar a regra de resolução para lidar com átomos contendo variáveis universalmente quantificadas.

Cláusulas em Lógica de Predicados

- As fórmulas bem formadas (FBF) na Lógica de Predicados não podem em geral ser colocadas numa “forma CNF” já que podem conter quantificadores no seu interior.
- Desta forma para obter na Lógica de Predicados as cláusulas correspondentes a uma FBF é preciso considerar os seguintes passos iniciais:

P1. Colocar a FBF na sua Forma Prenex;

P2. Colocar a matriz na forma CNF

- O exemplo seguinte permite ilustrar este procedimento :

A: $\exists x (Cube(x) \wedge \exists y Tet(y) \wedge \forall z (Large(z) \rightarrow Between(z, x, y)))$

A1: $\exists x \exists y \forall z (C(x) \wedge T(y) \wedge (L(z) \rightarrow B(z, x, y)))$

A2: $\exists x \exists y \forall z (C(x) \wedge T(y) \wedge (\neg L(z) \vee B(z, x, y)))$

Cláusulas em Lógica de Predicados

- Os próximos exemplos ilustram outros tipos de fórmulas com 3 quantificadores:

B: $\forall x (Cube(x) \rightarrow \exists y (Tet(y) \wedge \exists z (Between(z, x, y))))$

B1: $\forall x \exists y \exists z (C(x) \rightarrow (T(y) \wedge B(z, x, y)))$

B2: $\forall x \exists y \exists z ((\neg C(x) \vee T(y)) \wedge (\neg C(x) \vee B(z, x, y)))$

C: $\exists x (Cube(x) \wedge \forall y (Tet(y) \rightarrow \exists z Between(z, x, y)))$

C1: $\exists x \forall y \exists z (C(x) \wedge (T(y) \rightarrow B(z, x, y)))$

C2: $\exists x \forall y \exists z (C(x) \wedge (\neg T(y) \vee B(z, x, y)))$

D: $\forall x (Cube(x) \rightarrow \forall y (Tet(y) \rightarrow \exists z (Between(z, x, y))))$

D1: $\forall x \forall y \exists z (C(x) \rightarrow (T(y) \rightarrow (B(z, x, y))))$

D2: $\forall x \forall y \exists z (\neg C(x) \vee \neg T(y) \vee B(z, x, y))$

Skolemização

- Considere-se a frase “*existem um ou mais cubos*” representada pela fórmula P_1

$$P_1 =_{\text{def}} \exists x \text{ Cube } (x)$$

- Esta frase não é equivalente a outra que indica que “*existe 1 cubo*”. Caso o nome c não denote qualquer outro objecto (no contexto) esta segunda frase pode utilizar a atribuição deste nome e ser representada pela fórmula Q_1

$$Q_1 =_{\text{def}} \text{Cube } (c)$$

- Com efeito, a primeira é mais “geral”, sendo **compatível** com a possibilidade de existirem 2 cubos, enquanto que a primeira não.
- No entanto, provar que é falso que “*existem um ou mais cubos*” é equivalente a provar que é falso que “*existe 1 cubo*”, tenha ele o nome c ou outro nome qualquer.
- Desta forma, no *contexto de uma refutação*, a fórmula P pode ser substituída por Q , tal como proposto por Skolem.
- Esta constatação pode ser generalizada para outras frases existenciais.

Skolemização

- Considere-se agora a frase “*para cada cubo existem um ou mais tetraedros ao seu lado*” representada pela fórmula P_2

$$P_2 =_{\text{def}} \forall \mathbf{x} \exists \mathbf{y} \left(\text{Cube}(\mathbf{x}) \rightarrow (\text{Tet}(\mathbf{y}) \wedge \text{Adjoins}(\mathbf{x}, \mathbf{y})) \right)$$

- Tal como anteriormente, esta frase não é equivalente a outra que indica que “*para cada cubo existe 1 tetraedro ao seu lado*”. Esses tetraedro, único para cada cubo \mathbf{x} , mas que pode ser diferente para cubos diferentes, pode ser denotado por $\mathbf{t}(\mathbf{x})$ em que \mathbf{t} é uma função (não utilizada no contexto), e representar a frase pela fórmula Q_2

$$Q_2 =_{\text{def}} \forall \mathbf{x} \left(\text{Cube}(\mathbf{x}) \rightarrow \text{Tet}(\mathbf{t}(\mathbf{x})) \wedge \text{Adjoins}(\mathbf{x}, \mathbf{t}(\mathbf{x})) \right)$$

- Com efeito, a primeira é mais “geral”, sendo **compatível** com a possibilidade de existirem 2 tetraedros para cada cubo, enquanto que a primeira não.
- No entanto, provar que é falso que frase “*para cada cubo existem um ou mais tetraedros ao seu lado*” é equivalente a provar que é falso que “*para cada cubo existe 1 tetraedro ao seu lado*”, tenha ele o nome $\mathbf{t}(\mathbf{x})$ ou outro nome qualquer.
- No *contexto de uma refutação*, e generalizando o resultado anterior, a fórmula **P2** pode ser substituída por **Q2**.

Skolemização

- Os exemplos anteriores justificam pois a skolemização formalizada de seguida.

Skolemização:

Dada uma fórmula bem formada na forma Prenex, a sua forma normal skolemizada obtém-se substituindo todas as ocorrências na matriz de variáveis existencialmente quantificadas por novas funções das variáveis quantificadas universalmente que (no prefixo da forma Prenex) as precedem.

- Estas funções são denominadas funções de Skolem (ou constantes de Skolem se a variável existencialmente quantificada for a primeira no prefixo).

Exemplos:

- $\exists x \exists y \forall z P(x, y, z) \rightarrow \forall z P(a, b, z)$
- $\forall x \exists y \exists z P(x, y, z) \rightarrow \forall x P(x, f(x), g(x))$
- $\exists x \forall y \exists z P(x, y, z) \rightarrow \forall y P(a, y, h(y))$
- $\forall x \forall y \exists z P(x, y, z) \rightarrow \forall x \forall y P(x, y, f(x, y))$

Forma Clausal em Lógica de Predicados

- Uma vez skolemizadas as fórmulas ficamos em condições de concluir a transformação das fórmulas iniciais para a forma clausal com os seguintes passos adicionais:

P3. **Skolemizar** a FBF na forma Prenex com matriz em CNF.

P4. **Circunscrever** os quantificadores aos conjuntos CNF.

P5. **Renomear** as variáveis universalmente quantificadas.

P6. **Separar** as cláusulas, mantendo constantes e funções.

P7. **Eliminar** os quantificadores universais, renomeando as variáveis.

- A **circunscrição** e **renomeação** das variáveis tem em conta as equivalências

$$\forall \mathbf{x} (P(\mathbf{x}) \wedge Q(\mathbf{x}))$$

$$\Leftrightarrow \forall \mathbf{x} P(\mathbf{x}) \wedge \forall \mathbf{x} Q(\mathbf{x}) \quad - \text{ circunscrição}$$

$$\Leftrightarrow \forall \mathbf{x} P(\mathbf{x}) \wedge \forall \mathbf{y} Q(\mathbf{y}) \quad - \text{ renomeação}$$

generalizáveis ao caso em que existam várias variáveis quantificadas universalmente.

- A **eliminação** dos quantificadores é uma mera “operação de cosmética”. Uma vez que todas as variáveis que se mantêm são universalmente quantificadas, e têm diferentes “nomes”, podem apagar-se os quantificadores, ficando a quantificação subentendida (senão as fórmulas resultantes não teriam significado, pois as variáveis seriam livres).

Forma Clausal em Lógica de Predicados

- Podemos agora completar as transformações das fórmulas inicialmente apresentadas.

$$\mathbf{A: \exists x (Cube(x) \wedge \exists y Tet(y) \wedge \forall z (Large(z) \rightarrow Between(z, x, y)))}$$

- Passagem à forma Prenex

$$\mathbf{A1: \exists x \exists y \forall z (C(x) \wedge T(y) \wedge (L(z) \rightarrow B(z, x, y)))}$$

- Passagem a CNF da Matriz

$$\mathbf{A2: \exists x \exists y \forall z (C(x) \wedge T(y) \wedge (\neg L(z) \vee B(z, x, y)))}$$

- Skolemização da forma Prenex

$$\mathbf{A3: \forall z (C(a) \wedge T(b) \wedge (\neg L(z) \vee B(z, a, b)))}$$

- Circunscrição de Quantificadores

$$\mathbf{A4: C(a) \wedge T(b) \wedge \forall z (\neg L(z) \vee B(z, a, b))}$$

- Renomeação de Variáveis

$$\mathbf{A5: C(a) \wedge T(b) \wedge \forall x1 (\neg L(x1) \vee B(x1, a, b))}$$

- Separação de Cláusulas

$$\mathbf{A6: \{C(a) , T(b) , \forall x1 (\neg L(x1) \vee B(x1, a, b))\}}$$

- Apagamento de Quantificadores

$$\mathbf{A7: \{C(a) , T(b) , \neg L(x1) \vee B(x1, a, b) \}}$$

Forma Clausal em Lógica de Predicados

- As outras fórmulas são transformadas similarmente:

$$\mathbf{B}: \forall \mathbf{x} (\text{Cube}(\mathbf{x}) \rightarrow \exists \mathbf{y} (\text{Tet}(\mathbf{y}) \wedge \exists \mathbf{z} (\text{Between}(\mathbf{z}, \mathbf{x}, \mathbf{y}))))$$

$$\mathbf{B1}: \forall \mathbf{x} \exists \mathbf{y} \exists \mathbf{z} (\text{C}(\mathbf{x}) \rightarrow (\text{T}(\mathbf{y}) \wedge \text{B}(\mathbf{z}, \mathbf{x}, \mathbf{y})))$$

$$\mathbf{B2}: \forall \mathbf{x} \exists \mathbf{y} \exists \mathbf{z} ((\neg \text{C}(\mathbf{x}) \vee \text{T}(\mathbf{y})) \wedge (\neg \text{C}(\mathbf{x}) \vee \text{B}(\mathbf{z}, \mathbf{x}, \mathbf{y})))$$

$$\mathbf{B3}: \forall \mathbf{x} ((\neg \text{C}(\mathbf{x}) \vee \text{T}(\mathbf{f}(\mathbf{x}))) \wedge (\neg \text{C}(\mathbf{x}) \vee \text{B}(\mathbf{g}(\mathbf{x}), \mathbf{x}, \mathbf{f}(\mathbf{x}))))$$

$$\mathbf{B4}: \forall \mathbf{x} (\neg \text{C}(\mathbf{x}) \vee \text{T}(\mathbf{f}(\mathbf{x}))) \wedge \forall \mathbf{x} (\neg \text{C}(\mathbf{x}) \vee \text{B}(\mathbf{g}(\mathbf{x}), \mathbf{x}, \mathbf{f}(\mathbf{x})))$$

$$\mathbf{B5}: \forall \mathbf{x1} (\neg \text{C}(\mathbf{x1}) \vee \text{T}(\mathbf{f}(\mathbf{x1}))) \wedge \forall \mathbf{x2} (\neg \text{C}(\mathbf{x2}) \vee \text{B}(\mathbf{g}(\mathbf{x2}), \mathbf{x2}, \mathbf{f}(\mathbf{x2})))$$

$$\mathbf{B6}: \{ \forall \mathbf{x1} (\neg \text{C}(\mathbf{x1}) \vee \text{T}(\mathbf{f}(\mathbf{x1}))), \forall \mathbf{x2} (\neg \text{C}(\mathbf{x2}) \vee \text{B}(\mathbf{g}(\mathbf{x2}), \mathbf{x2}, \mathbf{f}(\mathbf{x2}))) \}$$

$$\mathbf{B7}: \{ \neg \text{C}(\mathbf{x1}) \vee \text{T}(\mathbf{f}(\mathbf{x1})), \neg \text{C}(\mathbf{x2}) \vee \text{B}(\mathbf{g}(\mathbf{x2}), \mathbf{x2}, \mathbf{f}(\mathbf{x2})) \}$$

- Neste exemplo de transformação da fórmula **B**, esta foi considerada isoladamente em relação à fórmula **A**. Se se pretendesse transformar simultaneamente as fórmulas **A** e **B** teriam de ser usadas variáveis, constantes e funções de Skolem distintas.
- Especificamente, na transformação exemplo, da fórmula **B**, não se poderia usar a variável **x1** (usada na transformação de **A**).

Forma Clausal em Lógica de Predicados

- As outras fórmulas são transformadas similarmente:

C: $\exists x (Cube(x) \wedge \forall y (Tet(y) \rightarrow \exists z Between(z, x, y)))$

C1: $\exists x \forall y \exists z (C(x) \wedge (T(y) \rightarrow B(z, x, y)))$

C2: $\exists x \forall y \exists z (C(x) \wedge (\neg T(y) \vee B(z, x, y)))$

C3: $\forall y (C(a) \wedge (\neg T(y) \vee B(a, y, f(y))))$

C4: $C(a) \wedge \forall y (\neg T(y) \vee B(a, y, f(y)))$

C5: $C(a) \wedge \forall x1 (\neg T(x1) \vee B(a, x1, f(x1)))$

C6: $\{ C(a), \forall x1 (\neg T(x1) \vee B(a, x1, f(x1))) \}$

C7: $\{ C(a), \neg T(x1) \vee B(a, x1, f(x1)) \}$

- Uma vez mais, neste exemplo, a fórmula **C** foi considerada isoladamente das fórmulas **A** e **B**. Se se pretendesse transformar simultaneamente a fórmula **C** com as fórmulas **A** e **B**, não poderiam ser usadas a variável **x1** (usada em **A** e em **B**) nem a constante **a** (usada em **A**) nem a função **f** (usada em **B**).

Forma Clausal em Lógica de Predicados

- Finalmente para a fórmula D, igualmente vista isoladamente:

D: $\forall x (Cube(x) \rightarrow \forall y (Tet(y) \rightarrow \exists z (Between(z, x, y))))$

D1: $\forall x \forall y \exists z (C(x) \rightarrow (T(y) \rightarrow (B(z, x, y))))$

D2: $\forall x \forall y \exists z (\neg C(x) \vee \neg T(y) \vee B(z, x, y))$

D3: $\forall x \forall y (\neg C(x) \vee \neg T(y) \vee B(f(x, y), x, y))$

D4: $\forall x \forall y (\neg C(x) \vee \neg T(y) \vee B(f(x, y), x, y))$

D5: $\forall x1 \forall x2 (\neg C(x1) \vee \neg T(x2) \vee B(f(x1, x2), x1, x2))$

D6: $\{ \forall x1 \forall x2 (\neg C(x1) \vee \neg T(x2) \vee B(f(x1, x2), x1, x2)) \}$

D7: $\{ \neg C(x1) \vee \neg T(x2) \vee B(f(x1, x2), x1, x2) \}$

- De notar que neste caso a função **f/2** tem dois argumentos e nunca se confundiria com as funções **f/1**, de um só argumento, usadas na transformação de **B** e de **C**.

Unificação

- Uma vez obtidas as cláusulas que deverão ser resolvidas, há que estender a regra de Resolução para o caso em que as cláusulas contêm variáveis.
- Para justificar esta extensão, consideremos as cláusulas, num mesmo contexto, correspondentes às fórmulas indicadas,

$$\forall \mathbf{x} (P(\mathbf{x}) \rightarrow Q(\mathbf{x})) \rightarrow \neg P(\mathbf{x1}) \vee Q(\mathbf{x1})$$

$$\forall \mathbf{x} (Q(\mathbf{x}) \rightarrow R(\mathbf{x})) \rightarrow \neg Q(\mathbf{x2}) \vee R(\mathbf{x2})$$

das quais se deveria poder deduzir a fórmula e respectiva cláusula

$$\forall \mathbf{x} (P(\mathbf{x}) \rightarrow R(\mathbf{x})) \rightarrow \neg P(\mathbf{x3}) \vee R(\mathbf{x3})$$

- Esta cláusula poderia ser obtida por resolução das cláusulas iniciais em ordem a Q, mas $Q(\mathbf{x1})$ não é um “literal complementar” de $\neg Q(\mathbf{x2})$, pois usam variáveis diferentes!
- Naturalmente sendo $\mathbf{x1}$ e $\mathbf{x2}$ variáveis universalmente quantificadas, elas podem tomar qualquer valor, e portanto ser ambas aplicadas ao mesmo objecto. Mas esta situação pode complicar-se por múltiplas ocorrências das variáveis.
- Mais formalmente, para resolver duas cláusulas, é necessário que os literais complementares que aparecem nas duas cláusulas sejam **unificados**.

Unificação

- Em geral, iremos utilizar a notação \mathbf{x}_i , \mathbf{y}_i , \mathbf{z}_i , ..., para as diferentes variáveis que ocorram numa cláusula i do conjunto de cláusulas que se pretende refutar.
- Sem se poder tratar desta matéria em toda profundidade, poderemos mesmo assim utilizar as seguintes definições e exemplos:

Unificação 1:

Dois **termos** são **unificáveis** se existir uma **substituição** das suas variáveis que os torne **idênticos**.

- Nota: Nesta definição, **termo** aplica-se quer a variáveis, quer a predicados, quer a funções, que como vimos aparecem naturalmente em cláusulas, nomeadamente através da Skolemização.

Substituição:

Uma substituição σ é um conjunto de pares v_i/ω_i em que as variáveis v_i são todas diferentes, e os termos ω_i não contêm as variáveis v .

Unificação

Exemplos:

- Uma variável pode ser unificada com uma constante, uma função de outras variáveis ou outra variável,

$$\text{Ex.1: } T(y1) \approx T(a) \quad \sigma = \{ y1/a \}$$

$$\text{Ex.2: } T(y1) \approx T(f(y2)) \quad \sigma = \{ y1/f(y2) \}$$

- Tendo em atenção que as variáveis que aparecem nas cláusulas são (implicitamente) quantificadas universalmente, a unificação de uma variável com uma constante ou função corresponde basicamente à sua instanciação universal.

$$\text{Ex.3: } T(y1) \approx T(y2) \quad \sigma = \{ y1/y2 \} \text{ ou } \{ y2/y1 \}$$

- A substituição de variáveis por outras variáveis corresponde apenas à sua renomeação, quer para garantir que se passam a referir ao mesmo objecto, quer para evitar ambiguidades sobre o escopo dos quantificadores que estão implícitos nas cláusulas.
- De notar que uma variável não pode ser unificada com uma função que a contém (**occurs-check**) o que será justificado adiante.

Unificação

- A unificação de termos exige normalmente a unificação em simultâneo do conjunto de todos os seus argumentos, geralmente um conjunto com mais do que um elemento.

$$\text{Ex. 4: } T(x_1, a) \approx T(b, y_2) \quad \sigma = \{ x_1/b, y_2/a \}$$

- O próximo exemplo ilustra como as coisas se podem complicar ...

$$\text{Ex. 5: } T(x_1, y_1, f(x_1, y_1)) \approx T(x_2, g(x_2), z_2)$$

$$\sigma = \{ x_1/x_2, y_1/g(x_2), z_2/f(x_2, g(x_2)) \} \quad \text{ou}$$

$$\sigma = \{ x_2/x_1, y_1/g(x_1), z_2/f(x_1, g(x_1)) \}$$

- Neste caso, ao substituir z_2 pelo termo $f(x_1, y_1)$ há que ter em atenção que quer x_1 quer x_2 são variáveis que já foram previamente substituídas e usar os termos pelos quais foram substituídos.
- Assim sendo, uma substituição de vários termos simultaneamente pode envolver recursivamente várias substituições prévias pelo que é fundamental garantir que essa recursividade termina.

Unificação

- Occurs-Check:

$$\text{Ex. 6: } T(\mathbf{x1}, \mathbf{x1}) \approx T(\mathbf{x2}, f(\mathbf{x2})) \quad \sigma = \{ \mathbf{x1}/\mathbf{x2}, \mathbf{x2}/f(\mathbf{x2}) \}$$

- Neste caso, o par $\mathbf{x2}/f(\mathbf{x2})$ aparece **erradamente** na substituição, já que a variável a substituir, $\mathbf{x2}$, aparece no termo $f(\mathbf{x2})$, pelo qual deveria voltar a ser substituída. Assim o processo de substituição deveria continuar recursivamente, originando o termo infinito

$$\mathbf{x2}/f(f(f(\dots f(\mathbf{x2}) \dots)))$$

- De um ponto de vista semântico, consideremos o predicado $G(\mathbf{x}, \mathbf{x})$ e a função $m(\mathbf{x})$ interpretados como “x gosta de y” e “mãe de x”, respectivamente.
- Se os termos $G(\mathbf{x1}, \mathbf{x1})$ e $G(\mathbf{x2}, m(\mathbf{x2}))$ fossem erradamente unificados como acima, incluindo o par $\mathbf{x1}/m(\mathbf{x1})$, então estaríamos numa situação em que *alguém seria a sua própria mãe*.
- Com efeito, se os predicados se aplicassem a uma pessoa, por exemplo à Ana, estaríamos a unificar “a Ana gosta de si própria” com “a Ana gosta da sua mãe”, ou seja que as pessoas de quem a Ana gosta (a Ana e a sua Mãe) seriam as mesmas!

Unificação e Unificadores

Unificador:

Um unificador de dois (ou mais) termos é uma substituição que os permite unificar.

Aplicação de Substituição:

A aplicação de uma substituição σ a uma fórmula T , denotada por $T\sigma$, obtém-se por substituição de toda a variável v que ocorre em T pelo termo ω , sempre que o par v/ω seja um elemento de σ .

- Com estes conceitos podemos redefinir a noção de Unificação:

Unificação 2:

Dois **termos** são **unificáveis** se existir um seu unificador.

De notar que dois termos podem ter mais do que um unificador, como pode ser verificado nos exemplos seguintes.

Unificadores Mais Gerais

- Na unificação de dois termos T_1 e T_2 , são identificados diferentes unificadores σ e os correspondentes termos unificados $T = P_1\sigma = P_2\sigma$

$$P(x_1, y_1) \approx P(x_2, y_2)$$

- $\sigma_1 = \{ x_1/x_2, y_1/y_2 \}$ $T = P(x_2, y_2)$
- $\sigma_2 = \{ x_2/x_1, y_2/y_1 \}$ $T = P(x_1, y_1)$
- $\sigma_3 = \{ x_2/x_1, y_1/b, y_2/b \}$ $T = P(x_1, b)$
- $\sigma_4 = \{ x_1/a, x_2/a, y_1/y_2 \}$ $T = P(a, y_2)$
- $\sigma_5 = \{ x_1/a, x_2/a, y_1/b, y_2/b \}$ $T = P(a, b)$
- $\sigma_6 = \{ x_1/c, x_2/c, y_1/d, y_2/d \}$ $T = P(c, d)$

- Os dois primeiros unificadores σ_1 e σ_2 correspondem a variantes do mesmo unificador (troca de variáveis substituídas pelas variáveis a substituir).
- Mas o unificador σ_1 é **mais geral** que o unificador σ_3 pois os termos resultantes podem ser unificados com mais termos. Por exemplo, $P(a, c)$ é unificável com $P(x_1, x_2)$ mas não com $P(x_1, b)$. De facto, σ_1 (e σ_2) é um **unificador mais geral (mgu)** dos termos P_1 e P_2 .
- Em casos simples, o unificador mais geral pode ser obtido “intuitivamente” não instanciando variáveis mais do que o necessário. De uma forma sistemática, pode ser usado o seguinte algoritmo para se obter um unificador mais geral de dois termos.

Algoritmo Martelli-Montanari

Algoritmo Martelli-Montanari (MM): Seja U um conjunto de pares $s_i \approx t_i$ de termos.

Enquanto for possível, escolher arbitrariamente (não-deterministicamente) um par de termos de U de um dos tipos abaixo, e executar a correspondente acção :

1. $f(s_1, s_2, \dots, s_n) \approx f(t_1, t_2, \dots, t_n)$
→ substituir em U pelos pares $s_1 \approx t_1, s_2 \approx t_2, \dots, s_n \approx t_n$
2. $f(s_1, s_2, \dots, s_n) \approx g(t_1, t_2, \dots, t_n)$ em que $f \neq g$
→ **abortar** - os termos não são unificáveis (**conflito**)
3. $x \approx x$
→ apagar o par (**substituição vazia**)
4. $t \approx x$ em que t não é uma variável
→ substituir por $x \approx t$
5. $x \approx t$ em que x não ocorre em t (mas pode ocorrer noutros pares)
→ substituir por x / t e aplicar a substituição $\{ x / t \}$ em todos os pares de U
6. $x \approx t$ em que x ocorre em t e $x \neq t$
→ **abortar** - os termos não são unificáveis (**occurs-check**).

Se não abortar (nas regras 2 ou 6) o algoritmo termina, com sucesso, quando mais nenhum par puder ser escolhido, sendo o conjunto U final um unificador mais geral.

Algoritmo Martelli-Montanari

Exemplo 1: Unificar os predicados $P(x_1, y_1, f(x_1, y_1))$ e $P(x_2, g(x_2), z_2)$.

- i. $U = \{ P(x_1, y_1, f(x_1, y_1)) \approx P(x_2, g(x_2), z_2) \}$
- ii. $U = \{ x_1 \approx x_2, y_1 \approx g(x_2), f(x_1, y_1) \approx z_2 \}$ por **MM-1**
- iii. $U = \{ x_1/x_2, y_1 \approx g(x_2), f(x_2, y_1) \approx z_2 \}$ por **MM-5**
- iv. $U = \{ x_1/x_2, y_1/g(x_2), f(x_2, g(x_2)) \approx z_2 \}$ por **MM-5**
- v. $U = \{ x_1/x_2, y_1/g(x_2), z_2 \approx f(x_2, g(x_2)) \}$ por **MM-4**
- vi. $\theta = \{ x_1/x_2, y_1/g(x_2), z_2/f(x_2, g(x_2)) \}$ por **MM-5**

- Aplicando-se esta substituição a qualquer dos predicados obtém-se o termo unificado

$$P(x_2, g(x_2), f(x_2, g(x_2)))$$

- De notar que o unificador mais geral obtido pelo algoritmo MM é uma variante do unificador que tínhamos obtido atrás de uma forma “intuitiva”

$$\sigma = \{ x_2/x_1, y_1/g(x_1), z_2/f(x_1, g(x_1)) \}$$

- De facto a única diferença é que neste unificador σ , x_2 é substituído por x_1 e no unificador θ obtido pelo algoritmo MM é o contrário que se verifica.