Lógica de Predicados

Fernando Cardoso Durier da Silva
fernando.durier@uniriotec.br

Lógica Proposicional

- Modela sentenças
- Fácil compreensão
- Primeiro aprendizado na área de representação do conhecimento e raciocínio
- Não consegue modelar de forma simples objetos e suas propriedades
- Não consegue modelar quantidades
- Não consegue enumerar objetos de forma simples

Lógica de Predicados

- Modela objetos através de variáveis
- Modela relações entre objetos
- Modela funções
- Permite composição de atributos
- Permite derivação de relacionamentos de forma mais simples
- Consegue tratar problemas de quantificação através de quantificadores

Lógica de Predicados x Lógica Proposicional

- As linguagens lógicas tem o papel de representar conhecimento para um agente. Esse conhecimento tem um aspecto Ontológico e um Epistemológico.
  - **Aspecto Ontológico**: representação do ambiente onde o agente está inserido
  - **Aspecto Epistemológico**: crença/valoração/sentido que o agente encontra nos fatos modelados

Linguagem	Aspecto Ontológico	Aspecto Epistemológico
L. Lógica Proposicional	Fatos	Verdadeiro, Falso e Desconhecido
L. Lógica de Predicados	Fatos, Objetos e Relações	Verdadeiro, Falso e Desconhecido

Sintaxe da Lógica de Predicados

- Os elementos básicos da lógica de predicados são símbolos que se referem a objetos, funções e relações.
  - **Variáveis:** Se referem a objetos ou conjunto de objetos. Exemplo: x | x é humano
  - **Constantes:** Se referem a instâncias específicas de objetos. Exemplo: Bob é humano; b | b é humano; b=Bob;
  - **Predicados**: Se referem a relações entre objetos. Exemplo: b=Bob, m=Maria; Mae(x,y) = x é mãe de y; Mae(m,b) = Maria é mãe de Bob;
  - **Funções**: Se referem a funções do objeto. Exemplo: x | x é humano; x = Bob; Aluno(x) = x é aluno; Aluno(Bob) = Bob é aluno.
  - **Termos**: expressões lógicas que se referem a um objeto.

Lógica de Predicados

- Convenção segundo [4]:
  - **Variáveis** - letras do fim do alfabeto
  - **Constantes** - letras do início do alfabeto
  - **Funções** - letras do início para o fim do alfabeto
  - **Predicados** - letras do meio para o fim do alfabeto
  - **Termos** - letras perto do fim do alfabeto
  - **Fórmulas** - letras maiúsculas do meio para o fim do alfabeto
  - **Conjunto de Fórmulas** - letras maiúsculas em negrito do meio para o fim do alfabeto

Quantificadores

- Grande parte da expressividade da lógica de predicados é devida ao uso dos conectivos lógicos, que nos permitem formar sentenças complexas a partir de sentenças mais simples. Entretanto, o que realmente torna a lógica de predicados mais expressiva que a lógica proposicional é a noção de variáveis e quantificadores [1]
- $\forall$ - Para Todo/Qualquer um: Podemos estabelecer fatos para todos os objetos de um contexto, sem ter que enumerá-los um a um.
  - Denota conjunção: $\forall X$ azul(X) $\Leftrightarrow$ azul(X1) $\wedge$ azul(X2) $\wedge$ azul(X3) $\wedge$ ... $\wedge$ azul(Xn-1) $\wedge$ azul(Xn)
- $\exists$ - Existe/Algum: Podemos estabelecer a existência de um objeto sem termos que explicitamente identificá-lo.
  - Denota disjunção: $\exists X$ vermelho(x) $\Leftrightarrow$ vermelho(A) $\vee$ vermelho(B) $\vee$ vermelho(x)
- **Exemplos:**
  - Nenhum humano é imortal >>> $\forall x$ (Humano(x) $\rightarrow$ $\neg$Imortal(x))
  - Pelo menos um humano é inteligente >>> $\exists x$ (Humano(x) $\wedge$ Inteligente (x))

Quantificadores - Enunciados Categóricos

- **Universal afirmativo**: Todos os humanos são mortais.
  - $\forall X\ h(X) \rightarrow m(X)$ | x é humano; h(X):= X é humano; m(X):= X é mortal;
- **Universal negativo**: Nenhum humano é extra-terrestre.
  - $\forall X\ h(X) \rightarrow \neg e(X)$ | x é humano; h(X):= X é humano; e(X):= X é extra-terrestre;
- **Particular afirmativo**: Alguns humanos são cultos.
  - $\exists X\ h(X) \land c(X)$ | x é humano; h(X):= X é humano; c(X):= X é culto;
- **Particular negativo**: Alguns humanos não são cultos
  - $\exists X\ h(X) \land \neg c(X)$ | x é humano; h(X):= X é humano; c(X):= X é culto;

Equivalência entre Sentenças

- **Exemplo Sentenças**
  - Nem tudo que brilha é ouro.
  - Existe algo que brilha e não é ouro.

Equivalência entre Sentenças

- **Exemplo Sentenças**
  - Nem tudo que brilha é ouro.
  - Existe algo que brilha e não é ouro.
- **Fórmulas**
  - $\neg \forall X[b(X) \rightarrow o(X)]$
  - $\exists X[b(X) \land \neg o(X)]$

Equivalência entre Sentenças

- **Exemplo Sentenças**
  - Nem tudo que brilha é ouro.
  - Existe algo que brilha e não é ouro.
- **Fórmulas**
  - $\neg\forall X[b(X) \rightarrow o(X)]$
  - $\exists X[b(X) \land \neg o(X)]$
- **Equivalência**
  - $\neg\forall X[b(X) \rightarrow o(X)]$

Equivalência entre Sentenças

- **Exemplo Sentenças**
  - Nem tudo que brilha é ouro.
  - Existe algo que brilha e não é ouro.
- **Fórmulas**
  - $\neg \forall X[b(X) \rightarrow o(X)]$
  - $\exists X[b(X) \land \neg o(X)]$
- **Equivalência**
  - $\neg \forall X[b(X) \rightarrow o(X)]$
  - $\equiv \neg \forall X[\neg b(X) \lor o(X)]$

Equivalência entre Sentenças

- **Exemplo Sentenças**
  - Nem tudo que brilha é ouro.
  - Existe algo que brilha e não é ouro.
- **Fórmulas**
  - $\neg \forall X[b(X) \rightarrow o(X)]$
  - $\exists X[b(X) \land \neg o(X)]$
- **Equivalência**
  - $\neg \forall X[b(X) \rightarrow o(X)]$
  - $\equiv \neg \forall X[\neg b(X) \lor o(X)]$
  - $\equiv \exists X \neg [\neg b(X) \lor o(X)]$

Equivalência entre Sentenças

- **Exemplo Sentenças**
  - Nem tudo que brilha é ouro.
  - Existe algo que brilha e não é ouro.
- **Fórmulas**
  - $\neg \forall X\ b(X) \rightarrow o(X)$
  - $\exists X\ b(X) \land \neg o(X)$
- **Equivalência**
  - $\neg \forall X[b(X) \rightarrow o(X)]$
  - $\equiv \neg \forall X[\neg b(X) \lor o(X)]$
  - $\equiv \exists X \neg [\neg b(X) \lor o(X)]$
  - $\equiv \exists X\ [b(X) \land \neg o(X)]$

Exemplos

- “Toda cobra é venenosa”
- “Os remédios são perigosos”
- “Nenhuma bruxa é bela”
- “Não existe bêbado feliz”
- “Algumas pedras são preciosas”
- “Existem plantas que são carnívoras”
- “Alguns políticos não são honestos”
- “Há aves que não voam"

Exemplos

- “Toda cobra é venenosa”: $\forall X$ cobra(X) $\rightarrow$ venenosa(X)
- “Os remédios são perigosos”: $\forall X$ remedio(X) $\rightarrow$ perigoso(X)
- “Nenhuma bruxa é bela”: $\forall X$ bruxa(X) $\rightarrow \neg$bela(X)
- “Não existe bêbado feliz”: $\forall X$ bebado(X) $\rightarrow \neg$feliz(X)
- “Algumas pedras são preciosas”: $\exists X$ pedra(X) $\wedge$ preciosa(X)
- “Existem plantas que são carnívoras”: $\exists X$ planta(X) $\wedge$ carnivora(X)
- “Alguns políticos não são honestos”: $\exists X$ politico(X) $\wedge \neg$honesto(X)
- “Há aves que não voam”: $\exists X$ ave(X) $\wedge \neg$voa(X)

Quantificadores - Escopo de Variáveis

- Na lógica de predicados as variáveis são regidas pela presença ou não de quantificadores, podendo ser classificadas em:
  - **Livres (free)** - Variáveis que não são regidas por um quantificador.
    - Pai(P, B) $\land$ Mae(M, B) $\rightarrow$ $\neg$Orfão(B)
  - **Ligadas (bounded)** - Variáveis que são regidas por um quantificador.
    - $\forall X \exists Y \; P(X) \rightarrow Q(Y)$

Quantificadores - Escopo de Sentenças

- Assim como as variáveis, as sentenças podem ser classificadas como:
  - **Abertas** - Sentenças que todas as variáveis estão livres
    - Irmão (x, y) → MesmaMãe (x, y)
  - **Fechadas** - Sentenças que todas as variáveis estão ligadas
    - ∃ x Irmão(Caio, x)
  - **Primitivas** - Sentenças que não possuem variáveis livres nem variáveis ligadas
    - Irmão(Caio, João) → MesmaMãe(Caio, João)

Inferência

- A inferência em lógica de predicados é semelhante à vista em lógica proposicional, porém mais complexa devido à:
  - Quantificadores
  - Variáveis
- Para fazermos inferência nesse tipo de lógica precisamos de alguns conceitos:
  - Substituição
  - Forma Normal Prenex
  - Skolemização
  - Unificação

Substituição

- Uma substituição α é um conjunto finito
  - {v1/t1, v2/t2, v3/t3, ..., vn/tn}
  - onde:
    - vi é uma variável
    - ti é um termo distinto de vi
    - as variáveis v1 , ..., vn são distintas.
  - cada elemento vi/ti é chamado de ligação
- α pode ser classificado como:
  - **Substituição Base**: se cada ti é um termo que não contém variáveis
  - **Substituição Pura de Variáveis**: se cada ti é uma outra variável

Substituição

- **Exemplos:**
  - α1 = {x1/Bob, x2/Maria, x3/João}
    - Substituição Base
  - α2 = {x/a, y/b, z/c}
    - Substituição Pura de Variáveis
  - α3 = {}
    - Substituição vazia

Substituição - Instância

- Dado uma substituição α e uma expressão E. A instância E por α é dada por Eα.
- A expressão resultante é a expressão E substituindo cada variável vi em α pelo termo ti correspondente.
- **Exemplo:**
  - E = p(x,y,z)
  - α = {x/3,y/5,z/8}
  - Eα = p(3,5,8)
  - F = w(x) → m(y)
  - β = {x/Will, y/Maria}
  - Fβ = w(Will) → m(Maria)

Substituição - Composição

- Dado uma substituição α e outra β denotadas por:
  - α = {u1/s1, u2/s2, … , un/tn}
  - β = {v1/t1, v2/t2, … , vm/tm}
- A composição entre α e β, denotada por αβ, é definida como:
  - {u1/s1β, u2/s2β, … , un/tnβ, v1/t1, v2/t2, … , vm/tm}
  - onde u1 = s1β
  - vj/tj para as quais vj ∈ {u1 ,...,um }
- A composição pode ser entendida como uma sequência de substituições de ui para termos si, que são variáveis, e essas variáveis vi, que entre elas tem termos si, para termos ti.

Substituição - Composição

- Exemplo:
  - α={x/y, z/w }
  - β={y/João, w/Maria}.
  - $\alpha\beta$ = {x/yβ, z/wβ, y/João, w/Maria}
  - $\alpha\beta$ = {x/João, z/Maria, y/João, w/Maria}

Substituição - Substituição mais Geral

- Dado uma substituição α e β
- Podemos dizer que α é mais geral do que β, se existir uma substituição ɣ que:
  - $\beta = \alpha\gamma$
- Exemplo:
  - $\alpha = \{x/f(g(x,y)), y/g(z,b)\}$
  - $\beta = \{x/f(g(a,h(z))), y/g(h(x),b), z/h(x)\}$
  - $\gamma = \{x/a, y/h(z), z/h(x)\}$
  - $\alpha\gamma = \{x/f(g(a,h(z))), y/g(h(x),b), z/h(x)\} = \beta$
- Exemplo:
  - $\alpha = \{x/f(i), y/g(j)\}$
  - $\beta = \{x/f(h(x)), y/g(l(y)), k/m(z)\}$
  - $\gamma = \{i/h(x), j/l(y), k/m(z)\}$
  - $\alpha\gamma = \{x/f(h(x)), y/g(l(y)), k/m(z)\} = \beta$

Forma Normal Prenex (FNP)

- Lógica Proposicional apresenta duas formas normais:
  - Disjuntiva
  - Conjuntiva
- Lógica de Predicados apresenta mais uma forma normal:
  - Prenex
- A razão para utilizar uma forma normal prenexa de uma fórmula usualmente é para simplificar métodos de prova, assim como as próprias fórmulas em si.

Forma Normal Prenex (FNP)

- Uma fórmula é tida como estando em sua FNP se, e só se, cada variável desta fórmula se encontra sob o escopo de um quantificador, e se todos os quantificadores estão juntos precedendo uma sentença livre de quantificadores.
- Denotada por:
  - $Q_1x_1, Q_2x_2, \ldots, Q_nx_n \circ \alpha$
    - $\circ$:= operação de produto cartesiano (dot product)
    - α := fórmula livre de quantificadores
    - $Q_1x_1, Q_2x_2, \ldots, Q_nx_n$:= prefixo da fórmula
    - $x_1, x_2, \ldots, x_n$:= variáveis distintas
    - Q_i := São quantificadores tanto universais ($\forall$) quanto existenciais ($\exists$), para cada variável distinta presente em α variando de 1 à n.

Forma Normal Prenex (FNP)

- Regras de Manipulação dos Quantificadores
  - **Conjunção**
    - $(\forall x\, f(x)) \land g(x) \equiv \forall x\, f(x) \land g(x)$
    - $(\exists x\, f(x)) \land g(x) \equiv \exists x\, f(x) \land g(x)$
  - **Disjunção**
    - $(\forall x\, f(x)) \lor g(x) \equiv \forall x\, f(x) \lor g(x)$
    - $(\exists x\, f(x)) \lor g(x) \equiv \exists x\, f(x) \lor g(x)$
  - **Negação**
    - $\neg \forall x\, f(x) \equiv \exists x\, \neg f(x)$
    - $\neg \exists x\, f(x) \equiv \forall x\, \neg f(x)$

- as regras da conjunção e disjunção requerem atenção ao fato de a variável livre não ter aparecido antes quantificada

Forma Normal Prenex (FNP)

- Regras de Manipulação dos Quantificadores
    - **Implicação**
        - $(\forall x\, f(x)) \rightarrow g(x)$
            - $\equiv \neg(\forall x\, f(x)) \vee g(x)$
            - $\equiv \exists x\, \neg f(x) \vee g(x)$
            - $\equiv \exists x\, f(x) \rightarrow g(x)$
        - $(\exists x\, f(x)) \rightarrow g(x)$
            - $\equiv \neg(\exists x\, f(x)) \vee g(x)$
            - $\equiv \forall x\, \neg f(x) \vee g(x)$
            - $\equiv \forall x\, f(x) \rightarrow g(x)$

Forma Normal Prenex (FNP)

- Exemplo:
  - Fórmula que não está na FNP
    - $\forall x\ ((\exists y\ f(y)) \vee ((\exists z\ g(z)) \rightarrow p(x)))$

Forma Normal Prenex (FNP)

- Exemplo:
  - Fórmula que não está na FNP
    - $\forall x\, ((\exists y\, f(y)) \lor ((\exists z\, g(z)) \rightarrow p(x)))$
  - Agora na FNP
    - $\forall x\, \exists y\, \forall z\, (f(y) \lor (g(z) \rightarrow p(x)))$

Skolemização

- Uma fórmula lógica está na forma normal de Skolem se sua forma normal prenex contiver somente quantificadores universais.
- Cada fórmula de primeira ordem pode ser convertida na forma normal de Skolem por meio do processo de skolemização.
- A fórmula resultante deste processo não é necessariamente equivalente à original, mas é satisfatível se e somente se a original também o for.

Skolemização

- Substituir cada variável y, quantificada existencialmente por um termo
  - f(x1, ..., xn)
  - f é uma nova função (não existe ocorrência de f na fórmula).
- Se a fórmula estiver em FNP, {x1, ..., xn} são variáveis universalmente quantificadas cujos quantificadores precedem a variável y.
- A função f nesse processo é dita função de Skolem e o termo é dito termo de Skolem.
- No caso de ocorrência de uma variável y quantificada existencialmente, onde essa quantificação não é precedida por um quantificador universal, então a variável y é substituída por uma constante.

Normalização de Fórmulas

1. Eliminar Variáveis Livres
2. Eliminar Quantificadores Redundantes
3. Mover Negação para o Interior da Fórmula
4. Eliminar Quantificadores Existenciais (Skolemização)
5. Renomear Variáveis Quantificadas mais de Uma Vez
6. Remover Biimplicações (Equivalências) e Implicações
7. Forma Normal Prenex e Remover Quantificadores Universais
8. Colocar a composição de FNP na forma conjuntiva

Normalização de Fórmulas

- Eliminar Variáveis Livres
- Se P tiver uma variável livre, x, substituir P por $\exists x(P(x))$. Isto deve ser repetido até que a fórmula não contenha mais variáveis livres.

Normalização de Fórmulas

- Eliminar Quantificadores Redundantes
- Eliminar todo quantificador $\forall x$ ou $\exists x$ que não contenha nenhuma ocorrência livre de x no seu escopo isto é, eliminar todo quantificador 'desnecessário'.

Normalização de Fórmulas

- Mover Negação para o Interior da Fórmula
- Os sinais de negação são movidos de fora dos parênteses para a frente dos átomos, usando as generalizações de De Morgan e o fato de $\neg(\neg P) \rightarrow P$.

Normalização de Fórmulas

- Eliminar Quantificadores Existenciais (Skolemização)
- Aplicação de Skolemização
- $\forall y\ (\exists x\ P(x,y)) \equiv \exists f(y)\ \forall y\ P(f(y),y)$
- $\exists x\ \forall y\ (P(x,y)) \equiv \forall y\ P(cte,y)$

Normalização de Fórmulas

- Renomear Variáveis Quantificadas mais de Uma Vez
- Se uma mesma variável é governada por dois quantificadores, substituir a variável de um deles e todas as suas ocorrências livres no escopo do quantificador, por uma nova variável que não ocorra na fórmula. Esse passo deve ser repetido até que todos os quantificadores governem variáveis diferentes.
- Assim, ao invés de:
    - $\forall y\, p(y) \rightarrow \forall y\, m(y) \equiv \forall x\, p(x) \rightarrow \forall y\, m(y)$

Normalização de Fórmulas

- Remover Biimplicações (Equivalências) e Implicações
- $F \leftrightarrow G$ por $(\neg F \lor G) \land (F \lor \neg G)$
- $F \rightarrow G$ por $(\neg F \lor G)$

Normalização de Fórmulas

- Forma Normal Prenex e Remover Quantificadores Universais
- Neste ponto não existem mais quantificadores existenciais e cada quantificador universal tem sua própria variável. Pode-se pois mover todos os quantificadores universais para a frente da fórmula bem formada (fbf) e deixar que o escopo de cada um deles inclua a fbf inteira.
- Esse procedimento deixará a fórmula da FNP.

Normalização de Fórmulas

- Colocar a matriz da FNP na forma conjuntiva
- Qualquer matriz pode ser escrita como uma conjunção de um conjunto finito de disjunções, forma conhecida como forma normal conjuntiva.
- $(M(x) \land N(x)) \lor P(x) \equiv (M(x) \lor P(x)) \land (N(x) \lor P(x))$
- $M(x) \lor (N(x) \land P(x)) \equiv (M(x) \lor N(x)) \land (M(x) \lor P(x))$

Normalização de Fórmulas

- Exemplo:
- $\forall x(\exists z(\neg p(x,z)) \lor \exists y(p(y,x)))$
- passo1: Prenex
    - A fórmula já se encontra em prenex, não há variáveis livres
- passo2: Skolemização
    - $\forall x \; \exists z \; \exists y \; ((\neg p(x,z)) \lor (p(y,x)))$
    - $\exists f(x) \; \exists g(x) \; \forall x \; ((\neg p(x, f(x))) \lor (p(g(x), x)))$
    - $\forall x \; ((\neg p(x, f(x))) \lor (p(g(x), x)))$
- passo3: Remover Quantificador Universal
    - $\neg p(x, f(x)) \lor p(g(x), x)$
- passo4: FNC
    - Fórmula já em FNC
- passo5: Conjunto de Cláusulas
    - $\{ \neg p(x, f(x)) \lor p(g(x), x) \}$

Unificação

- Sejam duas expressões E e F e uma substituição α.
- Se ocorrer que $E\alpha \equiv F\alpha$ então se diz que a substituição α é um unificador de E e F.
- Também, neste caso, pode-se dizer que E e F são unificáveis.

Unificação

- Exemplo :
  - $E \equiv q(x,y,a)$
  - $F \equiv q(z,w,a)$
  - $\alpha = \{x/d, y/e, z/d, w/e\}$.
  - Aplicando a substituição em E e F, temos:
    - $E\alpha \equiv q(d,e,a)$
    - $F\alpha \equiv q(d,e,a)$
    - Como $E\alpha \equiv F\alpha$, então α é um unificador de E e F.

Unificação - Unificador Mais Geral (ufmg)

- Sejam duas expressões E e F e um unificador α de E e F.
- Diz-se que α é um unificador mais geral (umg) de E e F se for mais geral do que qualquer outro unificador de E e F.
- Para se encontrar um umg (caso exista) para duas expressões E e F deve-se. Indicar uma substituição α tal que Eα ≡ Fα (unificador);
- Provar que para qualquer outra substituição β, α é mais geral que β (umg)

Algoritmo de Unificação

- Entrada
  - par de fórmulas atômicas (A,B)
- Saída
  - “não”, caso não seja possível obter um umg para A e B;
  - “sim”, caso seja possível obter um umg para A e B, fornecendo neste caso o umg encontrado para A e B.

Algoritmo de Unificação

- Passo 1:
  - Se o símbolo de predicado de A for diferente do símbolo de predicado de B então emitir "não" como saída e parar
  - Senão criar uma coleção P constituída por pares de termos (t_i, s_i), $i=\{1,2,\dots,n\}$, obtidos de $A \equiv p(t_1, \dots, t_n)$ e $B \equiv p(s_1, \dots, s_n)$.
  - Obs.: uma vez que o símbolo de predicado de A e B são o mesmo, suas aridades são iguais

Algoritmo de Unificação

- Passo 2:
  - Escolher aleatoriamente um par (C,D) de P (se P estiver vazio, ir para o Passo 9)
- Passo 3:
  - Se C ≡ f(s1 ,...,sn) e D ≡ f(t1 ,...,tn) Então remover o par (C,D) de P e incluir os pares (s1 ,t1),...,(sn ,tn) em P e voltar ao Passo 2
  - Senão vá para o próximo passo
- Passo 4:
  - Se C ≡ f(s1 ,...,sn) e D ≡ g(t1 ,...,tn) Então emitir “não” como saída e parar
  - Senão vá para o próximo passo.

Algoritmo de Unificação

- Passo 5:
  - Se C ≡ D ≡ x (x é uma variável) ou C ≡ D ≡ c (c é uma constante) Então remover o par (C,D) de P e voltar ao Passo 2
- Passo 6:
  - Se C ≡ c (constante) e D ≡ d (constante) Então emitir “não” como saída e parar
  - Senão vá para o próximo passo.
- Passo 7:
  - Se C ≡ t (t é um termo que não é variável) e D ≡ x (x é uma variável) Então remover o par (C,D) de P e incluir o par (D,C) em P e voltar ao Passo 2
  - Senão vá para o próximo passo.

Algoritmo de Unificação

- Passo 8:
    - Se C ≡ x (x é uma variável) e D ≡ t (t é um termo) e x ≠ t e x ocorre em algum outro par de P
    - Então
        - Se x ocorrer em t
        - Então emitir “não” e parar
        - Senão substituir cada x pelo termo t em todos os outros pares de P e voltar ao Passo 2
    - Senão vá para o próximo passo.

Algoritmo de Unificação

- Passo 9:
    - Se nenhum passo anterior puder ser executado
    - Então emitir “sim”, retornar o umg de A e B dado pela substituição composta pelos pares (C,D) pertencentes ao conjunto P e parar com sucesso.
    - Senão voltar ao Passo 2.

Unificação

- Exemplo:
  - A = A: Unificação bem sucedida (tautologia)
  - *A = B, B = abc* : *A* e *B* são unificados com o átomo *abc*
  - *xyz = C, C = D* : A unificação é simétrica
  - *abc = abc* : A unificação é bem sucedida
  - *abc = xyz* : Falha em unificar porque os átomos são diferentes
  - *f*(*A*) = *f*(*B*) : A é unificado com B
  - *f*(*A*) = *g*(*B*) : Falha porque as cabeças dos termos são diferentes
  - *f*(*A*) = *f*(*B*, *C*) : A unificação falha porque os termos têm aridades diferentes
  - *f*(*g*(*A*)) = *f*(*B*) : Unifica *B* com o termo *g*(*A*)
  - *f*(*g*(*A*), *A*) = *f*(*B*, *xyz*) : Unifica *A* com o átomo *xyz* e *B* com o termo *g*(*xyz*)
  - *A* = *f*(*A*) : Unificação infinita, *A* é unificado com *f*(*f*(*f*(*f*(...)))). Na Lógica de Primeira Ordem propriamente dita e em vários dialetos modernos de Prolog, isto é proibido (ou reforçado pelo Occurs check)

Unificação

- Aplicações da Unificação
  - Tableaux para lógica de predicados
    - Agilizando as provas automáticas
    - instaurando políticas mais justas de operações no tableaux
  - Prolog
    - Funcionamento básico e principal
    - Construção de Conhecimento
    - Recuperação de Conhecimento
  - Haskell
    - Similar ao Prolog

Tableaux em Lógica de Predicados

- O Tableaux em Lógica de Predicados funciona da mesma forma como funcionava para lógica proposicional, porém, demanda atenção para alguns detalhes como:
  - Quantificadores
  - Unificação

Tableaux em Lógica de Predicados

- Exemplo:
  - ∀x ¬P(x,a) ∧ P(a,b) ⊢ ¬(∀x∀y ¬P(x,y) ∨ P(y,x))
  - {(∀x ¬P(x,a)), (P(a,b)), (∀x∀y ¬P(x,y) ∨ P(y,x))}

1. ∀x ¬P(x,a)
2. P(a,b)
3. ∀x∀y ¬P(x,y) ∨ P(y,x)
4. ¬P(b,a) — 1, particularização universal
5. ∀y ¬P(a,y) ∨ P(y,a) — 3, particularização universal
6. ¬P(a,b) ∨ P(b,a) — 5, particularização universal
7. ¬P(a,b) | P(b,a) — 6, remoção da disjunção

Tableaux em Lógica de Predicados

- Dado: {∀x P(x), ∃x (¬P(x) ∨ ¬P(f(x)))}
- Temos o seguinte Tableaux

1.	∀x P(x)		h1
2.	∃x (¬P(x) ∨ ¬P(f(x)))		h2
3.	¬P(c) ∨ ¬P(f(c))		2(∃)
4.	P(c)		1(∀)
5.	¬P(c)	¬P(f(c))	3(∨)
6.		P(f(c))	1(∀)

Tableaux em Lógica de Predicados

- Agora com unificação, trazemos uma política de "justiça" à resolução
- Utilizamos uma substituição genérica que depois é resubstituída a partir de estratégia de unificação para achar o conjunto de substituições para o x' de modo à dar clash

1.	$\forall x\, P(x)$		h1
2.	$\exists x\, (\neg P(x) \vee \neg P(f(x)))$		h2
3.	$P(x')$	x'=f(c)	$1(\forall)$
4.	$\neg P(c) \vee \neg P(f(c))$		$2(\exists)$
5.	$\neg P(c)$	$\neg P(f(c))$	$3(\vee)$
6.	x''=c $P(x'')$		$1(\forall)$

Referências:

1. do Lago Pereira, S. Lógica de Predicados.
2. Luger, G. F. (2005). *Artificial intelligence: structures and strategies for complex problem solving*. Pearson education.
3. Brandão, A. A. F., Costa, A. H. R., & Sichman, J. S. PCS 2428/PCS 2059-Inteligência Artificial.
4. em Lógica, P. a Linguagem PROLOG; Marco A. *Casanova, Fernando AC Giorno, Antônio L. Furtado*.
5. http://www.ppgsc.ufrn.br/~rogerio/material_auxiliar/FMC20121T2_repres_clausal_form.pdf
6. ABE, Jair Minoro; SCALZITTI, Alexandre; SILVA FILHO, João Inácio da. Introdução à lógica para a ciência da computação. São Paulo: Arte & Ciência, 2001. 247 p. ISBN 85-7473-045-9