UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO
http://www.icmc.usp.br
Departamento de Ciências de Computação

SCC-630 - Capítulo 2
Lógica de Predicados

João Luís Garcia Rosa[1]

[1]Departamento de Ciências de Computação
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - São Carlos
http://www.icmc.usp.br/~joaoluis

2011

Sumário

Sumário

Proposições Lógicas

- Para representar o conhecimento do mundo que um sistema de IA necessita, explora-se o uso da lógica proposicional.
- Vai-se representar os fatos do mundo real através das fórmulas bem formadas ou proposições lógicas, como mostrado abaixo:
  - Está chovendo.
    - $chovendo$
  - Está ensolarado.
    - $ensolarado$
  - Se está chovendo, então não está ensolarado.
    - $chovendo \rightarrow \neg ensolarado$
- Lógica das proposições, ou Lógica Proposicional

Sumário

Sintaxe das Linguagens Proposicionais

Um alfabeto proposicional α consiste de

- símbolos lógicos:
  - pontuação: (,)
  - conectivos:
    - $\neg$ (negação)
    - $\wedge$ (conjunção)
    - $\vee$ (disjunção)
    - $\rightarrow$ (implicação)
    - $\leftrightarrow$ ou $\equiv$ (bi-implicação ou equivalência)
- símbolos não-lógicos: um conjunto finito **P** de símbolos proposicionais diferentes dos símbolos lógicos. Ex. *p*, *q*, etc.

Sintaxe das Linguagens Proposicionais

- O conjunto de fórmulas proposicionais é o menor conjunto de cadeias satisfazendo às seguintes condições:
  - todo símbolo proposicional é uma fórmula;
  - se p e q são fórmulas, então $(\neg p)$, $(p \wedge q)$, $(p \vee q)$, $(p \rightarrow q)$ e $(p \equiv q)$ também são fórmulas.
  - Uma fórmula q é uma subfórmula de uma fórmula p se, sozinha, continua a ser uma fórmula.
  - A linguagem proposicional, denotada por $L(\alpha)$, é o conjunto das fórmulas proposicionais.

Sumário

Semântica das Linguagens Proposicionais

- As fórmulas de uma linguagem proposicional, (que inclui os símbolos proposicionais), terão como significado os valores-verdade *FALSO* ou *VERDADEIRO*, abreviados F e V, respectivamente.
- Seja **P** o conjunto de símbolos proposicionais de α. Uma atribuição de valores-verdade para α é uma função $a : \mathbf{P} \Rightarrow \{F, V\}$.

Semântica das Linguagens Proposicionais

- Seja a uma atribuição de valores-verdade. A função de avaliação para $L(\alpha)$ induzida por a é a função $v : L(\alpha) \Rightarrow \{F, V\}$ definida da seguinte forma:
    - $v(p) = a(p)$, se p é um símbolo proposicional
    - $v(\neg p) = V$, se $v(p) = F$
      $= F$, se $v(p) = V$
    - $v(p \wedge q) = V$, se $v(p) = v(q) = V$
      $= F$, em caso contrário
    - $v(p \vee q) = F$, se $v(p) = v(q) = F$
      $= V$, em caso contrário
    - $v(p \rightarrow q) = F$, se $v(p) = V$ e $v(q) = F$
      $= V$, em caso contrário
    - $v(p \equiv q) = V$, se $v(p) = v(q)$
      $= F$, em caso contrário

Semântica das Linguagens Proposicionais

- Sejam **P** e **Q** conjuntos de fórmulas em $L(\alpha)$ e r uma fórmula em $L(\alpha)$.
  - r é *verdadeira* em uma atribuição de valores-verdade a se e somente se $v(r) = V$. Em caso contrário, r é *falsa*.
  - r é uma *tautologia* se e somente se, para toda atribuição de valores-verdade a, $v(r) = V$.
  - uma atribuição de valores-verdade a *satisfaz* a **P**, ou a é um *modelo* para **P**, se e somente se, para toda fórmula s em **P**, $v(s) = V$.
  - **P** é *satisfazível* se e somente se existe uma atribuição de valores-verdade a que satisfaz **P**. Em caso contrário, **P** é *insatisfazível*.
  - r é uma *conseqüência lógica* de **P**, ou **P** *implica logicamente* r (notação: $P \models r$), se e somente se, para toda atribuição de valores-verdade a, se a satisfaz **P** então a satisfaz r.

O Método da Tabela-Verdade

$\mathbf{P} = \{p \rightarrow \neg q,\ q \wedge p,\ q\}$
$\mathbf{Q} = \{p \vee q,\ q \rightarrow p\}$
$r = \neg q \rightarrow p$

Table: O Método da Tabela-Verdade.

p	q	$\neg q$	$p \rightarrow \neg q$	$q \wedge p$	q	$p \vee q$	$q \rightarrow p$	$\neg q \rightarrow p$
F	F	V	V	F	F	F	V	F
F	V	F	V	F	V	V	F	V
V	F	V	V	F	F	V	V	V
V	V	F	F	V	V	V	V	V

Sumário

Lógica de Primeira Ordem

A Lógica de primeira ordem, ou Cálculo de Predicados de Primeira Ordem (CPPO) pode ser caracterizada como um sistema formal apropriado a definição de teorias do universo de discurso da Matemática. A motivação para se estudar esta lógica é que a lógica sentencial não dá conta da representação de frases do tipo:

- Sócrates é homem.
- Platão é homem.
- Todos os homens são mortais.

Sumário

Alfabeto de Primeira Ordem

Um alfabeto de primeira ordem α consiste de:

- símbolos lógicos:
  - pontuação: (,)
  - conectivos: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\equiv$
  - quantificadores:

    $\forall$ (quantificador universal)

    $\exists$ (quantificador existencial)
  - variáveis: um conjunto de símbolos distintos dos demais, por convenção, representadas por letras maiúsculas: X, Y, Z, etc.
  - símbolo de igualdade (opcional): $=$
- símbolos não-lógicos:
  - constantes
  - símbolos funcionais n-ários ($n > 0$)
  - símbolos predicativos n-ários ($n > 0$)

Termo de Primeira Ordem

O conjunto de *termos de primeira ordem* é o menor conjunto satisfazendo às seguintes condições:

- toda variável é um termo;
- toda constante é um termo;
- se $t_1, ..., t_n$ são termos e f é um símbolo funcional n-ário, então $f(t_1, ..., t_n)$ também é um termo.

Sumário

Fórmula de Primeira Ordem

O conjunto de *fórmulas* é o menor conjunto satisfazendo às seguintes condições:

- se t_1,..., t_n são termos e p é um símbolo predicativo n-ário, então $p(t_1, ..., t_n)$ é uma fórmula, chamada de *fórmula atômica*.
- se t_1,..., t_n são termos e "$=$" é um símbolo de α então $(t_1 = t_2)$ é uma fórmula, também chamada de fórmula atômica.
- se p e q são fórmulas, então $(\neg p)$, $(p \wedge q)$, $(p \vee q)$, $(p \rightarrow q)$ e $(p \equiv q)$ também são fórmulas.
- se p é uma fórmula e X é uma variável, então $\forall X(p)$ e $\exists X(p)$ também são fórmulas.

Funções e Predicados Computáveis

- Predicados comuns: $pai(jose, maria)$
- Predicados computáveis "*maior_que*" e "*menor_que*":
- Funções computáveis: $maior\_que(mais(2,3),1)$

Linguagem de Primeira Ordem

- A *linguagem de primeira ordem*, denotada por $L(\alpha)$, é o conjunto de termos e fórmulas de primeira ordem.
- Em uma fórmula da forma $\forall X(q)$ (ou da forma $\exists X(q)$), q é o *escopo* de $\forall X$ (ou de $\exists X$).
- Uma ocorrência de uma variável X em uma fórmula p é *ligada* em p, se a ocorrência se dá em uma subfórmula de p da forma $\forall X(q)$ ou da forma $\exists X(q)$. Caso contrário, a ocorrência de X é *livre*.
- Uma variável X é *livre* em p se existe uma ocorrência livre de X em p.
- Uma fórmula p é uma *sentença* se e somente se nenhuma variável ocorre livre em p.

Forma Normal Conjuntiva

- Dada uma fórmula p, com variáveis livres $X_1,\ldots, X_n$, o *fecho universal* de p é a fórmula $\forall X_1 \ldots \forall X_n(p)$ e o *fecho existencial* de p é a fórmula $\exists X_1 \ldots \exists X_n(p)$.
- Uma fórmula p está na *forma normal prenex* se e somente se p for da forma $q(M)$ onde q, o prefixo de p, é uma cadeia de quantificadores e M, a matriz de p, é uma fórmula sem ocorrências de quantificadores.
- Uma fórmula p é uma *conjunção* se e somente se, omitindo-se os parênteses, for da forma $p_1 \wedge \ldots \wedge p_n$
- Uma fórmula p é uma *disjunção* se e somente se, omitindo-se os parênteses, for da forma $p_1 \vee \ldots \vee p_n$
- Uma fórmula p está na *forma normal conjuntiva* se e somente se estiver na forma normal prenex e a sua matriz for uma conjunção de disjunções de fórmulas atômicas, negadas ou não.

Modus Ponens

Regra de Inferência (*Modus Ponens*)

a partir de p e de $(p \rightarrow q)$, deduza q.

Sumário

Literal

- Um *literal positivo* é uma fórmula atômica.
- Um *literal negativo* é a negação de uma fórmula atômica.
- Um *literal* é ou um literal positivo ou um literal negativo.
- Dois literais têm *sinais opostos* se e somente se um deles for positivo e o outro for negativo.
- Dois literais são *complementares* se e somente se um deles for a negação do outro.
- Uma fórmula atômica f é o *átomo* de um literal l, denotado por $|l|$, se e somente se l for f ou $\neg f$.

Cláusula

- Uma *cláusula* é ou uma seqüência não vazia de literais ou a cláusula vazia, denotada por $\square$.
- A *linguagem de cláusulas* é o conjunto de todas as cláusulas.
- Uma *interpretação* I satisfaz uma cláusula não vazia c (denotado por $I \models c$) se e somente se I satisfaz a sentença f definida como

$$\forall X_1 ... \forall X_m (l_1 \vee ... \vee l_n)$$

  onde $X_1, ..., X_m$ são as variáveis ocorrendo em c e $l_1, ..., l_n$ são os literais de c. Diz-se ainda que c e f são *equivalentes*. Por convenção, a cláusula vazia é sempre insatisfazível.

Representação Clausal

- Um conjunto de cláusulas **S** é uma *representação clausal* para uma fórmula *p* se e somente se *p* é satisfazível se e somente se **S** é satisfazível.
- A obtenção da representação clausal de uma fórmula é um processo mecânico, como descrito a seguir.
  - *entrada*: uma fórmula *p*
  - *saída*: uma representação clausal **S** para *p*

Algoritmo de Representação Clausal

1. Tome o fecho existencial de p
2. Elimine quantificadores redundantes
3. Renomeie variáveis quantificadas mais de uma vez.
4. Elimine os conectivos "$\rightarrow$" e "$\equiv$"
5. Mova "$\neg$" para o interior da fórmula
6. Mova os quantificadores para o interior da fórmula. Objetivo: diminuir os escopos dos quantificadores,
7. Elimine os quantificadores existenciais
8. Obtenha a forma normal prenex
9. Obtenha a forma normal conjuntiva
10. Obtenha a representação clausal

Sumário

Substituição

- Um par (X, t) é uma *substituição simples* (lê-se "X substituído por t") se e somente se X é uma variável e t é um termo.
- Um conjunto finito β de substituições simples é uma *substituição* se e somente se duas substituições simples em β não coincidem no primeiro elemento.
- Uma substituição β é uma *substituição básica* se e somente se, para todo (X, t) em β, t é um termo sem ocorrências de variáveis.
- β é a *substituição vazia* ϵ se e somente se β for o conjunto vazio.
- Uma substituição β é uma *renomeação de variáveis* se e somente se cada par (X, t) em β for tal que t é uma variável e não existirem dois pares (X, U) e (Y, V) em β tais que $X \neq Y$ e $U = V$.

Composição de Substituições

- Seja *c* uma cláusula e β uma substituição. A *instanciação* de *c* por β, denotada por $c\beta$, é a cláusula obtida instanciando-se *c* por β e eliminando-se as ocorrências repetidas do mesmo literal, exceto a ocorrência mais à esquerda.
- A *composição de substituições* é a função, denotada por "$\circ$", que mapeia pares de substituições em uma substituição e é definida da seguinte forma:
  - Para todo par de substituições
  $$\beta = \{X_1/T_1,\ldots,\ X_n/T_n,\ Y_1/S_1,\ldots,\ Y_k/S_k\}$$
  $$\theta = \{Y_1/R_1,\ldots,Y_k/R_k,\ Z_1/Q_1,\ldots,\ Z_m/Q_m\}$$
  onde $X_1, \ldots, X_n, Y_1, \ldots, Y_k, Z_1, \ldots, Z_m$ são variáveis distintas, a composição de β com θ será a substituição:
  $$\beta \circ \theta =$$
  $$\{X_1/(T_1)\theta, \ldots, X_n/(T_n)\theta, Y_1/(S_1)\theta, \ldots, Y_k/(S_k)\theta, Z_1/Q_1, \ldots, Z_m/Q_m\}$$

Propriedades da Composição de Substituições

- Exemplo: Seja $\beta = \{X/f(Y), Y/Z\}$ e $\theta = \{X/a, Y/b, Z/Y\}$. Então a composição das substituições β com θ é a substituição
$$\beta \circ \theta = \{X/f(b), Y/Y, Z/Y\}$$
que é obtida, aplicando-se θ aos termos das substituições simples em β, formando o conjunto $X/f(b)$, Y/Y e acrescentando a única substituição simples de θ, "Z/Y", cujo primeiro elemento não coincide com o primeiro elemento de uma substituição simples em β.
- Propriedades:
  - $\theta \circ \varepsilon = \varepsilon \circ \theta = \theta$
  - $(E\theta)\beta$ = $E(\theta \circ \beta)$, qualquer que seja a expressão E
  - Se $E\theta = E\beta$ então $\theta = \beta$, qualquer que seja a expressão E
  - $(\theta \circ \beta) \circ \varphi = \theta \circ (\beta \circ \varphi)$

Unificador Mais Geral

- Seja $c = l_1 l_2 \ldots$ uma cláusula com conjunto de literais **L** = $\{l_1, l_2\}$ e β uma substituição.
  - β é um *unificador* para **L** se e somente se $(l_1)\beta = (l_2)\beta$.
  - β é um *unificador mais geral* (ou, abreviadamente, um u.m.g.) de **L** se e somente se β é um unificador de **L** e, para todo unificador θ de **L**, existe uma substituição φ tal que $\theta = \beta \circ \varphi$.
  - o conjunto **L** é *unificável* se e somente se existe um unificador para **L**.

Conjunto de Discórdia

- Um conjunto de termos **D** é o *conjunto de discórdia* de um conjunto de literais **L** = $\{l_1, ..., l_n\}$ se e somente se
  - **D** = $\emptyset$, se $n = 1$;
  - **D** = $\{t_1, ..., t_n\}$, se $n > 1$ e todas as expressões em **L** são idênticas até o i-ésimo símbolo, exclusive, e t_j é o termo ocorrendo em **L** que começa no i-ésimo símbolo, para $j = 1, ..., n$.
  - Uma substituição simples X/t *satisfaz o teste de ocorrência* se e somente se X não ocorre em t.
  - Um conjunto de discórdia **D** satisfaz o teste de ocorrência se e somente se existem uma variável X e um termo t em **D** tais que X não ocorre em t.

Algoritmo da Unificação:

- *entrada*: um conjunto **L** de literais
- *saída*: um u.m.g. β de **L**, se **L** for unificável
  "NÃO", se **L** não for unificável

Algoritmo da Unificação

1. início
2. $\beta := \epsilon$;
3. **W** := **L**;
4. **D** := conjunto de discórdia de **W**;
5. enquanto (número de literais de **W**) $>$ 1 e **D** satisfaz o teste de ocorrência faça
    1. selecione uma variável X e um termo t em **D** tais que X não ocorre em t;
    2. $\beta := \beta \circ \{X/t\}$;
    3. **W** := **W**$\{X/t\}$;
    4. **D** := conjunto de discórdia de **W**;
6. se (número de literais de **W**) = 1
    - então retorne β
    - senão retorne "NÃO"
7. fim

Sumário

Base de Conhecimento em Língua Natural

1. Marco era um homem.
2. Marco era um pompeiano.
3. Marco nasceu em 40 d.C.
4. Todos os homens são mortais.
5. Todos os pompeianos morreram em 79 d.C. e o vulcão Vesúvio entrou em erupção em 79 d.C.
6. Nenhum mortal vive mais de 150 anos.
7. Vivo significa não morto.
8. Se alguém morre, então está morto para sempre.

Alfabeto de Primeira Ordem - Símbolos não lógicos

1. Constantes:
    - marco
    - vesúvio
2. Predicados:
    - homem(X) = X é homem
    - pompeiano(X) = X é pompeiano
    - nascer(X, Y) = X nasceu no ano Y
    - mortal(X) = X é mortal
    - morrer(X, Y) = X morreu no ano Y
    - erupção(X, Y) = X entrou em erupção no ano Y
    - maior(X, Y) = X é maior que Y
    - morto(X, Y) = X está morto no ano Y
    - vivo(X, Y) = X está vivo no ano Y
3. Símbolo Funcional:
    - menos(X, Y) = X - Y

Base de Conhecimento em Fórmulas da LPPO

1. homem(marco)
2. pompeiano(marco)
3. nascer(marco,40)
4. $\forall X$ (homem(X) $\rightarrow$ mortal(X))
5. $\forall X$ (pompeiano(X) $\rightarrow$ morrer(X,79)) $\wedge$ erupção(vesúvio,79)
6. $\forall X \forall T_1 \forall T_2$((mortal($X$) $\wedge$ nascer(X, T_1) $\wedge$ maior(menos(T_2, T_1),150)) $\rightarrow$ morto(X, T_2))
7. $\forall X \forall T$((vivo(X, T) $\rightarrow \neg$ morto(X, T)) $\wedge$ ($\neg$ morto(X, T) $\rightarrow$ vivo(X, T)))
8. $\forall X \forall T_1 \forall T_2$ ((morrer(X, T_1) $\wedge$ maior(T_2, T_1)) $\rightarrow$ morto(X, T_2))

Algoritmo de Representação Clausal

1. Tome o fecho existencial
2. Elimine quantificadores redundantes
3. Renomeie variáveis quantificadas mais de uma vez
4. Elimine os conectivos "$\rightarrow$" e "$\equiv$"
5. Mova "$\neg$" para o interior da fórmula
6. Mova os quantificadores para o interior da fórmula
7. Elimine os quantificadores existenciais
8. Obtenha a forma normal prenex
9. Obtenha a forma normal conjuntiva
10. Obtenha a representação clausal

Base de Conhecimento na Notação Clausal

1. homem(marco)
2. pompeiano(marco)
3. nascer(marco,40)
4. $\neg$ homem(X) mortal(X)
5.
   - $\neg$ pompeiano(X) morrer(X,79)
   - erupção(vesúvio,79)
6. $\neg$ mortal(X) $\neg$ nascer(X, T_1) $\neg$ maior(menos(T_2, T_1),150) morto(X, T_2)
7.
   - $\neg$ vivo(X, T) $\neg$ morto(X, T)
   - morto(X, T) vivo(X, T)
8. $\neg$ morrer(X, T_1) $\neg$ maior(T_2, T_1) morto(X, T_2)

Referências I

Rosa, J. L. G.
Fundamentos da Inteligência Artificial.
Editora LTC. Rio de Janeiro, 2011. *No prelo*.

Casanova, M. A., Giorno, F. A. C., Furtado, A. L.
Programação em Lógica e a Linguagem Prolog.
Ed. Edgard Blücher Ltda., 1987