

LÓGICA DE PRIMEIRA ORDEM

ADAPTADO DE
[HTTP://AIMA.EECS.BERKELEY.EDU/SLIDES-TEX/](http://aima.eecs.berkeley.edu/slides-tex/)

Resumo

- ◇ Motivação para a Lógica de Primeira Ordem
- ◇ Sintaxe e semântica da LPO
- ◇ Engenharia do conhecimento em LPO
- ◇ Exemplos de representação em LPO

Prós e contras da lógica proposicional

- + A lógica proposicional é *declarativa*: a sintaxe permite expressar factos
- + A lógica proposicional permite representar informação parcial, disjuntiva e negativa (contrariamente à maioria das estruturas de dados e bases de dados)
- + A lógica proposicional é *composicional*: o significado de $B_{1,1} \wedge P_{1,2}$ é obtido compondo o significado de $B_{1,1}$ e de $P_{1,2}$
- + O significado da lógica proposicional é *independente do contexto*
(contrariamente à linguagem natural, em que o significado depende do contexto)
- A lógica proposicional tem um poder expressivo muito limitado
(contrariamente à linguagem natural)
E.g., não se consegue dizer “buracos provocam brisa em casas adjacentes” a não ser que se escreva uma proposição para cada casa do mundo

Lógica de Primeira Ordem

Enquanto que a lógica proposicional assume que o mundo contém *factos*, a lógica de primeira ordem (tal como a linguagem natural) assume que o mundo pode conter

- **Objectos:** pessoas, casas, números, teorias, Gil, cores, jogos de futebol, guerras, séculos . . .
- **Relações:** vermelho, redondo, errado, primo, arranha-céus . . . , irmão de, maior do que, dentro de, parte de, tem cor, ocorreu após, tem, vende, . . .
- **Funções:** pai de, melhor amigo, prolongamento de, um a mais do que, princípio de . . .

Outras designações

A **lógica de primeira ordem** (LPO) é também conhecida através de outras designações:

- ◇ Cálculo de Predicados de Primeira Ordem (first-order predicate calculus)
- ◇ Cálculo de predicados de ordem inferior (lower predicate calculus)
- ◇ Lógica de Predicados (predicate logic)
- ◇ Linguagem de lógica de primeira ordem (language of first-order logic)

Lógicas em geral

Linguagem	Compromisso Ontológico (o que existe no mundo)	Compromisso Epistemológico (o que o agente acredita sobre os factos)
Lógica Proposicional	factos	verdadeiro/falso/desconhecido
Lógica de Primeira Ordem	factos, objectos, relações	verdadeiro/falso/desconhecido
Lógica Temporal	factos, objectos, relações, tempos	verdadeiro/falso/desconhecido
Teoria da Probabilidade	factos	grau de crença $\in [0, 1]$
Lógica Vaga/Difusa	graus de verdade $\in [0, 1]$	intervalo conhecido de valores

Existem inúmeras lógicas, variando com o seu domínio de aplicação:

- ◇ Lógicas terminológicas
- ◇ Lógica de primeira ordem tipificada
- ◇ Lógica de segunda ordem
- ◇ Lógicas de ordem superior
- ◇ Lógica de primeira ordem intuicionista

Sintaxe da LPO: Elementos Básicos

O vocabulário da LPO é constituído pelos seguintes elementos:

1. Símbolos de **constantes de predicados** de aridade ≥ 1 *Brother, >, Irmao, Gato, ...*
2. Símbolos de **constantes de objectos** *KingJohn, 2, UNL, Portugal, Benfica, Reitor, ...*
3. Símbolos de **constantes de funções** *Sqrt, LeftLegOf, ...*
4. Um número infinito de **variáveis** *x, y, a, b, ...*
5. **Conectivos** lógicos $\wedge \vee \neg \Rightarrow \Leftrightarrow$
6. **Quantificadores** $\forall \exists$
7. Parêntesis esquerdo, direito e vírgula $() ,$
8. Igualdade $=$

A lógica de primeira ordem não atribui qualquer interpretação pré-definida aos seus símbolos não lógicos: constantes de predicados, objectos e funções

Regras de formação: termos

O poder expressivo adicional da lógica de primeira ordem advém da sua possibilidade de referir objectos no domínio de discurso. Sintacticamente, os **termos** da lógica de primeira ordem denotam esses objectos:

Um termo é definido recursivamente de acordo com as seguintes regras:

- ◇ Uma constante de objecto é um termo (denotando um objecto concreto do domínio de discurso)
- ◇ Uma variável é um termo (denotando um objecto “anónimo” do domínio de discurso)
- ◇ Qualquer expressão $f(t_1, \dots, t_n)$ é um termo, em que os seus $n \geq 1$ argumentos são termos e f é um símbolo de função com aridade n . Também é designado por expressão funcional.
- ◇ Nada mais é um termo.

Exemplos de termos

Objectos

- 1, 1.54, i, e, 12e40, pi, -3, MMVII, 0x20
- Portugal, UNL, Benfica
- Tweety, Diabo, Bem, Unicórnio
- Abc, Xpto123, Key123, 'Uma cadeia de caracteres muito longa'

Expressões funcionais

- $\text{Exp}(1.0)$, $\text{Exp}(\text{Mult}(I, \text{Pi}))$, $+(x, 0, 0.35)$, $0 * x$ (notação infixa), $\text{Log}(x, 2) + \text{Ln}(E)$
- $\text{Idade}(\text{Carlos}, '17-04-2007 10:23:00 \text{ GMT}')$
- $\text{Peso}(\text{Manuel})$
- 'C' & '++'
- $\text{Mãe}(\text{Árbitro}(\text{Jogo}(\text{Sporting}, \text{Benfica}, \text{Época}200607)))$

Frases Atômicas: (Átomos)

As frases ou fórmulas atômicas são construídas a partir dos símbolos de predicados e termos na linguagem:

- ◇ Se P é um símbolo constante de predicado de aridade n e cada $termo_i (1 \leq i \leq n)$ são termos, então $P(a_1, \dots, a_n)$ é uma fórmula atômica.
- ◇ Se a linguagem inclui igualdade, então $termo_1 = termo_2$ é uma fórmula atômica.

Exemplos de frases atómicas

Brother(KingJohn, RichardTheLionheart)

> (Length(LeftLegOf(Richard)), Length(LeftLegOf(KingJohn)))

*Exp(I * Pi) + 1 = 0*

Matriculado(s123, inf, ciclo1)

0 + x = x

Chove

Arco(a1, a2)

Fórmulas ou frases bem formadas

As fórmulas bem formadas (fbf) definem-se recursivamente através das seguintes regras:

- ◇ Qualquer frase atômica é uma fórmula bem formada (fbf)
- ◇ Se ϕ é uma fórmula bem formada então $\neg\phi$ é uma fbf.
- ◇ Se ϕ and ν são fbfs então $(\phi \wedge \nu)$, $(\phi \vee \nu)$, $(\phi \Rightarrow \nu)$ e $(\phi \Leftrightarrow \nu)$ também são fbfs.
- ◇ Se ϕ é uma fbf e x é uma variável então $\forall x \phi$ e $\exists x \phi$ é uma fbf.
- ◇ Nada mais é uma fbf.

Frases complexas são construídas a partir de frases atômicas utilizando os conectivos e os quantificadores.

Exemplos de fórmulas bem formadas

$Sibling(KingJohn, Richard) \Rightarrow Sibling(Richard, KingJohn)$

$>(1, 2) \vee \leq(1, 2)$

$>(1, 2) \wedge \neg >(1, 2)$

$\forall x \forall y (x + y = y + x)$

$\forall x \forall y \exists z (x < y \Rightarrow (x < z \wedge z < y))$

$\forall y \exists x Progenitor(x, y)$

$\forall x (Humano(x) \Leftrightarrow (Mulher(x) \vee Homem(x)))$

$\exists x (Humano(x) \wedge \neg \exists y Progenitor(x, y))$

Variáveis livres e ligadas

- ◇ Se ϕ é uma fórmula atômica então x é livre sse x ocorre em ϕ .
- ◇ x é livre em $\neg\phi$ sse x é livre em ϕ .
- ◇ x é livre em $(\phi \wedge \nu)$, $(\phi \vee \nu)$, $(\phi \Rightarrow \nu)$, $(\phi \Leftrightarrow \nu)$ sse x é livre em ϕ ou ν .
- ◇ x é livre em $\forall y \phi$ ou $\exists y \phi$ sse $x \neq y$ e x é livre em ϕ .

De forma semelhante define-se a noção de variável ligada (aquelas que ocorrem no âmbito de algum quantificador).

Uma variável pode estar livre e ligada na mesma fórmula:

$$(\forall x (R(x, y) \Rightarrow P(x)) \wedge \forall y (\neg R(x, y) \vee \forall x P(x)))$$

Nota: Qualquer fórmula pode ser reescrita numa fórmula equivalente em que as variáveis livres e ligadas são disjuntas.

Semântica da Lógica de Primeira Ordem

As fbfs são avaliadas em interpretações (ou estruturas) constituídas por um par $M = \langle D, I \rangle$ em que D é um conjunto não vazio (o domínio de discurso) e I uma **função de interpretação**.

O domínio de discurso contém ≥ 1 objectos (**elementos do domínio**) e relações entre eles

A função de interpretação especifica referentes para

símbolos de constante \rightarrow **objectos** $I(c) \in D$ ou

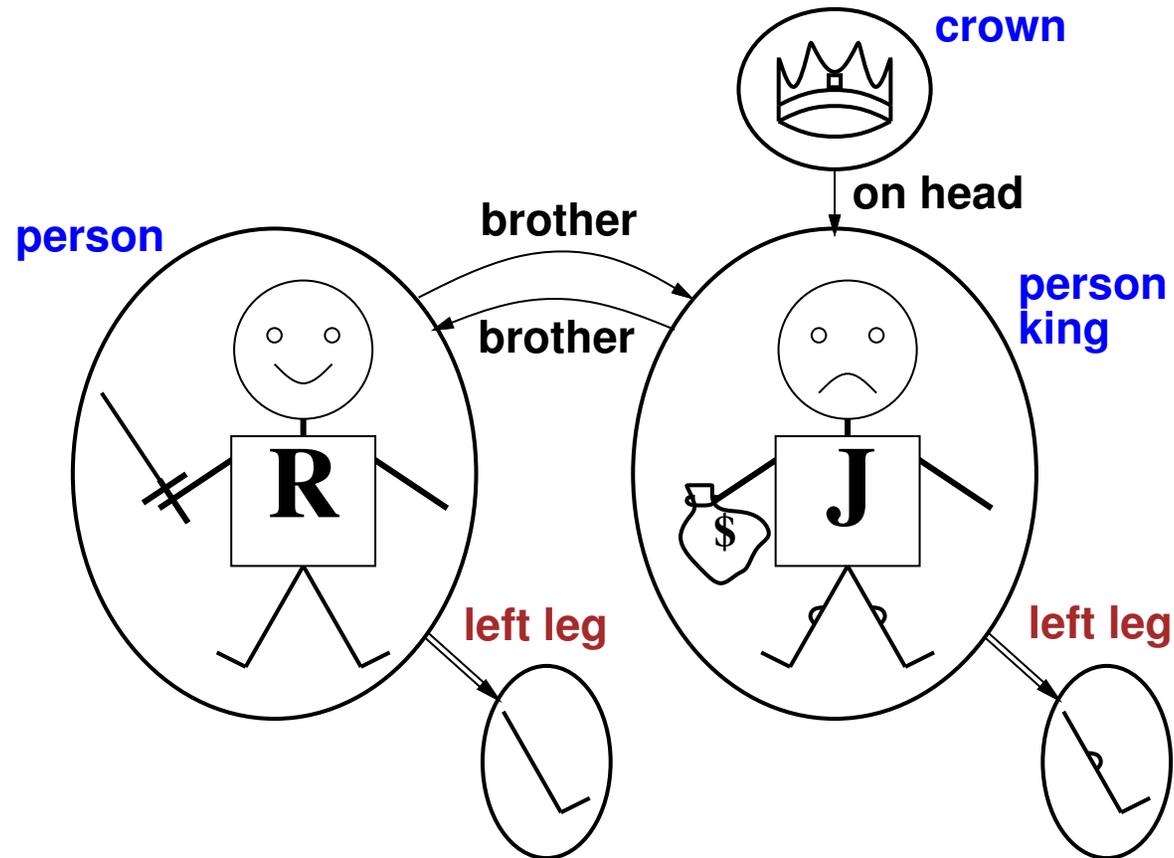
símbolos de predicado \rightarrow **relações** $I(P) \subseteq D^n$ para predicado P/n

símbolos de função \rightarrow **relações funcionais** $I(f) : D^n \rightarrow D$

Uma frase atómica $\text{predicado}(termo_1, \dots, termo_n)$ é verdade sse os **objectos** referidos por $termo_1, \dots, termo_n$ se encontram na **relação** referida por *predicado*

Quando temos fórmulas com variáveis livres é necessário considerar atribuições de variáveis que mapeiam variáveis em elementos do domínio de discurso

Interpretações de LPO: Exemplo



Interpretações para a LPO: Imensas!

Para alguns casos restritos, podemos *tentar* enumerar as interpretações para um dado vocabulário de uma KB:

Para cada número de elementos no domínio n de 1 até ∞

Para cada predicado k -ário P_k no vocabulário

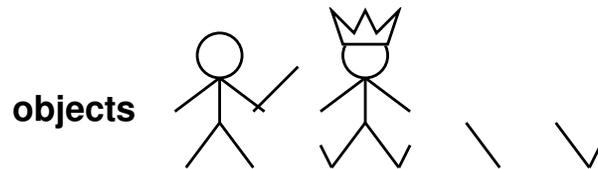
Para cada relação k -ária possível com n objectos

Para cada símbolo de constante C no vocabulário

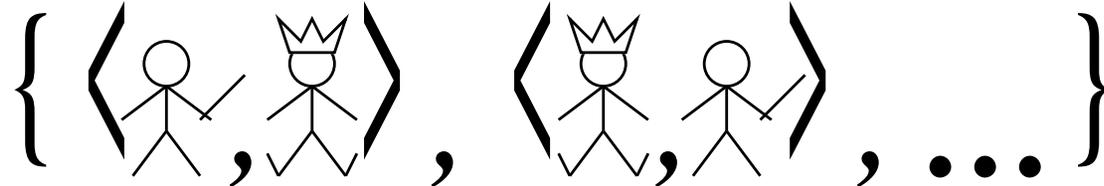
Para cada escolha de referente para C em n objectos ...

A obtenção das conclusões lógicas por enumeração não vai ser nada fácil!

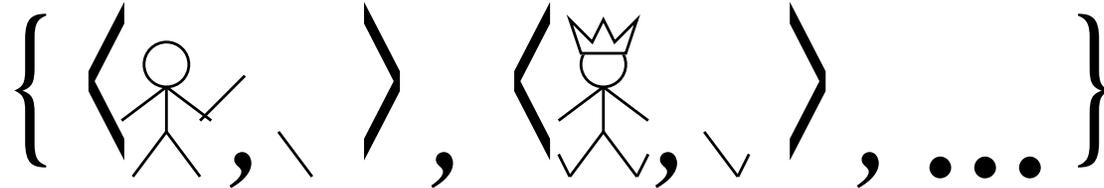
Interpretações para a LPO: Imensas!



relations: sets of tuples of objects



functional relations: all tuples of objects + "value" object



Denotação de um termo numa interpretação

Seja t um termo e s uma atribuição de variáveis numa estrutura M .

A denotação $t_{\mathbf{M}}[s]$ de t em M é definida recursivamente:

◇ $x_{\mathbf{M}}[s] = s(x)$ para uma variável x .

◇ $c_{\mathbf{M}}[s] = I(c)$ para uma constante c .

◇ $f(t_1, \dots, t_n)_{\mathbf{M}}[s] = I(f) ((t_1)_{\mathbf{M}}[s], \dots, (t_n)_{\mathbf{M}}[s])$

Relação de satisfação

A noção de verdade (relativa) em LPO é capturada através da relação de satisfação.

Seja M uma interpretação e s uma atribuição de variáveis em M .

$$M, s \models t_1 = t_2 \text{ sse } (t_1)_M = (t_2)_M$$

$$M, s \models P(t_1, \dots, t_n) \text{ sse } ((t_1)_M, \dots, (t_n)_M) \in I(P)$$

$$M, s \models \neg\phi \text{ sse não é o caso } M, s \models \phi$$

$$M, s \models (\phi \wedge \nu) \text{ sse } M, s \models \phi \text{ e } M, s \models \nu$$

$$M, s \models (\phi \vee \nu) \text{ sse } M, s \models \phi \text{ ou } M, s \models \nu$$

$$M, s \models (\phi \Rightarrow \nu) \text{ sse não é o caso } M, s \models \phi \text{ ou } M, s \models \nu$$

$$M, s \models \forall x \phi \text{ sse } M, s' \models \phi, \text{ para toda a atribuição de variáveis } s' \\ \text{idêntica a } s \text{ excepto possivelmente na variável } x$$

$$M, s \models \exists x \phi \text{ sse } M, s' \models \phi, \text{ para alguma de atribuição de variáveis } s' \\ \text{idêntica a } s \text{ excepto possivelmente na variável } x$$

Consequência Lógica

- ◇ Uma fórmula ϕ é satisfazível se existir uma interpretação M e uma atribuição de variáveis s tal que $M, s \models \phi$.
- ◇ Um conjunto de fbfs Γ é satisfazível se existir uma interpretação M e uma atribuição de variáveis s tal que $M, s \models \psi$ para toda a fórmula ψ de Γ . Se Γ for um conjunto fechado de fórmulas diz-se que M é um modelo de Γ .
- ◇ Se Γ é um conjunto de sentenças então Uma fórmula ϕ é **logicamente verdadeira** ou **válida** se $M, s \models \phi$ para toda a interpretação M e atribuição de variáveis s (representado por $\models \phi$).
- ◇ Seja Γ um conjunto de fórmulas bem formadas e ϕ uma fbfs. Diz-se que ϕ é uma consequência de Γ sse para toda a interpretação M e atribuição de variáveis s se $M, s \models \psi$ para toda a fórmula ψ de Γ então $M, s \models \phi$. Representa-se este facto através de $\Gamma \models \phi$.

Grupos (Abelianos)

Os grupos são definidos pelos seguintes axiomas, num vocabulário contendo uma constante e , um símbolo de função unário $^{-1}$ e um símbolo de função binário \odot :

$$\forall x \quad e \odot x = x \wedge x \odot e = x$$

$$\forall x \quad x^{-1} \odot x = e \wedge x \odot x^{-1} = e$$

$$\forall x \quad \forall y \quad \forall z \quad (x \odot y) \odot z = x \odot (y \odot z)$$

Se o grupo for comutativo diz-se que é abeliano em homenagem a Niels Henrik Abel

$$\forall x \quad \forall y \quad x \odot y = y \odot x$$

Indique uma interpretação M que satisfaça estes axiomas.

Modelos para a teoria dos grupos abelianos

Exemplos típicos de grupos abelianos:

- ◇ Inteiros com adição
- ◇ Números reais sem zero com multiplicação

Algumas consequências lógicas da teoria de grupos:

- Um grupo tem exactamente um único elemento neutro.
- Todo o elemento só tem um inverso.
- Para todo a e b existe um único x tal que $a \odot x = b$.
- Para todo o a , tem-se que $(a^{-1})^{-1} = a$.
- Para todo o a e b $(a \odot b)^{-1} = b^{-1} \odot a^{-1}$

A teoria dos grupos abelianos é decidível!

Quantificação Universal

\forall *<variáveis>* *<frase>*

Toda a gente na UNL é inteligente:

$\forall x \text{ Em}(x, UNL) \Rightarrow \text{Inteligente}(x)$

$\forall x P$ é verdade num dado modelo e dada interpretação nesse modelo sse P é verdade para todo o objecto x do modelo (i.e. x percorre todos os objectos possíveis do modelo)

Pode ser entendido como a **conjunção** das **instanciações** de P

$\text{Em}(\text{ReiAfonsoI}, UNL) \Rightarrow \text{Inteligente}(\text{ReiAfonsoI})$
 $\wedge \text{Em}(\text{Ana}, UNL) \Rightarrow \text{Inteligente}(\text{Ana})$
 $\wedge \text{Em}(UNL, UNL) \Rightarrow \text{Inteligente}(UNL)$
 $\wedge \dots$

Erro comum a evitar

Normalmente, \Rightarrow é o o conectivo principal de \forall

Erro comum: utilizar \wedge como conectivo principal de \forall :

$$\forall x \text{ Em}(x, UNL) \wedge \text{Inteligente}(x)$$

significa “Toda a gente está na UNL e toda a gente é inteligente”

Quantificação Existencial

$\exists \langle \text{variáveis} \rangle \langle \text{frase} \rangle$

Alguém em Almada é inteligente:

$\exists x \text{ Em}(x, \text{Almada}) \wedge \text{Inteligente}(x)$

$\exists x P$ é verdade num dado modelo e dada interpretação nesse modelo sse P é verdade para algum objecto x possível do modelo

Pode ser entendido como a **disjunção** das **instanciações** de P

$\text{Em}(\text{ReiAfonsoI}, \text{Almada}) \wedge \text{Inteligente}(\text{ReiAfonsoI})$
 $\vee \text{Em}(\text{Ana}, \text{Almada}) \wedge \text{Inteligente}(\text{Ana})$
 $\vee \text{Em}(\text{Almada}, \text{Almada}) \wedge \text{Inteligente}(\text{Almada})$
 $\vee \dots$

Outro erro comum a evitar

Normalmente, \wedge é o conectivo principal de \exists

Erro comum: utilizar \Rightarrow como conectivo principal de \exists :

$$\exists x \text{ Em}(x, \text{Almada}) \Rightarrow \text{Inteligente}(x)$$

é verdade se existir alguém que não está em Almada!

Propriedades dos quantificadores

$\forall x \forall y$ é o mesmo que $\forall y \forall x$ (porquê??)

$\exists x \exists y$ é o mesmo que $\exists y \exists x$ (porquê??)

$\exists x \forall y$ **não** é o mesmo que $\forall y \exists x$

$\exists x \forall y \text{ Ama}(x, y)$

“Existe alguém que ama toda a gente no mundo”

$\forall y \exists x \text{ Ama}(x, y)$

“Toda a gente no mundo é amada pelo menos por uma pessoa”

Dualidade dos quantificadores: cada um pode ser expresso utilizando o outro

$\forall x \text{ Gosta}(x, \text{Gelado}) \quad \neg \exists x \neg \text{Gosta}(x, \text{Gelado})$

$\exists x \text{ Gosta}(x, \text{Bróculos}) \quad \neg \forall x \neg \text{Gosta}(x, \text{Bróculos})$

Equivalências importantes da LPO

$$\neg \forall x P(x) \Leftrightarrow \exists x \neg P(x)$$

$$\neg \exists x P(x) \Leftrightarrow \forall x \neg P(x)$$

$$\forall x \forall y P(x, y) \Leftrightarrow \forall y \forall x P(x, y)$$

$$\exists x \exists y P(x, y) \Leftrightarrow \exists y \exists x P(x, y)$$

$$\forall x P(x) \wedge \forall x Q(x) \Leftrightarrow \forall x (P(x) \wedge Q(x))$$

$$\exists x P(x) \vee \exists x Q(x) \Leftrightarrow \exists x (P(x) \vee Q(x))$$

$$P \wedge \exists x Q(x) \Leftrightarrow \exists x (P \wedge Q(x)) \quad \text{com } x \text{ não livre em } P$$

$$P \vee \forall x Q(x) \Leftrightarrow \forall x (P \vee Q(x)) \quad \text{com } x \text{ não livre em } P$$

Implicações importantes da LPO

$$\exists x \forall y P(x, y) \Rightarrow \forall y \exists x P(x, y)$$

$$\forall x P(x) \vee \forall x Q(x) \Rightarrow \forall x (P(x) \vee Q(x))$$

$$\exists x (P(x) \wedge Q(x)) \Rightarrow \exists x P(x) \wedge \exists x Q(x)$$

$$\exists x P(x) \wedge \forall x Q(x) \Rightarrow \exists x (P(x) \wedge Q(x))$$

$$\forall x P(x) \Rightarrow P(c) \quad \text{se } c \text{ é uma variável, então não deve ocorrer quantificada em } P(x)$$

$$P(c) \Rightarrow \exists x P(x) \quad x \text{ não deve ocorrer livre em } P(c)$$

Alguns Exemplos

Irmãos são amigos

Alguns Exemplos

Irmãos são amigos

$$\forall x, y \text{ Irmão}(x, y) \Rightarrow \text{Amigo}(x, y).$$

A “Amizade” é simétrica

Alguns Exemplos

Irmãos são amigos

$$\forall x, y \text{ Irmão}(x, y) \Rightarrow \text{Amigo}(x, y).$$

A “Amizade” é simétrica

$$\forall x, y \text{ Amigo}(x, y) \Leftrightarrow \text{Amigo}(y, x).$$

A mãe de alguém é o seu progenitor feminino

Alguns Exemplos

Irmãos são amigos

$$\forall x, y \text{ Irm\~{a}o}(x, y) \Rightarrow \text{Amigo}(x, y).$$

A “Amizade” é simétrica

$$\forall x, y \text{ Amigo}(x, y) \Leftrightarrow \text{Amigo}(y, x).$$

A mãe de alguém é o seu progenitor feminino

$$\forall x, y \text{ Mãe}(x, y) \Leftrightarrow (\text{Feminino}(x) \wedge \text{Progenitor}(x, y)).$$

Um primo direito é um filho de um dos irmãos dos pais

Alguns Exemplos

Irmãos são amigos

$$\forall x, y \text{ Irm\~{a}o}(x, y) \Rightarrow \text{Amigo}(x, y).$$

A “Amizade” é simétrica

$$\forall x, y \text{ Amigo}(x, y) \Leftrightarrow \text{Amigo}(y, x).$$

A mãe de alguém é o seu progenitor feminino

$$\forall x, y \text{ Mãe}(x, y) \Leftrightarrow (\text{Feminino}(x) \wedge \text{Progenitor}(x, y)).$$

Um primo direito é um filho de um dos irmãos dos pais

$$\forall x, y \text{ PrimoDireito}(x, y) \Leftrightarrow \exists p, ps \text{ Progenitor}(p, x) \wedge \text{Irm\~{a}o}(ps, p) \wedge \text{Progenitor}(ps, y)$$

Igualdade

$termo_1 = termo_2$ é verdade numa dada interpretação
sse $termo_1$ e $termo_2$ se referem ao mesmo objecto

E.g., $1 = 2$ e $\forall x \times (Sqrt(x), Sqrt(x)) = x$ são satisfazíveis
 $2 = 2$ é válida

E.g., definição de *Irmão* em termos de *Progenitor*:

$$\forall x, y \text{ Irmão}(x, y) \Leftrightarrow [\neg(x = y) \wedge \exists m, f \neg(m = f) \wedge \\ Progenitor(m, x) \wedge Progenitor(f, x) \wedge Progenitor(m, y) \wedge Progenitor(f, y)]$$

Lógicas de Descrição ou Terminológicas

Os resultados de indecidibilidade da lógica de primeira ordem motivaram a investigação de subconjuntos decidíveis “interessantes” da LPO.

A lógica terminológica de base é a lógica \mathcal{ALC} (Attributive Logic with Complements) com a seguinte sintaxe (repare-se na ausência explícita de variáveis).

Sintaxe	Significado	Tradução LPO
\perp	conjunto vazio de indivíduos	\perp
A	Conceito simples: <i>Pessoa</i>	$Pessoa(x)$
$\neg C$	Negação de um conceito: $\neg Homem$	$\neg Homem(x)$
$C \sqcap D$	Conjunção: <i>Mulher</i> \sqcap <i>Estudante</i>	$Mulher(x) \wedge Estudante(x)$
$C \sqcup D$	Disjunção: <i>Bonito</i> \sqcup <i>Rico</i>	$Bonito(x) \vee Rico(x)$
$\exists R.C$	Algum: <i>Mulher</i> \sqcap $\exists progenitor.Pessoa$	$Mulher(x) \wedge$ $\exists_y (progenitor(x, y) \wedge Pessoa(y))$
$\forall R.C$	Qualquer: <i>Mulher</i> \sqcap $\forall progenitor.Homem$	$Mulher(x) \wedge$ $\forall_y (progenitor(x, y) \Rightarrow Homem(y))$

A Web Semântica

As lógicas de descrição estão a ser utilizadas para modelar informação na Web Semântica. Uma linguagem suportada pelo W3C é a Ontology Web Language 1.1 (OWL), correspondendo à linguagem de descrição *sROIQ*.

$Pessoa \sqsubseteq SerVivo$

$Pessoa \equiv Homem \sqcup Mulher$

$Homem \sqsubseteq \neg Mulher$

$Mãe \equiv Mulher \sqcap progenitor.Pessoa$

$\top \equiv (\geq 1 \text{ progenitor}^-.Homem) \sqcap (\leq 1 \text{ progenitor}^-.Homem)$

$\top \equiv (\geq 1 \text{ progenitor}^-.Mulher) \sqcap (\leq 1 \text{ progenitor}^-.Mulher)$

$progenitor \sqsubseteq antecessor$

$progenitor \circ antecessor \sqsubseteq antecessor$

$carlos : Homem$

$paula : Mulher$

$(carlos, rita) : progenitor$

$(paula, rita) : progenitor$

Interacção com KBs em LPO

Suponhamos que um agente do mundo do Wumpus recorre a uma KB em LPO e percepção um cheiro e uma brisa (mas não uma cintilação) em $t = 5$:

$Tell(KB, Percept([Smell, Breeze, None], 5))$

$Ask(KB, \exists a \text{ BestAction}(a, 5))$

I.e., Será que a KB conclui alguma acção particular para $t = 5$?

Resposta: *Yes*, $\{a/Shoot\}$ ← substituição (binding list)

Dada uma frase S e a substituição σ ,

$S\sigma$ denota o resultado aplicar σ a S ; e.g.,

$S = MaisInteligente(x, y)$

$\sigma = \{x/Hillary, y/Bill\}$

$S\sigma = MaisInteligente(Hillary, Bill)$

$Ask(KB, S)$ devolve alguns/todos os σ tal que $KB \models S\sigma$

Base de conh. para o mundo do Wumpus

“Percepção”

$\forall b, g, t \text{ Percept}([Smell, b, g], t) \Rightarrow Smelt(t)$

$\forall s, b, t \text{ Percept}([s, b, Glitter], t) \Rightarrow AtGold(t)$

Reflexos: $\forall t \text{ AtGold}(t) \Rightarrow BestAction(Grab, t)$

Reflexos com estado interno: já temos o ouro?

$\forall t \text{ AtGold}(t) \wedge \neg Holding(Gold, t) \Rightarrow BestAction(Grab, t)$

Holding(Gold, *t*) não pode ser observado

\Rightarrow manter as alterações é essencial

Dedução de propriedades invisíveis

Propriedades das posições:

$$\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Smelt}(t) \Rightarrow \text{Smelly}(x)$$

$$\forall x, t \text{ At}(\text{Agent}, x, t) \wedge \text{Breeze}(t) \Rightarrow \text{Breezy}(x)$$

Casas são ventosas ao pé de um buraco:

Regra de **diagnóstico** —inferir causa a partir do efeito

$$\forall y \text{ Breezy}(y) \Rightarrow \exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)$$

Regra **causal** —inferir efeitos a partir das causas

$$\forall x, y \text{ Pit}(x) \wedge \text{Adjacent}(x, y) \Rightarrow \text{Breezy}(y)$$

Nenhuma delas é completa—e.g., a regra causal não diz como casas longe dos buracos podem ser ventosas

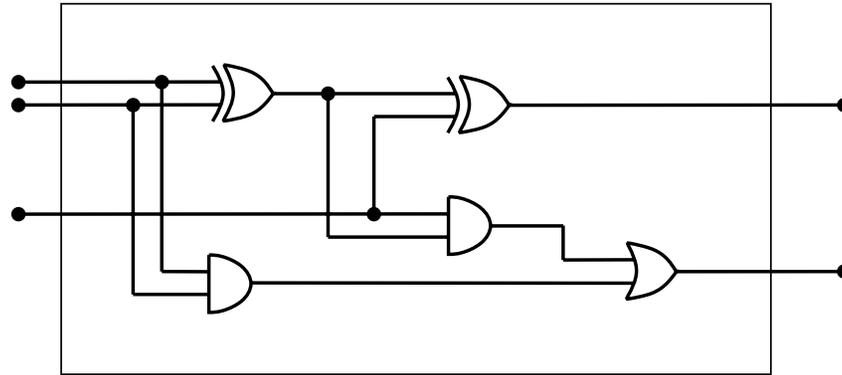
Definição para o predicado *Breezy*:

$$\forall y \text{ Breezy}(y) \Leftrightarrow [\exists x \text{ Pit}(x) \wedge \text{Adjacent}(x, y)]$$

Engenharia de Conhecimento em LPO

1. Identificar a tarefa
2. Obter o conhecimento relevante
3. Decidir qual o vocabulário: predicados, funções e constantes
4. Codificar conhecimento genérico acerca do domínio
5. Codificar uma instância concreta
6. Interrogar a teoria utilizando um motor de inferência
7. Depurar a base de conhecimento

Circuitos digitais



Vocabulário

Objectos : $A_1, A_2, X_1, X_2, O_1, C_1, 0, 1, 2, 3, OR, AND, XOR, NOT$.

Funções : *Sinal/1, Type/1, In/2 e Out/2*.

Predicados : *Ligado/2*.

Modelação de um circuito digital

◇ Conhecimento genérico sobre o domínio – ligações:

1. Se dois terminais estão ligados então têm o mesmo sinal:

$$\forall t_1 \forall t_2 \text{ Ligado}(t_1, t_2) \Rightarrow \text{Sinal}(t_1) = \text{Sinal}(t_2)$$

2. O sinal para todo o terminal ou é 0 ou 1 (mas não ambos)

$$\forall t \text{ Sinal}(t) = 0 \vee \text{Sinal}(t) = 1$$
$$1 \neq 0$$

3. O predicado *Ligado* é comutativo

$$\forall t_1 \forall t_2 \text{ Ligado}(t_1, t_2) \equiv \text{Ligado}(t_2, t_1)$$

Modelação de um circuito digital

◇ Conhecimento genérico sobre o domínio – portas:

1. A saída de uma porta OR é 1 sse pelo menos uma das suas entradas é 1

$$\forall_g \text{Type}(g) = OR \Rightarrow (\text{Sinal}(\text{Out}(1, g)) = 1 \equiv \exists_n \text{Sinal}(\text{In}(n, g)) = 1)$$

2. A saída de uma porta AND é 0 sse pelo menos uma das entradas é 0

$$\forall_g \text{Type}(g) = AND \Rightarrow (\text{Sinal}(\text{Out}(1, g)) = 0 \equiv \exists_n \text{Sinal}(\text{In}(n, g)) = 0)$$

3. A saída de uma porta XOR é 1 sse as suas duas entradas são diferentes

$$\forall_g \text{Type}(g) = XOR \Rightarrow \\ (\text{Sinal}(\text{Out}(1, g)) = 1 \equiv \text{Sinal}(\text{In}(1, g)) \neq \text{Sinal}(\text{In}(2, g)))$$

4. A saída de uma porta NOT é diferente da sua entrada:

$$\forall_g \text{Type}(g) = NOT \Rightarrow (\text{Sinal}(\text{Out}(1, g)) \neq \text{Sinal}(\text{In}(1, g)))$$

Modelação de um circuito digital

◇ Codificação da instância

Portas existentes:

$$Type(X_1) = XOR \quad Type(X_2) = XOR$$

$$Type(A_1) = AND \quad Type(A_2) = AND \quad Type(O_1) = OR$$

Ligações entre os componentes:

$$Ligado(Out(1, X_1), In(1, X_2)) \quad Ligado(In(1, C_1), In(1, X_1))$$

$$Ligado(Out(1, X_1), In(2, A_2)) \quad Ligado(In(1, C_1), In(1, A_1))$$

$$Ligado(Out(1, A_1), In(1, O_1)) \quad Ligado(In(2, C_1), In(2, X_1))$$

$$Ligado(Out(1, A_2), In(2, O_1)) \quad Ligado(In(2, C_1), In(2, A_1))$$

$$Ligado(Out(1, X_2), Out(1, C_1)) \quad Ligado(In(3, C_1), In(2, X_2))$$

$$Ligado(Out(1, O_1), Out(2, C_1)) \quad Ligado(In(3, C_1), In(1, A_2))$$

Interrogações à teoria

- ◇ Saber quais os valores de input necessários para se ter a primeira saída a 0 e a segunda saída a 1 ?

$$\begin{aligned} \text{Signal}(In(1, C_1)) = i_1 \wedge \text{Signal}(In(2, C_1)) = i_2 \wedge \text{Signal}(In(3, C_1)) = i_3 \\ \wedge \\ \text{Signal}(Out(1, C_1)) = 0 \wedge \text{Signal}(Out(2, C_2)) = 1 \end{aligned}$$

- ◇ Obter a tabela de entrada-saída para o circuito:

$$\begin{aligned} \text{Signal}(In(1, C_1)) = i_1 \wedge \text{Signal}(In(2, C_1)) = i_2 \wedge \text{Signal}(In(3, C_1)) = i_3 \\ \wedge \\ \text{Signal}(Out(1, C_1)) = o_1 \wedge \text{Signal}(Out(2, C_2)) = o_2 \end{aligned}$$

Nota: Pode-se generalizar a modelação para efectuar diagnóstico: sabendo quais os valores de entrada e saída determinar quais os circuitos que se encontram avariados (e quais as suas falhas).

Sumário

- ◇ Lógica de Primeira Ordem:
 - objectos e relações são primitivas semânticas
 - sintaxe: constantes, funções, predicados, igualdade, quantificadores
- ◇ Maior poder expressivo: suficiente para definir o mundo do Wumpus e circuitos electrónicos.
- ◇ Apesar da sua expressividade existem limitações inerentes que não são ultrapassáveis.
- ◇ Teoremas meta-lógicos fornecem resultados fundamentais para toda a Matemática e Informática.
- ◇ Raciocínio em Lógica de Primeira Ordem é indecidível.
- ◇ Estão a ser activamente estudados fragmentos decidíveis e suficientemente expressivos para representar conhecimento ontológico.