Sistemas Baseados em Conhecimento
Aula 4

Renata Wassermann

renata@ime.usp.br

2017

Números Naturais

- Uma constante: 0
- Uma função: S (sucessor)

Axiomas de Peano

- NatNum(0)
- $\forall$n NatNum(n) $\rightarrow$ NatNum(S(n))

NatNum: 0, S(0), S(S(0)), S(S(S(0))), ...

Números Naturais

Restrições da função sucessor:

- $\forall n\ \neg(0{=}S(n))$
- $\forall m,n\ \neg(m{=}n) \rightarrow \neg(S(m){=}S(n))$

Soma:

- $\forall m\ NatNum(m) \rightarrow +(m,0){=}m$
- $\forall m,n\ NatNum(m) \wedge NatNum(n) \rightarrow +(m,S(n)){=}S(+(m,n))$

Números Naturais

Restrições da função sucessor:

- $\forall n\ \neg(0{=}S(n))$
- $\forall m,n\ \neg(m{=}n) \rightarrow \neg(S(m){=}S(n))$

Soma (infixa):

- $\forall m\ NatNum(m) \rightarrow m+0=m$
- $\forall m,n\ NatNum(m) \wedge NatNum(n) \rightarrow m+S(n) = S(m+n)$

Números Naturais

- Multiplicação
- Exponenciação
- Divisão inteira
- ...
- Teoria dos números pode ser construída a partir de uma constante, uma função, um predicado e quatro axiomas.

Conjuntos

- Constante: $\emptyset$
- Predicado unário: Set
- Predicados binários: $\in$, $\subseteq$
- Funções binárias: $\cup$, $\cap$, $\{x|s\}$ (elemento x adicionado ao conjunto s)

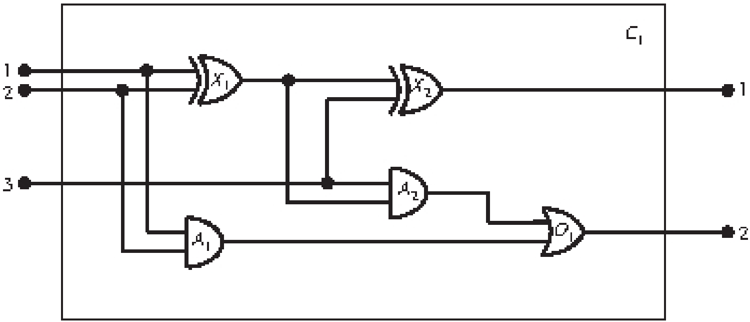
Conjuntos - axiomas

- $\forall$s Set(s) $\leftrightarrow$ (s $=\emptyset$) $\vee$ $(\exists x, s_2 \text{ Set}(s_2) \wedge s = \{x|s2\})$
- $\neg\exists x, s\ \{x|s\} = \emptyset$
- $\forall x, s\ x \in s \rightarrow s = \{x|s\}$
- $\forall x, s\ x \in s \leftrightarrow \exists y, s_2(s = \{y|s_2\} \wedge (x = y \vee x \in s_2))$
- $\forall s_1, s_2\ s_1 \subseteq s_2 \leftrightarrow (\forall x\ x \in s_1 \rightarrow x \in s_2)$
- $\forall s_1, s_2\ s_1 = s_2 \leftrightarrow (s_1 \subseteq s_2 \wedge s_2 \subseteq s_1)$
- $\forall x, s_1, s_2\ x \in (s_1 \cap s_2) \leftrightarrow (x \in s_1 \wedge x \in s_2)$
- $\forall x, s_1, s_2\ x \in (s_1 \cup s_2) \leftrightarrow (x \in s_1 \vee x \in s_2)$

Engenharia de Conhecimento em FOL

1. Identificar uma tarefa;
2. Agregar conhecimento relevante;
3. Definir um vocabulário de predicados, funções, e constantes (ontologia);
4. Codificar o conhecimento geral sobre o domínio;
5. Codificar uma descrição da instância específica do problema;
6. Formular consultas ao procedimento de inferência e obter respostas;
7. Depurar a base de conhecimento.

Exemplo: circuito somador de bit



1. Identificar a tarefa

- O circuito adiciona de maneira correta? (verificação do circuito)
- Se todas as entradas são 1, qual a saída de A2?
- ...

Não queremos saber:

- Custo de produção
- Consumo de energia
- ...

2. Agregar conhecimento relevante

- Composto de cabos e portas
- Tipos de portas (AND, OR, XOR, NOT)
- Cada porta recebe sinais de entrada e produz um sinal de saída
- ...

Irrelevante: tamanho, forma, cor, ...

3. Definir um vocabulário

- Constantes para portas: A_1, A_2, X_1, X_2, O_1, C
- Tipo das portas:
  - Type(X_1) $=$ XOR (função e constante XOR) ou
  - Type(X_1,XOR) (predicado e constante XOR) ou
  - XOR(X_1) (predicado XOR)
  - Função assegura um único tipo para cada porta!
- Terminais: $X_1 In_1 \times In(1, X_1)$
- Conexões: $Connected(Out(1, X_1), In(1, X_2))$
- Sinal: constantes 1 e 0 e função *Signal*

4. Codificar o conhecimento **geral** sobre o domínio

- $\forall t_1, t_2 Connected(t_1, t_2) \rightarrow Signal(t_1) = Signal(t_2)$
- $\forall t Signal(t) = 1 \vee Signal(t) = 0$
  $\neg(1 = 0)$
- $\forall t_1, t_2 Connected(t_1, t_2) \rightarrow Connected(t_2, t_1)$
- $\forall g Type(g) = OR \rightarrow$
  $(Signal(Out(1, g)) = 1 \leftrightarrow \exists n Signal(In(n, g)) = 1)$
- $\forall g Type(g) = AND \rightarrow$
  $(Signal(Out(1, g)) = 0 \leftrightarrow \exists n Signal(In(n, g)) = 0)$
- $\forall g Type(g) = XOR \rightarrow$
  $(Signal(Out(1, g)) = 1 \leftrightarrow Signal(In(1, g)) \neq Signal(In(2, g)))$
- $\forall g Type(g) = NOT \rightarrow$
  $\neg(Signal(Out(1, g)) = Signal(In(1, g)))$

5. Codificar **instância específica** do problema

$$
\begin{array}{ll}
Type(X_1) = XOR & Type(X_2) = XOR \\
Type(A_1) = AND & Type(A_2) = AND \\
Type(O_1) = OR &
\end{array}
$$

$$
\begin{array}{ll}
Connected(Out(1, X_1), In(1, X_2)) & Connected(In(1, C), In(1, X_1)) \\
Connected(Out(1, X_1), In(2, A_2)) & Connected(In(1, C), In(1, A_1)) \\
Connected(Out(1, A_2), In(1, O_1)) & Connected(In(2, C), In(2, X_1)) \\
Connected(Out(1, A_1), In(2, O_1)) & Connected(In(2, C), In(2, A_1)) \\
Connected(Out(1, X_2), Out(1, C)) & Connected(In(3, C), In(2, X_2)) \\
Connected(Out(1, O_1), Out(2, C)) & Connected(In(3, C), In(1, A_2))
\end{array}
$$

6. Formular consultas

Que combinações de entradas fariam a primeira saída de C (o bit de soma) ser 0 e a segunda saída de C (o bit de transporte) ser 1?

$$\exists i_1, i_2, i_3 Signal(In(1, C)) = i_1 \land Signal(In(2, C)) = i_2 \land \\ Signal(In(3, C)) = i_3 \land Signal(Out(1, C)) = 0 \land \\ Signal(Out(2, C)) = 1$$

Resposta: conjunto de substituições

$$\{\{i_1/1, i_2/1, i_3/0\}, \{i_1/1, i_2/0, i_3/1\}, \{i_1/0, i_2/1, i_3/1\}\}$$

6. Formular consultas

Quais são os possíveis valores de todos os terminais do circuito?

$$\exists i_1, i_2, i_3, o_1, o_2 \, Signal(In(1, C)) = i_1 \wedge Signal(In(2, C)) = i_2 \wedge$$
$$Signal(In(3, C)) = i_3 \wedge Signal(Out(1, C)) = o_1 \wedge$$
$$Signal(Out(2, C)) = o_2$$

Resposta: Tabela completa de entrada e saída (verificação).

7. Depurar a base de conhecimento

- Sistema só responde para entradas 000 e 110.
- Testar porta X_1:
  $\exists i_1, i_2, o Signal(In(1, C)) = i_1 \wedge$
  $\quad Signal(In(2, C)) = i_2 \wedge Signal(Out(1, X_1)) = o$
- Falha para entradas 10 e 01.
- Axioma para XOR:
  $Signal(Out(1, X_1)) = 1 \leftrightarrow$
  $\quad Signal(In(1, X_1)) \neq Signal(In(2, X_1))$
- Para 10:
  $Signal(Out(1, X_1)) = 1 \leftrightarrow 1 \neq 0$

Omissão de $1 \neq 0$: