Lógica Computacional

Aula Teórica 2: Sintaxe da Lógica Proposicional e Definições Indutivas

Ricardo Gonçalves

Departamento de Informática

15 de setembro de 2023

Representação do Conhecimento

- Modelação de um cenário usando linguagem lógica
- Permite depois raciocínio automático para tirar conclusões implícitas
- Exemplo:
  SNOMED CT
  Systematized Nomenclature of Medicine – Clinical Terms
- Lógicas de Descrição
  Subconjunto da Lógica de Primeira Ordem que permite raciocínio eficiente

Tradução da linguagem natural para lógica proposicional

Símbolos proposicionais

- Identificar as asserções básicas
  - Passíveis de terem valor de verdade
- Exemplos:
  - 'Estudo hoje' - eh
  - 'Vou à FCT' - vf
  - 'Tenho boa nota a LC' - bn
- Associar um símbolo proposicional a cada asserção básica

Tradução da linguagem natural para lógica proposicional

Negação

- Negar explicitamente alguma afirmação mais simples
- usamos o símbolo $\neg$ para representar formalmente
- identificar "não" ou "não é verdade que" ou semelhante
  - "O João não está doente"
  - "Não é verdade que a Maria tenha faltado às aulas"

Tradução da linguagem natural para lógica proposicional

Disjunção

- Representa alternativa
- usamos o símbolo $\vee$ para representar formalmente
- Identificar 'ou' na frase
    - 'Estudo hoje ou amanhã'
- Pode ser ambíguo (quando combinado com conjunção)
- Frases que comecem por 'ou' procuram não ser ambíguas:
    - 'Ou estudo hoje ou (estudo amanhã e não vou ao jogo)'

Tradução da linguagem natural para lógica proposicional

Conjunção

- Só é verdade se cada parte for
- usamos o símbolo $\wedge$ para representar formalmente
- Identificar 'e' na frase
  - 'Estudo hoje e amanhã'
- Pode ser ambíguo (quando combinado com disjunção)
- Outras frases que representam conjunções:
  - 'tanto p como q'
  - 'p tal como q'

Tradução da linguagem natural para lógica proposicional

Implicação

- Captura a noção de consequência
- usamos o símbolo $\rightarrow$ para representar formalmente
- À esquerda está o *antecedente* e à direita o *consequente*.
  - 'Se estudar para o teste então vou ter boa nota'

Muito importante perceber!

Só há um cenário em que uma implicação 'Se A então B' é falsa:

A é verdade e B é falso

Em todos os outros casos a implicação é verdadeira!

Tradução da linguagem natural para lógica proposicional

Implicação

- Frases que são representadas por $p \to q$:
    - 'se p então q' ou 'se p, q'
    - 'caso p então q' ou 'caso p, q' ou 'como p, q'
    - 'q se p'
    - 'q desde que p'
    - 'p só se q'
- Frase que é representada por $\neg p \to q$:
    - 'q a não ser que p'

Tradução da linguagem natural para lógica proposicional

A implicação não é tão intuitiva como os outros conectivos.

"desde que" indica uma condição suficiente

"Passo a LC desde que tenha 10 ou mais no exame"

$$\texttt{dez+LC} \rightarrow \texttt{passarLC}$$

"só se" indica uma condição necessária

"Passo a LC só se tiver estudado"

$$\texttt{passarLC} \rightarrow \texttt{estudo}$$

"a não ser que" indica uma única excepção

"Vou às aulas de LC a não ser que esteja doente"

$$\neg\texttt{doente} \rightarrow \texttt{vouLC}$$

Em caso de dúvida, pensem em cenários que falsifiquem a situação descrita. Estes devem também falsificar a fórmula.

Tradução da linguagem natural para lógica proposicional

Asserções básicas e compostas

- ‘gosto de lógica e de álgebra’ - traduz-se para $p \wedge q$
- asserções básicas
  - ‘gosto de lógica’ escreve-se p
  - ‘gosto de álgebra’ escreve-se q
- ‘gosto de lógica ou de álgebra’ traduz-se para $p \vee q$;
- ‘gosto de lógica ou de álgebra e de análise’ é ambígua, mas
- ‘ou gosto de lógica ou de álgebra e de análise’ $p \vee (q \wedge r)$;
- Há ambiguidades difíceis de resolver:
- ‘o Pedro foi ao médico e ficou doente’- $m \wedge d$ ou $m \wedge (m \rightarrow d)$
- Comutatitividade? ‘o Pedro ficou doente e foi ao médico ’ aponta mais para $d \wedge (d \rightarrow m)$ do que para $d \wedge m$.

Tradução da linguagem natural para lógica proposicional

Exercícios

- A Ana foi ao jogo da sua equipa porque tinha bilhete e companhia.
- O José vai ao jogo da sua equipa se tiver bilhete e companhia, a não ser que esteja mau tempo.

Menor conjunto gerado por regras

Forma "construtiva" (ou incremental) de definir conjuntos *infinitos*

Como funciona?

- Indica-se primeiro ("axiomatiza-se") quais são os elementos "básicos" do conjunto (em número *finito*).
- Depois definem-se "regras" (em número *finito*) para obter novos elementos a partir dos que já estão no conjunto.

O conjunto resultante contém *todos* os elementos que se podem gerar com as regras, e *apenas* esses.

Definição indutiva: menor conjunto gerado por regras

Nota

- Como não temos limite no número de vezes que aplicamos as regras, podemos obter um número infinito de elementos a partir de um número finito de axiomas e regras!
- Cada elemento do conjunto tem no entanto uma justificação, prova ou *derivação finita*: a sequência (finita) das regras aplicadas para o obter.

Exemplos

- A linguagem F_P da Lógica Proposicional
- Os números naturais

Definição indutiva dos naturais

Definição indutiva de $\mathbb{N}_0$

- *ZERO*: $0 \in \mathbb{N}_0$
  - Fixa-se primeiro que zero é natural - axioma
- *SUCC*: se $n \in \mathbb{N}_0$ então $n + 1 \in \mathbb{N}_0$
  - Obter novos elementos a partir dos que já pertencem ao conjunto - regra

Prova de pertença a conjunto definido indutivamente

Pertença de elemento a conjunto definido indutivamente

Justificação é uma sequência finita de passos tal que:

- último passo é o elemento que se quer justificar
- cada passo é um axioma ou uma regra aplicada a passos anteriores

Definição indutiva dos naturais

Exemplo

Justificação para $3 \in \mathbb{N}_0$:

1. $0 \in \mathbb{N}_0$ por *ZERO*
2. $1 = 0 + 1 \in \mathbb{N}_0$ por *SUCC* aplicado ao passo 1
3. $2 = 1 + 1 \in \mathbb{N}_0$ por *SUCC* aplicado ao passo 2
4. $3 = 2 + 1 \in \mathbb{N}_0$ por *SUCC* aplicado ao passo 3

Definição indutiva de pilhas de inteiros

Definição indutiva de *PilhaInt*

- *VAZIA*: $vazia \in PilhaInt$
  - A pilha vazia é uma pilha de inteiros - axioma
- *PUSH*: se $i \in \mathbb{Z}$ e $p \in PilhaInt$ então $push(i, p) \in PilhaInt$
  - Obter uma nova pilha a partir de um inteiro e de uma pilha já gerada - regra

Qual a formal geral de um elemento de *PilhaInt*?

Definição indutiva de pilhas de inteiros

Exemplo

Justificação para $push(4, push(-7, push(-1, vazia))) \in PilhaInt$:

1. $vazia \in PilhaInt$
   - axioma *VAZIA*
2. $push(-1, vazia) \in PilhaInt$
   - regra *PUSH* aplicada a (1) e a $-1 \in \mathbb{Z}$
3. $push(-7, push(-1, vazia)) \in PilhaInt$
   - regra *PUSH* aplicada a (2) e a $-7 \in \mathbb{Z}$
4. $push(4, push(-7, push(-1, vazia))) \in PilhaInt$
   - regra *PUSH* aplicada a (3) e a $4 \in \mathbb{Z}$

Provas como árvores etiquetadas

A ideia

- Apresentar as provas em árvore, dita de dedução ou derivação.
- Cada árvore é construída a partir de árvores singulares (ou folhas) utilizando as regras da definição indutiva.
- Cada novo nível da árvore é obtido por aplicação de uma regra.
- A etiquetas dos nós são elementos do conjunto.
  - Os elementos nas folhas resultam de axiomas.
  - O elemento na raíz é a conclusão da prova. Diz-se que a árvore é uma derivação desse elemento.

Que sequências são fórmulas?

Recordemos a definição indutiva do conjunto F_P

Linguagem proposicional induzida por Alf_P

A *linguagem proposicional induzida por Alf_P*, denotada F_P, é o conjunto definido indutivamente pelas seguintes regras:

- *BOT*: $\bot \in F_P$
- *PROP*: se $p \in P$ então $p \in F_P$
- *NEG*: se $\varphi \in F_P$ então $\neg\varphi \in F_P$
- *DIS*: se $\varphi, \psi \in F_P$ então $(\varphi \vee \psi) \in F_P$
- *CON*: se $\varphi, \psi \in F_P$ então $(\varphi \wedge \psi) \in F_P$
- *IMP*: se $\varphi, \psi \in F_P$ então $(\varphi \to \psi) \in F_P$

Que sequências são fórmulas?

Exemplo de derivação

Sejam $p, q, r \in P$.

Provar que $(\neg p \to (q \wedge (r \vee p))) \in F_P$ usando árvores:

$$\dfrac{\dfrac{\dfrac{}{p}\,(\text{PROP})}{\neg p}\,(\text{NEG}) \qquad \dfrac{\dfrac{}{q}\,(\text{PROP}) \qquad \dfrac{\dfrac{}{r}\,(\text{PROP}) \qquad \dfrac{}{p}\,(\text{PROP})}{r \vee p}\,(\text{DIS})}{(q \wedge (r \vee p))}\,(\text{CON})}{(\neg p \to (q \wedge (r \vee p)))}\,(\text{IMP})$$

- Para simplificar notação, omitimos "$\in F_P$" em cada nó
- As provas não são (necessariamente) únicas
- Podemos construir top-down ou bottom-up

Que sequências são fórmulas?

Exemplo de derivação

Árvore de derivação para mostrar que $(p \to \neg\neg q) \in F_P$?

Funções sobre conjuntos definidos indutivamente

Definir funções sobre um conjunto definido indutivamente

- Valor da função para os elementos básicos - axiomas
- Valor da função para elementos complexos à custa do valor de elementos mais simples - regras

Exemplo: conjunto das subfórmulas de uma dada fórmula

A função $subF : F_P \to \mathcal{P}(F_P)$ que a cada $\varphi \in F_P$ associa o conjunto das suas subfórmulas é definida da seguinte forma:

- se φ é $\perp$ ou $p \in P$ então $\mathrm{subF}(\varphi) = \{\varphi\}$ - axioma
- se φ é da forma $\neg\delta$ então:

  $\mathrm{subF}(\varphi) = \{\varphi\} \cup \mathrm{subF}(\delta)$ - regra
- se φ é da forma $(\delta \vee \psi)$ ou $(\delta \wedge \psi)$ ou $(\delta \to \psi)$ então:

  $\mathrm{subF}(\varphi) = \{\varphi\} \cup \mathrm{subF}(\delta) \cup \mathrm{subF}(\psi)$ - regra

Funções sobre conjuntos definidos indutivamente

Exemplo de cálculo: símbolos proposicionais de uma fórmula

Sejam $p, q, r \in P$

- $\text{subF}(\neg p) = \{\neg p\} \cup \text{subF}(p) = \{\neg p\} \cup \{p\} = \{\neg p, p\}$;
- $\text{subF}((p \vee q) \to r) = \{(p \vee q) \to r\} \cup \text{subF}(p \vee q) \cup \text{subF}(r) = \{(p \vee q) \to r\} \cup \{p \vee q\} \cup \text{subF}(p) \cup \text{subF}(q) \cup \{r\} = \{(p \vee q) \to r\} \cup \{p \vee q\} \cup \{p\} \cup \{q\} \cup \{r\} = \{(p \vee q) \to r, p \vee q, p, q, r\}$.

Símbolos proposicionais de uma fórmula

O conjunto $\text{sProp}(\varphi)$ é definido por $\text{subF}(\varphi) \cap P$ e contém os símbolos proposicionais de uma fórmula $\varphi \in F_P$.

Exemplo de cálculo de símbolos proposicionais de fórmula

- $\text{sProp}(\neg p) = \text{subF}(\neg p) \cap P = \{p\}$
- $\text{sProp}((p \vee q) \to r) = \text{subF}((p \vee q) \to r) \cap P = \{p, q, r\}$

Funções sobre conjuntos definidos indutivamente

Exercício

Definir indutivamente a função $sProp : F_P \to \mathcal{P}(P)$ que a cada fórmula associa o conjunto dos símbolos proposicionais que ocorrem na fórmula.

Mais um exemplo

A função $size : PilhaInt \to \mathbb{N}_0$ que associa a cada pilha de inteiros o seu tamanho.

- $size(vazia) = 0$
- $size(push(n, p)) = 1 + size(p)$

Qual o valor de $size(push(4, push(-7, push(-1, vazia))))$?

Funções sobre conjuntos definidos indutivamente

Um exercício

A função *soma* : $PilhaInt \to \mathbb{Z}$ que associa a cada pilha de inteiros a soma dos seus elementos.

- $soma(vazia) =?$
- $soma(push(n, p)) =?$

O princípio de indução estrutural

Motivação

- Seja S um conjunto definido indutivamente
- Qualquer elemento de S é obtido usando uma regra de um operador n-ário op, que tem a seguinte forma geral:
- Se para qualquer i tal que $1 \leq i \leq n$, com $n \geq 0$,
  se $e_i \in S_i$, então $\mathrm{op}(e_1, \ldots, e_n) \in S$,
  onde cada S_i ou é o conjunto S ou é outro conjunto já previamente definido.

- Tal como se faz indução sobre os naturais, pode-se fazer indução sobre (os construtores de) qualquer conjunto definido indutivamente.
- A indução natural é um caso particular da indução *estrutural*.

O princípio de indução estrutural

Definição

Seja S definido indutivamente e P predicado sobre elementos de S.

- Se para cada construtor op da definição indutiva de S conseguirmos mostrar que:
- $P(\text{op}(e_1, \ldots, e_k))$ é verdade sempre que $P(e_1), \ldots, P(e_k)$ são todos verdade

então

- conseguimos mostrar que $P(e)$ é verdade para qualquer $e \in S$.

Intuição

Para provarmos a propriedade P sobre um conjunto S temos que:

- Mostrar que os elementos básicos de S satisfazem P
- Mostrar que cada operador das regras de S preserva P

Provas por indução estrutural: um exemplo

Prove, por indução estrutural, que toda a fórmula da Lógica Proposicional tem um número finito de símbolos proposicionais.

F_P o conjunto das fórmulas da Lógica Proposicional
Propriedade a provar: sProp(φ) *é finito*

Prova por indução estrutural

- Base:
  - $\varphi = p$, para $p \in P$.
    Pela definição, sProp(p) $= \{p\}$, que é finito.
  - $\varphi = \perp$. Pela definição, sProp($\perp$) $= \emptyset$, que é finito.
- Passo: (apenas para $\varphi = \psi_1 \wedge \psi_2$; outros casos são semelhantes).
  Temos que sProp(φ) $=$ sProp(ψ_1) $\cup$ sProp(ψ_2), que é finito, pois por hipótese de indução sProp(ψ_1) e sProp(ψ_2) são finitos, e a união de dois conjuntos finitos é um conjunto finito.

Definição indutiva VS prova de pertença

Intuitivamente indicámos que podemos mostrar que um elemento pertence a um conjunto definido indutivamente usando uma prova.

Mas há duas questões importantes:

- Porque é que a existência de tal prova garante que o elemento pertence ao conjunto?
- Será que há uma prova para cada elemento do conjunto?

Como podemos responder formalmente a estas questões?

Equivalência entre definição indutiva e existência de prova

Proposição

$\varphi \in F_P$ se e só se existe uma prova para φ

Prova do sentido "só se": por indução estrutural

Hipótese: $\varphi \in F_P$. Tese: existe uma prova para φ.

Base: $\varphi = \bot$ ou $\varphi = p$ para $p \in P$. Por *BOT* ou por *PROP*, a sequência que tem φ como único elemento é uma prova.

Passo: seja $\varphi = \psi_1 \vee \psi_2$ (os restantes casos têm prova semelhante).
Por hipótese de indução (aplicada a ψ_1 e ψ_2), existem sequências $\psi_{11} \cdots \psi_{1n}$ e $\psi_{21} \cdots \psi_{2m}$ (com $n, m \geq 1$), que provam respectivamente ψ_1 e ψ_2.
Logo, como $\psi_1 = \psi_{1n}$ e $\psi_2 = \psi_{2m}$, por *DIS* a estes passos, a sequência $\psi_{11} \cdots \psi_{1n}\psi_{21} \cdots \psi_{2m}\varphi$ é uma prova para φ.

Equivalência entre a definição indutiva e a existência de prova

Prova do sentido "se": por indução no comprimento da prova

Hipótese: existe uma prova para φ. Tese: termo $\varphi \in F_P$.

Base: a prova é a sequência σ com comprimento 1. Então, a sequência tem apenas uma fórmula ($\sigma = \varphi$), e logo, $\varphi = \perp$ ou $\varphi = p$ para $p \in P$. Por *BOT* ou por *PROP* (respectivamente), a fórmula $\varphi \in F_P$.

Passo: A prova é uma sequência σ de comprimento $1 + n$, sendo $\varphi = \psi_1 \vee \psi_2$ a última fórmula da sequência (os restantes casos têm prova semelhante). Então, a regra *DIS* foi aplicada a ψ_1 e ψ_2, que ocorreram antes na sequência. Logo tanto ψ_1 como ψ_2 têm provas de comprimento inferior a n.
Por hipótese de indução $\psi_1 \in F_P$ e $\psi_2 \in F_P$. Logo, por *DIS*, $\varphi \in F_P$.

Provas por indução estrutural: um exercício

Prove, por indução estrutural, que toda a fórmula da Lógica Proposicional tem um número par de parêntesis.