# Lógica Computacional

## Aula Teórica 10: Algoritmo de Horn

António Ravara    Simão Melo de Sousa

Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

Departamento de Informática, Faculdade Engenharia, LISP & Release Group Universidade Beira Interior

# Como determinar a natureza de uma fórmula na FNC?

## Custo computacional

- Transformar fórmulas arbitrárias para a FNC é moroso.
- Se FNC($\varphi$) então verificar que $\models \varphi$ é simples: demora no máximo um tempo proporcional ao número de símbolos proposicionais da fórmula.
- Determinar se dada fórmula (não válida) é possível ou contraditória requer mais trabalho:
  - é contraditória se uma disjunção for um literal e outra a sua negação (implica análise combinatória);
  - é possível se todas as disjunções forem possíveis e a fórmula não for contraditória.

Só no caso da validade a análise é eficiente. Mas há partida não sabemos a natureza da fórmula dada... Vamos ver um algoritmo eficiente, mas que só funciona para uma dada classe de fórmulas.

# Fórmulas de Horn básicas

## Terminologia

- Uma fórmula atómica ($p$, com $p \in P$, ou $\perp$) diz-se um literal positivo.
- A negação de um literal positivo ($\neg p$, com $p \in P$, ou $\top$) diz-se um literal negativo.

## Definição

Uma fórmula de Horn básica é uma disjunção de literais (com no máximo um a ocorrer positivamente).

## Exemplos

- $\perp$, $p$, $p \vee \neg q$, $\neg p \vee \neg q$ são fórmulas de Horn básicas;
- $p \vee q$ ou $\perp \vee p$ não são fórmulas de Horn básicas.

# Fórmulas de Horn básicas

## Há 3 casos de fórmulas de Horn básicas

- Sem literais negativos (sendo então apenas uma fórmula atómica).
- Sem literais positivos.
- Com literais negativos e um positivo.

## Lema 10.1

Seja $L$ um literal positivo.

1. $L \equiv \top \to L$
2. $\bigvee_{i=1}^{n} \neg L_i \equiv (\bigwedge_{i=1}^{n} L_i) \to \bot$
3. $\bigvee_{i=1}^{n} \neg L_i \vee L \equiv (\bigwedge_{i=1}^{n} L_i) \to L$

## Provas

Por via axiomática (exercícios simples).

# Fórmulas de Horn

## Definição 10.1: conjunto $E_P$ das fórmulas proposicionais

O conjunto $E_P$ é definido pelas regras que definem o conjunto $F_P$ e pela regra de fecho para a negação (considerando tal operador parte do alfabeto proposicional).

## Definição 10.2: Fórmula de Horn

Uma fórmula $\varphi \in E_P$ tal que $\mathrm{FNC}(\varphi)$ é uma fórmula de Horn, se cada disjunção tem no máximo um literal positivo.

## Note-se que a conjunção de:

- fórmulas de Horn básicas é uma fórmula de Horn;
- fórmulas de Horn é uma fórmula de Horn, mas a sua disjunção não.

# Linguagem de Horn

Relembra-se que $\top \stackrel{\text{abv}}{=} \neg\bot$ (sendo então um literal negativo).

## Definição 10.3: Forma de Horn

Se $\varphi \in E_P$ então, para $n \geq 1$, tem-se $\varphi = \bigwedge_{i=1}^{n}(C_i \rightarrow L_i)$, sendo para todo o $i \in \{1, \ldots, n\}$, $C_i = \top$ ou $C_i = \bigwedge_{j=1}^{k_i} L_{i,j}$, com $k_i \geq 1$ e onde cada $L_{i,j}$ é um literal positivo.

## Notação

Considera-se $\{C_i\} = \{L_{i,j} \mid \forall\, j \in \{1, \ldots, k_i\}\}$.

# Algoritmo de Horn

Definição: seja $\mathcal{H}(\varphi) : E_P \to \{0, 1\}$ a seguinte função recursiva

Seja $\varphi = \bigwedge_{j=1}^{n}(C_j \to A_j)$, com $n \geq 1$.

$$\mathcal{H}(\varphi) = \begin{cases} 1, & \text{se } \perp \notin \mathcal{A}(\varphi, \{\top\}) \\ 0, & \text{caso contrário} \end{cases}$$

sendo $\mathcal{A} : E_P \times (\{\perp, \top\} \cup P) \to (\{\perp, \top\} \cup P)$ a seguinte função:

$$\mathcal{A}(\varphi, \mathcal{C}) = \begin{cases} \mathcal{A}(\varphi \setminus (C_i \to A_i), \mathcal{C} \cup \{A_i\}), & \text{se } \exists_{i \in \{1,\ldots,n\}}.\{C_i\} \subseteq \mathcal{C} \\ \mathcal{C}, & \text{caso contrário ou se } \varphi \equiv \top \end{cases}$$

onde

- $\varphi \setminus (C_i \to A_i) \stackrel{\text{def}}{=} (\bigwedge_{j=1}^{i-1}(C_j \to A_j)) \wedge (\bigwedge_{j=i+1}^{n}(C_j \to A_j))$, se $i > 1$;
- $\varphi \setminus \varphi \stackrel{\text{def}}{=} \top$

# Correcção do algoritmo $\mathcal{H}$

## Objectivo do algoritmo

A função $\mathcal{H} : E_P \to \{0, 1\}$ determina se dada fórmula de Horn é contraditória ou possível.

## Teorema 10.1: correcção e completude de $\mathcal{H}$

Dada uma forma de Horn $\varphi \in E_P$, tem-se que:

- $\mathcal{H}(\varphi) = 1$ se e só se $\varphi$ é possível;
- $\mathcal{H}(\varphi) = 0$ se e só se $\varphi$ é contraditória.

# Resultados sobre o algoritmo $\mathcal{A}$

Se o algoritmo diz que a fórmula é possível, então atribuindo 1 aos símbolos da fórmula que ocorrem no output de $\mathcal{A}$ e 0 aos que não ocorrem, obtém-se uma valoração que satisfaz a fórmula.

## Proposição 10.1

Se $\mathcal{A}(\varphi, \{\top\}) = \mathcal{C}$ e $\perp \notin \mathcal{C}$ então tomando $V$ tal que $V(p) = 1$ para cada $p \in \mathcal{C}$ e $V(q) = 0$ para cada $q \in (\mathcal{C} \setminus P) \cap \mathrm{SMB}(\varphi)$, tem-se que $V \Vdash \varphi$.

## Lema 10.2

A função $\mathcal{A}$ é monótona crescente.

Logo, se no cálculo da função se coloca $\perp$ no conjunto, pode-se parar esse cálculo (pois $\perp$ estará no conjunto final).

# Resultados sobre o algoritmo $\mathcal{A}$

O algoritmo só devolve literais que ocorrem na fórmula.
Seja $\varphi$ uma forma de Horn, *i.e.*, $\varphi = \bigwedge_{i=1}^{n}(C_i \to L_i)$.

Lema 10.3: literais omissos

$$\{\top\} \subseteq \mathcal{A}(\varphi, \{\top\}) \subseteq \{\top\} \cup \bigcup_{i=1}^{n} L_i$$

Se nenhum dos $L_i$ é o $\perp$ ou dos $C_i$ é o $\top$, então $\varphi$ é possível.

Corolário 10.1

1. Se $\perp \notin \bigcup_{i=1}^{n} L_i$ então $\perp \notin \mathcal{A}(\varphi, \{\top\})$.
2. Se $\top \notin \bigcup_{i=1}^{n}\{C_i\}$ então $\mathcal{A}(\varphi, \{\top\}) = \{\top\}$

# Utilização do algoritmo para determinar validade

O algoritmo de Horn pode ser usado para *automaticamente* verificar validade

- Pelo Lema 5.2.1, a negação de uma fórmula válida é contraditória (e vice-versa).
- Seja $\psi \stackrel{\text{def}}{=} \mathcal{T}(\neg\varphi)$. Se $\psi$ é uma forma de Horn, então calcula-se $\mathcal{H}(\psi)$. Se o resultado for 0, então $\varphi$ é válida pois $\psi$ é contraditória.

## Utilização do algoritmo para determinar consequências semânticas

O algoritmo de Horn pode também ser usado para *automaticamente* verificar consequências semânticas

- Pelo Teorema 6.1,
  $\{\varphi_1, \ldots, \varphi_n\} \models \varphi$ se e só se $\models (\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow \varphi$.
- Seja $\psi \stackrel{\text{def}}{=} \mathcal{T}(\neg((\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow \varphi))$. Se $\psi$ é uma forma de Horn, então calcula-se $\mathcal{H}(\psi)$. Se o resultado for 0, então $(\varphi_1 \wedge \ldots \wedge \varphi_n) \rightarrow \varphi$ é válida pois $\psi$ é contraditória. Logo, verifica-se $\{\varphi_1, \ldots, \varphi_n\} \models \varphi$.

# Aplicação do algoritmo

## Exemplo

- Qual a natureza de
  $p \wedge (\neg r \vee s) \wedge (r \vee \neg p \vee \neg q) \wedge (\neg r \vee \neg s) \wedge q$ ?
- A fórmula está na FNC;
- A fórmula não é válida (pelo Lema da validade das disjunções);
- Como está na forma de Horn, aplica-se o algoritmo.

# Emulação do algoritmo

## Natureza de $p \wedge (\neg r \vee s) \wedge (r \vee \neg p \vee \neg q) \wedge (\neg r \vee \neg s) \wedge q$

Converte-se primeiro para a forma de Horn. Seja
$\varphi = (\top \to p) \wedge (r \to s) \wedge ((p \wedge q) \to r) \wedge ((r \wedge s) \to \bot) \wedge (\top \to q)$
Aplica-se agora o algoritmo. Sejam
$\varphi_1 = (r \to s) \wedge ((p \wedge q) \to r) \wedge ((r \wedge s) \to \bot) \wedge (\top \to q)$,
$\varphi_2 = (r \to s) \wedge ((p \wedge q) \to r) \wedge ((r \wedge s) \to \bot)$.

$$
\begin{array}{rcl}
\mathcal{A}(\varphi, \{\top\}) & = & \mathcal{A}(\varphi_1, \{\top, p\}) \\
& = & \mathcal{A}(\varphi_2, \{\top, p, q\}) \\
& = & \mathcal{A}((r \to s) \wedge ((r \wedge s) \to \bot), \{\top, p, q, r\}) \\
& = & \mathcal{A}((r \wedge s) \to \bot, \{\top, p, q, r, s\}) \\
& = & \mathcal{A}(\top, \{\top, p, q, r, s, \bot\}) \\
& = & \{\top, p, q, r, s, \bot\} \stackrel{\text{def}}{=} \mathcal{C}
\end{array}
$$

Como $\bot \in \mathcal{C}$, $\mathcal{H}(\varphi) = 0$; pelo Teorema 10.1, $\varphi$ é contraditória.

# Emulação do algoritmo

## Natureza de $p \wedge (\neg r \vee s) \wedge (r \vee \neg p \vee \neg q) \wedge (\neg r \vee \neg s)$

Converte-se primeiro para a forma de Horn.
Seja $\varphi = (\top \to p) \wedge (r \to s) \wedge ((p \wedge q) \to r) \wedge ((r \wedge s) \to \bot)$
Aplica-se agora o algoritmo. Seja
$\psi = (r \to s) \wedge ((p \wedge q) \to r) \wedge ((r \wedge s) \to \bot))$.

$$\begin{aligned} \mathcal{A}(\varphi, \{\top\}) &= \\ \mathcal{A}(\psi, \{\top, p\}) &= \\ \{\top, p\} & \end{aligned}$$

Como $\bot \notin \{\top, p\}$, então $\mathcal{H}(\varphi) = 1$, logo pelo Teorema 10.1, a fórmula $\varphi$ é possível.
Considere-se a valoração $V$ tal que $V(p) = 1$ e $V(q) = V(r) = V(s) = 0$. Facilmente se verifica que $V \Vdash \varphi$.

# Emulação do algoritmo

## Natureza de $p \wedge (\neg r \vee s) \wedge (r \vee \neg p) \wedge \neg r$

Converte-se primeiro para a forma de Horn.
Seja $\varphi = (\top \to p) \wedge (r \to s) \wedge (p \to r) \wedge (r \to \bot)$,
$\varphi_1 = (r \to s) \wedge (p \to r) \wedge (r \to \bot)$, e
$\varphi_2 = (r \to s) \wedge (r \to \bot)$
Aplica-se agora o algoritmo.

$$\begin{array}{rl} \mathcal{A}(\varphi, \{\top\}) & = \\ \mathcal{A}(\varphi_1, \{\top, p\}) & = \\ \mathcal{A}(\varphi_2, \{\top, p, r\}) & \supseteq \\ \{\top, p, r, \bot\} & \end{array}$$

pois $\mathcal{A}$ é monótona crescente. Como $\bot \in \mathcal{A}(\varphi, \{\top\})$, então $\mathcal{H}(\varphi) = 0$. Logo pelo Teorema 10.1, a fórmula $\varphi$ é contraditória.