

Lógica Computacional

Aula Teórica 10: Algoritmo de conversão para FNC

António Ravara

Departamento de Informática

24 de Março de 2011

Como transformar uma fórmula arbitrária para FNC?

Conjunto das fórmulas proposicionais com negação e sem implicação

Objectivo

Definir um algoritmo para transformar qualquer fórmula de Lógica Proposicional para a Forma Normal Conjuntiva.

Conjuntos de fórmulas proposicionais

- Seja G_P o conjunto das fórmulas proposicionais que se obtém considerando o conectivo de negação ($\neg$) primitivo e definindo a constante $\perp$ como abreviatura ($\perp \stackrel{\text{abv}}{=} \varphi \wedge \neg\varphi$, sendo $\varphi \in G_P$).
- Seja $H_P \subseteq G_P$ o conjunto das fórmulas proposicionais que não contêm o conectivo de implicação ($\rightarrow$).

Como transformar uma fórmula arbitrária para FNC?

Algoritmos auxiliares

Eliminação da implicação

Seja $\text{ImplFree}: G_P \rightarrow H_P$ a seguinte função recursiva.

$$\text{ImplFree}(\varphi) = \begin{cases} \neg \text{ImplFree}(\varphi_1), & \text{se } \varphi = \neg \varphi_1 \\ \text{ImplFree}(\varphi_1) \vee \text{ImplFree}(\varphi_2), & \text{se } \varphi = \varphi_1 \vee \varphi_2 \\ \text{ImplFree}(\varphi_1) \wedge \text{ImplFree}(\varphi_2), & \text{se } \varphi = \varphi_1 \wedge \varphi_2 \\ \neg \text{ImplFree}(\varphi_1) \vee \text{ImplFree}(\varphi_2), & \text{se } \varphi = \varphi_1 \rightarrow \varphi_2 \\ \varphi, & \text{caso contrário} \end{cases}$$

Note-se que o caso base da definição recursiva é quando a fórmula φ é um símbolo proposicional.

Como transformar uma fórmula arbitrária para FNC?

Emulação de uma execução de ImplFree

$$\begin{aligned}
 & \text{ImplFree}(((p \vee \neg s) \rightarrow q) \rightarrow (p \rightarrow (r \wedge \neg s))) = \\
 & \neg \text{ImplFree}((p \vee \neg s) \rightarrow q) \vee \text{ImplFree}(p \rightarrow (r \wedge \neg s)) = \\
 & \quad \neg(\neg \text{ImplFree}(p \vee \neg s) \vee \text{ImplFree}(q)) \vee \\
 & \quad (\neg \text{ImplFree}(p) \vee \text{ImplFree}(r \wedge \neg s)) = \\
 & \quad \neg(\neg(\text{ImplFree}(p) \vee \text{ImplFree}(\neg s)) \vee q) \vee \\
 & \quad (\neg p \vee (\text{ImplFree}(r) \wedge \text{ImplFree}(\neg s))) = \\
 & \neg(\neg(p \vee \neg \text{ImplFree}(s)) \vee q) \vee (\neg p \vee (r \wedge \neg \text{ImplFree}(s))) = \\
 & \quad \neg(\neg(p \vee \neg s) \vee q) \vee (\neg p \vee (r \wedge \neg s))
 \end{aligned}$$

Como transformar uma fórmula arbitrária para FNC?

Algoritmos auxiliares

Forma normal da negação

Uma fórmula $\varphi \in H_P$ diz-se que está na forma normal da negação (e escreve-se $FNN(\varphi)$), se só as suas subfórmulas que são fórmulas atómicas estão negadas.

Eliminação das duplas negações

Seja $NNFC: H_P \rightarrow H_P$ a seguinte função recursiva.

$$NNFC(\varphi) = \begin{cases} NNFC(\varphi_1), & \text{se } \varphi = \neg\neg\varphi_1 \\ NNFC(\neg\varphi_1) \vee NNFC(\neg\varphi_2), & \text{se } \varphi = \neg(\varphi_1 \wedge \varphi_2) \\ NNFC(\neg\varphi_1) \wedge NNFC(\neg\varphi_2), & \text{se } \varphi = \neg(\varphi_1 \vee \varphi_2) \\ NNFC(\varphi_1) \vee NNFC(\varphi_2), & \text{se } \varphi = \varphi_1 \vee \varphi_2 \\ NNFC(\varphi_1) \wedge NNFC(\varphi_2), & \text{se } \varphi = \varphi_1 \wedge \varphi_2 \\ \varphi, & \text{caso contrário} \end{cases}$$

Note-se que o caso base da definição recursiva é quando a fórmula φ é um símbolo proposicional ou a sua negação.

Como transformar uma fórmula arbitrária para FNC?

Emulação de uma execução de NNFC

$$\begin{aligned}\text{NNFC}(\neg(\neg(p \vee \neg s) \vee q)) &= \\ \text{NNFC}(\neg\neg(p \vee \neg s)) \wedge \text{NNFC}(\neg q) &= \\ \text{NNFC}((p \vee \neg s)) \wedge \neg q &= \\ (\text{NNFC}(p) \vee \text{NNFC}(\neg s)) \wedge \neg q &= \\ (p \vee \neg s) \wedge \neg q &= \end{aligned}$$

Como transformar uma fórmula arbitrária para FNC?

Algoritmos auxiliares

Função de conversão de fórmulas na FNN para a FNC

Seja CNFC: $H_P \rightarrow H_P$ a seguinte função recursiva.

$$\text{CNFC}(\varphi) = \begin{cases} \text{Distr}(\text{CNFC}(\varphi_1), \text{CNFC}(\varphi_2)), & \text{se } \varphi = \varphi_1 \vee \varphi_2 \\ \text{CNFC}(\varphi_1) \wedge \text{CNFC}(\varphi_2), & \text{se } \varphi = \varphi_1 \wedge \varphi_2 \\ \varphi, & \text{caso contrário} \end{cases}$$

sendo Distr: $H_P \times H_P \rightarrow H_P$ a seguinte função.

$$\text{Distr}(\varphi_1, \varphi_2) = \begin{cases} \text{Distr}(\varphi_{11}, \varphi_2) \wedge \text{Distr}(\varphi_{12}, \varphi_2), & \text{se } \varphi_1 = \varphi_{11} \wedge \varphi_{12} \\ \text{Distr}(\varphi_1, \varphi_{21}) \wedge \text{Distr}(\varphi_1, \varphi_{22}), & \text{se } \varphi_2 = \varphi_{21} \wedge \varphi_{22} \\ \varphi_1 \vee \varphi_2, & \text{caso contrário} \end{cases}$$

Como transformar uma fórmula arbitrária para FNC?

Emulação de uma execução de CNFC

$$\begin{aligned}
 & \text{CNFC}((p \vee (q \wedge r)) \wedge ((r \wedge s) \vee t)) &= \\
 & \text{CNFC}(p \vee (q \wedge r)) \wedge \text{CNFC}((r \wedge s) \vee t) &= \\
 & \text{Distr}(\text{CNFC}(p), \text{CNFC}(q \wedge r)) \wedge \text{Distr}(\text{CNFC}(r \wedge s), \text{CNFC}(t)) &= \\
 & \text{Distr}(p, \text{CNFC}(q) \wedge \text{CNFC}(r)) \wedge \text{Distr}(\text{CNFC}(r) \wedge \text{CNFC}(s), t) &= \\
 & \text{Distr}(p, q \wedge r) \wedge \text{Distr}(r \wedge s, t) &= \\
 & \text{Distr}(p, q) \wedge \text{Distr}(p, r) \wedge \text{Distr}(r, t) \wedge \text{Distr}(s, t) &= \\
 & (p \vee q) \wedge (p \vee r) \wedge (r \vee t) \wedge (s \vee t)
 \end{aligned}$$

Algoritmo $\mathcal{T}$

O algoritmo seguinte converte fórmulas proposicionais para a FNC.

Definição: seja $\mathcal{T} : G_P \rightarrow H_P$ a seguinte função.

$$\mathcal{T}(\varphi) = \text{CNFC}(\text{NNFC}(\text{ImplFree}(\varphi)))$$

Teorema: o algoritmo $\mathcal{T}$ é correcto

Dado $\varphi \in G_P$, obtém-se $\psi \in H_P$ fazendo $\psi = \mathcal{T}(\varphi)$, sendo $\text{FNC}(\psi)$ e $\varphi \equiv \psi$.

Lemas de optimização

- 1 Se $\varphi \in H_P$ então $\text{ImplFree}(\varphi) = \varphi$.
- 2 Se $\text{FNN}(\varphi)$ então $\text{NNFC}(\varphi) = \varphi$.
- 3 Se $\text{FNC}(\varphi)$ então $\text{CNFC}(\varphi) = \varphi$.

Análise dos algoritmos de conversão

Lema

- 1 Dada $\varphi \in G_P$, a fórmula $\psi = \text{ImplFree}(\varphi)$ é tal que $\psi \in H_P$ e $\varphi \equiv \psi$.
- 2 Dada $\varphi \in H_P$, a fórmula $\psi = \text{NNFC}(\varphi)$ é tal que $\psi \in H_P$, $\varphi \equiv \psi$ e $\text{FNN}(\psi)$.
- 3 Dadas $\varphi_1, \varphi_2 \in H_P$ tais que $\text{FNN}(\varphi_1)$ e $\text{FNN}(\varphi_2)$, a fórmula $\psi = \text{Distr}(\varphi_1, \varphi_2)$ é tal que $\psi \in H_P$ e $\text{FNN}(\psi)$, e se $\text{FNC}(\varphi_1)$ e $\text{FNC}(\varphi_2)$ também $\text{FNC}(\psi)$.
- 4 Dada $\varphi \in H_P$ tal que $\text{FNN}(\varphi)$, a fórmula $\psi = \text{CNFC}(\varphi)$ é tal que $\psi \in H_P$, $\varphi \equiv \psi$ e $\text{FNC}(\psi)$.

Provas

Por indução na definição das funções.

Correcção do algoritmo $\mathcal{T}$

Teorema: o algoritmo $\mathcal{T}$ é correcto

Dado $\varphi \in G_P$, obtém-se $\psi \in H_P$ fazendo $\psi = \mathcal{T}(\varphi)$, sendo $\text{FNC}(\psi)$ e $\varphi \equiv \psi$.

Prova

Por indução estrutural, compondo por ordem os lemas anteriores. Note-se que todos os algoritmos terminam.

Determinar a validade de uma fórmula pelo algoritmo CNFC

Aplicação do algoritmo

Como verificar a validade de uma fórmula pelo algoritmo CNFC?

É necessário primeiro converter a fórmula (chame-se-lhe φ) para a Forma Normal Conjuntiva. Obtém-se uma fórmula ψ equivalente a φ que analisamos com o Lema da disjunção de literais.

Verificar φ

- 1 Calcular $\psi = \mathcal{T}(\varphi) = \text{CNFC}(\text{NNFC}(\text{ImplFree}(\varphi)))$.
- 2 A fórmula ψ é válida se cada uma das suas disjunções o for, e cada uma é-o se contiver um literal e a sua negação ou o $\top$.

Aplicação do algoritmo a $\varphi = ((\neg p \wedge q) \rightarrow (p \wedge (r \rightarrow q)))$

Eliminação das implicações

$$\begin{aligned}
 \text{ImplFree}(\varphi) &= \neg \text{ImplFree}(\neg p \wedge q) \vee \text{ImplFree}(p \wedge (r \rightarrow q)) \\
 &= \neg(\neg \text{ImplFree}(p) \wedge \text{ImplFree}(q)) \\
 &\quad \vee (\text{ImplFree}(p) \wedge \text{ImplFree}(r \rightarrow q)) \\
 &= \neg(\neg p \wedge q) \vee (p \wedge (\neg \text{ImplFree}(r) \vee \text{ImplFree}(q))) \\
 &= \neg(\neg p \wedge q) \vee (p \wedge (\neg r \vee q))
 \end{aligned}$$

Eliminação das duplas negações

$$\begin{aligned}
 &\text{NNFC}(\neg(\neg p \wedge q) \vee (p \wedge (\neg r \vee q))) = \\
 &\text{NNFC}(\neg(\neg p \wedge q)) \vee \text{NNFC}(p \wedge (\neg r \vee q)) = \\
 &(\text{NNFC}(\neg\neg p) \vee \text{NNFC}(\neg q)) \vee (\text{NNFC}(p) \wedge \text{NNFC}(\neg r \vee q)) = \\
 &(\text{NNFC}(p) \vee \neg \text{NNFC}(q)) \vee (p \wedge (\text{NNFC}(\neg r) \vee \text{NNFC}(q))) = \\
 &\quad (p \vee \neg q) \vee (p \wedge (\neg r \vee q))
 \end{aligned}$$

Aplicação do algoritmo a $\varphi = ((\neg p \wedge q) \rightarrow (p \wedge (r \rightarrow q)))$

Distribuição da disjunção sobre a conjunção

$$\begin{aligned} \text{CNFC}((p \vee \neg q) \vee (p \wedge (\neg r \vee q))) &= \\ \text{Distr}((p \vee \neg q), (p \wedge (\neg r \vee q))) &= \\ \text{Distr}(p \vee \neg q, p) \wedge \text{Distr}(p \vee \neg q, \neg r \vee q) &= \\ (p \vee \neg q \vee p) \wedge (p \vee \neg q \vee \neg r \vee q) \end{aligned}$$

$\models (p \vee \neg q \vee p) \wedge (p \vee \neg q \vee \neg r \vee q)$?

Não.

- ① $\models p \vee \neg q \vee \neg r \vee q$, pois contém q e $\neg q$, mas
- ② $\not\models p \vee \neg q \vee p$, pois fazendo $V(p) = 0$ e $V(q) = 1$ tem-se que $V \not\models p \vee \neg q \vee p$

A fórmula não é válida mas é possível, porque para $V'(p) = 0$ e $V'(q) = 0$ tem-se que $V' \models p \vee \neg q \vee p$.