Lógica Computacional

Aula Teórica 9: Algoritmo de conversão para FNC

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

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 $\bot$ como abreviatura ($\bot \stackrel{\text{abv}}{=} \varphi \wedge \neg\varphi$, sendo $\varphi \in G_P$).
- Seja $H_P \subseteq G_P$ o conjunto das fórmulas proposicionais que não contêm o conectivo de implicação ($\rightarrow$).

Algoritmos auxiliares

Eliminação da implicação

Seja 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.

Emulação de uma execução de ImplFree

$$
\begin{aligned}
\mathsf{ImplFree}(((p \vee \neg s) \to q) \to (p \to (r \wedge \neg s))) &= \\
\neg\,\mathsf{ImplFree}((p \vee \neg s) \to q) \vee \mathsf{ImplFree}(p \to (r \wedge \neg s)) &= \\
\neg(\neg\,\mathsf{ImplFree}(p \vee \neg s) \vee \mathsf{ImplFree}(q)) \vee & \\
(\neg\,\mathsf{ImplFree}(p) \vee \mathsf{ImplFree}(r \wedge \neg s)) &= \\
\neg(\neg(\mathsf{ImplFree}(p) \vee \mathsf{ImplFree}(\neg s)) \vee q) \vee & \\
(\neg p \vee (\mathsf{ImplFree}(r) \wedge \mathsf{ImplFree}(\neg s))) &= \\
\neg(\neg(p \vee \neg\,\mathsf{ImplFree}(s)) \vee q) \vee (\neg p \vee (r \wedge \neg\,\mathsf{ImplFree}(s))) &= \\
\neg(\neg(p \vee \neg s) \vee q) \vee (\neg p \vee (r \wedge \neg s)) &
\end{aligned}
$$

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(φ)), 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.

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

Note-se que o caso base da definição recursiva é quando a fórmula φ é um símbolo proposicional ou a sua negação (*i.e.*, um literal).

Emulação de uma execução de NNFC

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

Algoritmos auxiliares

Função de conversão de fórmulas na FNN para a FNC

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

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

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

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

Emulação de uma execução de CNFC

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

Algoritmo $\mathcal{T}$

O algoritmo seguinte converte fórmulas proposicionais para a FNC.

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

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

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

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

Lemas de optimização

1. Se $\varphi \in H_P$ então ImplFree(φ)=φ.
2. Se FNN(φ) então NNFC(φ)=φ.
3. Se FNC(φ) então CNFC(φ)=φ.

Análise dos algoritmos de conversão

Lema

1. Dada $\varphi \in G_P$, a fórmula $\psi = \mathrm{ImplFree}(\varphi)$ é tal que $\psi \in H_P$ e $\varphi \equiv \psi$.
2. Dada $\varphi \in H_P$, a fórmula $\psi = \mathrm{NNFC}(\varphi)$ é tal que $\psi \in H_P$, $\varphi \equiv \psi$ e FNN(ψ).
3. Dadas $\varphi_1, \varphi_2 \in H_P$ tais que $\mathrm{FNN}(\varphi_1)$ e $\mathrm{FNN}(\varphi_2)$, a fórmula $\psi = \mathrm{Distr}(\varphi_1, \varphi_2)$ é tal que $\psi \in H_P$ e FNN(ψ), e se FNC(φ_1) e FNC(φ_2) também FNC(ψ).
4. Dada $\varphi \in H_P$ tal que FNN(φ), a fórmula $\psi = \mathrm{CNFC}(\varphi)$ é tal que $\psi \in H_P$, $\varphi \equiv \psi$ e FNC(ψ).

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 FNC($\psi$) e $\varphi \equiv \psi$.

Prova

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

Como verificar a validade de uma fórmula pelo algoritmo $\mathcal{T}$?

1. Converte-se a fórmula para FNC:

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

2. Analisa-se ψ com o Lema da disjunção de literais.

Aplicação do algoritmo

Seja $\varphi \in G_P$ e $\psi = \mathcal{T}(\varphi)$; note-se que $\psi \in H_P$.

Análise de ψ

- É *válida*, se cada uma das suas disjunções o for, e cada uma é-o se contiver um literal e a sua negação (não ocorre em ψ o $\top$ porque $\psi \in H_P$).
- É contraditória se uma das disjunções for um literal e outra a sua negação (não ocorre em ψ o $\bot$ porque $\psi \in H_P$).
- É *possível*, se nenhuma das situações anteriores se verificar (*i.e.*, não é contraditória nem válida - alguma das suas disjunções é possível).

Considere-se agora também $\Phi \subseteq G_P$.

Como verificar se $\Phi \models \varphi$?

Uma vez que, pelo Teorema 6.1, $\Phi \models \varphi$ se e só se $\models \Phi \rightarrow \varphi$, analisa-se $\mathcal{T}(\Phi \rightarrow \varphi)$.

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

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

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

Distribuição da disjunção sobre a conjunção

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

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

Não. Pelo Lema da disjunção,

1. $\models p \vee \neg q \vee \neg r \vee q$, pois contém q e $\neg q$, mas
2. $\not\models p \vee \neg q \vee p$, por contra-recíproco.

A fórmula não é válida mas é possível, porque fazendo $V(p) = 0$ e $V(q) = 1$ tem-se que $V \not\Vdash p \vee \neg q \vee p$, mas para $V'(p) = 0$ e $V'(q) = 0$ tem-se que $V' \Vdash p \vee \neg q \vee p$.