Prenexing rules

Prenexing rules:

$$\exists\!\!\!\forall p A_1 \between \ldots \between A_n \;\Rightarrow\; \exists\!\!\!\forall p (A_1 \between \ldots \between A_n)$$

$$\forall p A_1 \rightarrow A_2 \;\Rightarrow\; \exists p (A_1 \rightarrow A_2) \quad \exists p A_1 \rightarrow A_2 \;\Rightarrow\; \forall p (A_1 \rightarrow A_2)$$

$$A_1 \rightarrow \forall p A_2 \;\Rightarrow\; \forall p (A_1 \rightarrow A_2) \quad A_1 \rightarrow \exists p A_2 \;\Rightarrow\; \exists p (A_1 \rightarrow A_2)$$

$$\neg \forall p A \;\Rightarrow\; \exists p \neg A \qquad \neg \exists p A \;\Rightarrow\; \forall p \neg A$$

Standard CNF transformation

$$
\begin{array}{rcll}
A \leftrightarrow B & \Rightarrow & (\neg A \vee B) \wedge (\neg B \vee A), & \\
A \rightarrow B & \Rightarrow & \neg A \vee B, & \\
\neg(A \wedge B) & \Rightarrow & \neg A \vee \neg B, & \\
\neg(A \vee B) & \Rightarrow & \neg A \wedge \neg B, & \\
\neg\neg A & \Rightarrow & A, & \\
(A_1 \wedge \ldots \wedge A_m) \vee B_1 \vee \ldots \vee B_n & \Rightarrow & (A_1 \vee B_1 \vee \ldots \vee B_n) & \wedge \\
 & & \ldots & \wedge \\
 & & (A_m \vee B_1 \vee \ldots \vee B_n). &
\end{array}
$$

Splitting

Lemma (i) A closed formula $\forall p A$ is true if and only if the formulas $A_p^\perp$ and $A_p^\top$ are true. (ii) A closed formula $\exists p A$ is true if and only if at least one of the formulas $A_p^\perp$ and $A_p^\top$ is true.

Splitting

Simplification rules for $\top$:

$$\neg\top \Rightarrow \bot$$

$$\top \wedge A_1 \wedge \ldots \wedge A_n \Rightarrow A_1 \wedge \ldots \wedge A_n$$

$$\top \vee A_1 \vee \ldots \vee A_n \Rightarrow \top$$

$$A \rightarrow \top \Rightarrow \top \qquad \top \rightarrow A \Rightarrow A$$

$$A \leftrightarrow \top \Rightarrow A \qquad \top \leftrightarrow A \Rightarrow A$$

$$\forall p \top \Rightarrow \top$$

$$\exists p \top \Rightarrow \top$$

Simplification rules for $\bot$:

$$\neg\bot \Rightarrow \top$$

$$\bot \wedge A_1 \wedge \ldots \wedge A_n \Rightarrow \bot$$

$$\bot \vee A_1 \vee \ldots \vee A_n \Rightarrow A_1 \vee \ldots \vee A_n$$

$$A \rightarrow \bot \Rightarrow \neg A \qquad \bot \rightarrow A \Rightarrow \top$$

$$A \leftrightarrow \bot \Rightarrow \neg A \qquad \bot \leftrightarrow A \Rightarrow \neg A$$

$$\forall p \bot \Rightarrow \bot$$

$$\exists p \bot \Rightarrow \bot$$

Splitting algorithm

procedure $splitting(G)$

input: closed rectified formula G ; **output**: 0 or 1

parameters: function $select\_literal$

begin

$G \ := \ simplify(G)$

if $G = \bot$ **then** **return** 0 ; **if** $G = \top$ **then** **return** 1

Let $\exists\!\!\!\forall$ be a quantifier and P a set of variables such that $G = \exists\!\!\!\forall P G_1$

$(p, b) \ := \ select\_literal(P, G)$

$G' \ := \ \exists\!\!\!\forall(P - \{p\})G_1$

if $b = 0$ **then** $(B_1, B_2) \ := \ (\bot, \top)$

else $(B_1, B_2) \ := \ (\top, \bot)$

case $(splitting((G')_p^{B_1})$ **of**

$(0, \forall) \Rightarrow$ **return** 0

$(0, \exists) \Rightarrow$ **return** $splitting((G')p^{B_2})$

$(1, \forall) \Rightarrow$ **return** $splitting((G')p^{B_2})$

$(1, \exists) \Rightarrow$ **return** 1

end

Pure literal rule

Lemma Let G be a prenex rectifi ed closed formula
$\exists\!\!\!\forall_1 P_1 \ldots \exists\!\!\!\forall_n P_n \exists\!\!\!\forall p A$.

1. If all occurrences of p in G are positive and $\exists\!\!\!\forall = \exists$, then G is equivalent to $\exists\!\!\!\forall_1 P_1 \ldots \exists\!\!\!\forall_n P_n A_p^{\top}$.
2. If all occurrences of p in G are positive and $\exists\!\!\!\forall = \forall$, then G is equivalent to $\exists\!\!\!\forall_1 P_1 \ldots \exists\!\!\!\forall_n P_n A_p^{\bot}$.
3. If all occurrences of p in G are negative and $\exists\!\!\!\forall = \exists$, then G is equivalent to $\exists\!\!\!\forall_1 P_1 \ldots \exists\!\!\!\forall_n P_n A_p^{\bot}$.
4. If all occurrences of p in G are negative and $\exists\!\!\!\forall = \forall$, then G is equivalent to $\exists\!\!\!\forall_1 P_1 \ldots \exists\!\!\!\forall_n P_n A_p^{\top}$.

DLL algorithm

<u>procedure</u> $DLL(Q, S)$

<u>input</u>: quantifier prefix $Q = \exists\forall_1 P_1 \ldots \exists\forall_n P_n$, set of clauses S

<u>output</u>: 0 or 1

<u>parameters</u>: function $select\_literal$

<u>begin</u>

S := $propagate(S)$

<u>if</u> S is empty <u>then</u> <u>return</u> 1 ; <u>if</u> S contains $\square$ <u>then</u> <u>return</u> 0

L := $select\_literal(\exists\forall_1 P_1, \exists\forall_2 P_2 \ldots \exists\forall_n P_n, S)$

$Q_1 = \exists\forall_1 (P_1 - \{p\}) \exists\forall_2 P_2 \ldots \exists\forall_n P_n$

<u>case</u> $(DLL(Q_1, S \cup \{L\}), \exists\forall_1)$ <u>of</u>

$(0, \forall) \Rightarrow$ <u>return</u> 0

$(0, \exists) \Rightarrow$ <u>return</u> $DLL(Q_1, S \cup \{\tilde{L}\})$

$(1, \forall) \Rightarrow$ <u>return</u> $DLL(Q_1, S \cup \{\tilde{L}\})$

$(1, \exists) \Rightarrow$ <u>return</u> 1

<u>end</u>