

Formale Techniken der Software-Entwicklung
 Übungsblatt 2
 Besprechung am 25.04.2014

Musterlösung

Aufgabe 1:

Transformieren Sie die folgenden Formeln in die konjunktive Normalform (CNF)

(a) $(\neg(A \wedge B)) \wedge (C \implies B) \wedge ((\neg A \wedge B) \implies \neg C) \wedge C$

Lösung:

$$\begin{aligned} \Phi &\equiv (\neg(A \wedge B)) \wedge (C \implies B) \wedge ((\neg A \wedge B) \implies \neg C) \wedge C \\ \text{IMPL_FREE}(\Phi) &\equiv \text{IMPL_FREE}((\neg(A \wedge B)) \wedge (C \implies B) \wedge ((\neg A \wedge B) \implies \neg C) \wedge C) \\ &\equiv \text{IMPL_FREE}(\neg(A \wedge B)) \wedge \text{IMPL_FREE}(C \implies B) \wedge \text{IMPL_FREE}((\neg A \wedge B) \implies \neg C) \wedge \text{IMPL_FREE}(C) \\ &\equiv \neg(A \wedge B) \wedge (\neg C \vee B) \wedge (\neg(\neg A \wedge B) \vee \neg C) \wedge C \\ \text{NNF}(\text{IMPL_FREE}(\Phi)) &\equiv \text{NNF}(\neg(A \wedge B) \wedge (\neg C \vee B) \wedge (\neg(\neg A \wedge B) \vee \neg C) \wedge C) \\ &\equiv \text{NNF}(\neg(A \wedge B)) \wedge \text{NNF}(\neg C \vee B) \wedge \text{NNF}(\neg(\neg A \wedge B) \vee \neg C) \wedge \text{NNF}(C) \\ &\equiv (\text{NNF}(\neg A) \vee \text{NNF}(\neg B)) \wedge (\text{NNF}(\neg C) \vee \text{NNF}(B)) \wedge (\text{NNF}(\neg(\neg A \wedge B)) \vee \text{NNF}(\neg C)) \wedge C \\ &\equiv (\neg A \vee \neg B) \wedge (\neg C \vee B) \wedge ((\text{NNF}(\neg\neg A) \vee \text{NNF}(\neg B)) \vee \neg C) \wedge C \\ &\equiv (\neg A \vee \neg B) \wedge (\neg C \vee B) \wedge ((A \vee \neg B) \vee \neg C) \wedge C \\ \text{CNF}(\text{NNF}(\text{IMPL_FREE}(\Phi))) &\equiv \text{CNF}((\neg A \vee \neg B) \wedge (\neg C \vee B) \wedge ((A \vee \neg B) \vee \neg C) \wedge C) \\ &\equiv (\neg A \vee \neg B) \wedge (\neg C \vee B) \wedge (A \vee \neg B \vee \neg C) \wedge C \end{aligned}$$

(b) $\neg((A \implies B) \wedge (A \wedge B \implies C) \implies (A \implies C))$

Lösung:

$$\begin{aligned} \Phi &= \neg((A \implies B) \wedge (A \wedge B \implies C) \implies (A \implies C)) \\ \text{IMPL_FREE}(\Phi) &\equiv \neg(\neg((\neg A \vee B) \wedge (\neg(A \wedge B) \vee C)) \vee (\neg A \vee C)) \\ \text{NNF}(\text{IMPL_FREE}(\Phi)) &\equiv ((\neg A \vee B) \wedge ((\neg A \vee \neg B) \vee C)) \wedge (A \wedge \neg C) \\ \text{CNF}(\text{NNF}(\text{IMPL_FREE}(\Phi))) &\equiv \text{CNF}((\neg A \vee B) \wedge (\neg A \vee \neg B \vee C)) \wedge (A \wedge \neg C) \end{aligned}$$

$$\begin{aligned}
&\equiv \text{CNF}((\neg A \vee B) \wedge (\neg A \vee \neg B \vee C)) \wedge \text{CNF}(A \wedge \neg C) \\
&\equiv \text{CNF}(\neg A \vee B) \wedge \text{CNF}((\neg A \vee \neg B) \vee C) \wedge \text{CNF}(A) \wedge \text{CNF}(\neg C) \\
&\equiv \text{DISTR}(\text{CNF}(\neg A), \text{CNF}(B)) \wedge \text{DISTR}(\text{CNF}(\neg A \vee \neg B), \text{CNF}(C)) \wedge A \wedge \neg C \\
&\equiv \text{DISTR}(\neg A, B) \wedge \text{DISTR}(\text{DISTR}(\text{CNF}(\neg A), \text{CNF}(\neg B)), C) \wedge A \wedge \neg C \\
&\equiv (\neg A \vee B) \wedge \text{DISTR}(\text{DISTR}(\neg A, \neg B), C) \wedge A \wedge \neg C \\
&\equiv (\neg A \vee B) \wedge \text{DISTR}((\neg A \vee \neg B), C) \wedge A \wedge \neg C \\
&\equiv (\neg A \vee B) \wedge ((\neg A \vee \neg B) \vee C) \wedge A \wedge \neg C \\
&\equiv (\neg A \vee B) \wedge (\neg A \vee \neg B \vee C) \wedge A \wedge \neg C
\end{aligned}$$

(c) $\neg(A \implies (\neg(B \wedge (\neg A \implies B))))$

Lösung:

$$\begin{aligned}
\Phi &\equiv \neg(A \implies (\neg(B \wedge (\neg A \implies B)))) \\
\text{IMPL_FREE}(\Phi) &\equiv \neg(\neg A \vee \neg(B \wedge (\neg \neg A \vee B))) \\
\text{NNF}(\text{IMPL_FREE}(\Phi)) &\equiv A \wedge (B \wedge (A \vee B)) \\
\text{CNF}(\text{NNF}(\text{IMPL_FREE}(\Phi))) &\equiv A \wedge B \wedge (A \vee B)
\end{aligned}$$

Aufgabe 2:

Implementieren Sie einen einfachen SAT-Solver, indem Sie den grundlegenden DPLL-Algorithmus umsetzen. Auf Wikipedia (http://en.wikipedia.org/wiki/DPLL_algorithm) finden Sie dazu Informationen. Implementieren Sie dabei auch eine Routine, die CNF-Probleme im DIMACS-Format einlesen kann. Dieses Format wird standardmäßig von SAT-Solvern verwendet und ist z.B. unter <http://people.sc.fsu.edu/~jburkardt/data/cnf/cnf.html> beschrieben.

Als Test für Ihren SAT-Solver können Sie die Dateien auf der Vorlesungs-Homepage verwenden.

Lösung:

<http://www.pst.ifi.lmu.de/Lehre/sose-2014/ftse/simplesat.zip>

Um das Projekt SimpleSAT zu verwenden, benötigen Sie das JDK 8 und Eclipse 4.3.2 mit Java 8 Extensions (siehe <http://eclipse.org/downloads/index-java8.php>).