J. Hoenicke
A. Nutz

19.01.2016
submit until 26.01.2016, 14:15

Tutorials for Decision Procedures
Exercise sheet 11

Exercise 1: DPLL(T_E)
Consider the following formula

$$\begin{aligned} & f_b(i) \neq f_c(i) \wedge \\ & f_b(j) = v \wedge (i \neq j \rightarrow f_b(i) = f_a(i)) \wedge \\ & f_c(j) = v \wedge (i \neq j \rightarrow f_c(i) = f_a(i)) \end{aligned}$$

(a) Compute the propositional core in CNF.

(b) Run the DPLL(T) algorithm by repeatedly applying the rules from the lecture. Is the formula satisfiable?

Exercise 2: DPLL(T_A)
Use DPLL(T_A) to decide satisfiabilty formula F_6 on slide 260 in the slide set on the array theory (printed below).

$$\begin{aligned} & \lambda \neq j \ \rightarrow \ a[\lambda] = b[\lambda] \ \wedge \ j \neq j \ \rightarrow \ a[j] = b[j] \\ & \wedge \ k \neq j \ \rightarrow \ a[k] = b[k] \ \wedge \ \lambda \neq k \ \rightarrow \ a[\lambda] \neq b[\lambda] \\ & \wedge \ j \neq k \ \rightarrow \ a[j] \neq b[j] \ \wedge \ k \neq k \ \rightarrow \ a[k] \neq b[k] \\ & \wedge \ \lambda \neq j \ \wedge \ \lambda \neq k \end{aligned}$$

Exercise 3: The πVC compiler (pivc)
Install and run the program pivc. You do not need an external solver as the new versions of πVC use a webservice for these tasks. Thus you have to be online to use it. You can find πVC and some samples on `http://cs.stanford.edu/people/jasonaue/pivc/`.

(a) Load the example program `Abs`. Replace the post condition with `true`. Hit "Compile". All verification conditions should now be valid (green).

(b) Check the flag "Generate Runtime Assertions" in the compile menu and add the loop invariants that are needed to prove them.

(c) Now set the post-condition back to

```
forall ix.  (0 <= ix && ix < |rv| -> rv[ix] >= 0)
```

Add the necessary invariants to prove the correctness of the program.