J. Hoenicke
T. Schindler

17.01.2017
submit until 24.01.2017, 14:15

Tutorials for Decision Procedures
Exercise sheet 12

Exercise 1: DPLL(T)

In the last lecture we presented the CDCL algorithm in the form of the six rules Decide, Propagate , Conflict , Explain, Learn, Backtrack.

In the lecture on propositional logic we presented the same algorithm as a functional program (printed below).

Which lines of the functional code correspond to which of the six rules? (There may not always be an exact correspondence, in such cases please add a short explanation.)

```
let rec DPLL =
  let PROP U =
    let ℓ = CHOOSE U ∩ unassigned in
    val[ℓ] := ⊤
    let C = DPLL in
    if (C = satisfiable)
      satisfiable
    else
      val[ℓ] := undef
      if (ℓ̄ ∉ C) C
      else U \ {ℓ} ∪ C \ {ℓ̄}
  if conflictclauses ≠ ∅
    CHOOSE conflictclauses
  else if unitclauses ≠ ∅
    PROP (CHOOSE unitclauses)
  else if coreclauses ≠ ∅
     let ℓ = CHOOSE (⋃coreclauses) ∩ unassigned in
     val[ℓ] := ⊤
     let C = DPLL in
     if (C = satisfiable) satisfiable
     else
        val[ℓ] := undef
        if (ℓ̄ ∉ C) C
        else LEARN C; PROP C
  else satisfiable
```

Exercise 2: DPLL($T_{\mathbb{Q}}$)
Consider the following formula

$$(z \leq 1 \ \rightarrow \ x \leq y) \ \wedge \ y + z \leq x \ \wedge \ z \geq 0 \ \wedge \ (z \geq 1 \ \rightarrow \ x + z \leq y)$$

(a) Compute the propositional core in CNF.

(b) Run the DPLL(T) algorithm by repeatedly applying the rules from the lecture. Use the notation introduced on slides 317 - 320 to record the single steps. Is the formula satisfiable?

Exercise 3: DPLL(T_A)
Use DPLL(T_A) to decide satisfiabilty of formula F_6 on slide 260 in the slide set on the array theory (printed below). Use the notation on slides 317 - 320 as in exercise 2.

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