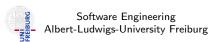
### **Decision Procedures**

#### Jochen Hoenicke



Winter Term 2019/2020



#### **Dates**

- Lecture is Tuesday 14-16 (c.t) and Thursday 14-15 (c.t).
- Tutorials will be given on Thursday 15–16. Starting next week (this week is a two hour lecture).
- Exercise sheets are uploaded on Tuesday.
   They are due on Tuesday the week after.

To successfully participate, you must

- prepare the exercises (at least 50 %)
- actively participate in the tutorial
- pass an oral examination



# THE CALCULUS OF COMPUTATION: Decision Procedures with Applications to Verification

by Aaron Bradley Zohar Manna

Springer 2007



Decision Procedures are algorithms to decide formulae.

These formulae can arise

- in Hoare-style software verification,
- in hardware verification,
- in synthesis,
- in scheduling,
- in planning,
- . . .

## Consider the following program:

How can we prove that the formula is a loop invariant?

# Motivation (3)



Prove the Hoare triples (one for if case, one for else case)

```
assume \ell \leq i \leq u \land (rv \leftrightarrow \exists j. \ \ell \leq j < i \land a[j] = e) assume i \leq u assume a[i] = e rv := true; i := i + 1 \emptyset \ \ell \leq i \leq u \land (rv \leftrightarrow \exists j. \ \ell \leq j < i \land a[j] = e)
```

```
assume \ell \leq i \leq u \land (rv \leftrightarrow \exists j. \ \ell \leq j < i \land a[j] = e) assume i \leq u assume a[i] \neq e i := i + 1 \emptyset \ \ell \leq i \leq u \land (rv \leftrightarrow \exists j. \ \ell \leq j < i \land a[j] = e)
```

A Hoare triple  $\{P\}$  S  $\{Q\}$  holds, iff

$$P \rightarrow wp(S, Q)$$

(wp denotes is weakest precondition)

For assignments wp is computed by substitution:

```
\begin{array}{l} \operatorname{assume} \ \ell \leq i \leq u \wedge (\mathit{rv} \leftrightarrow \exists j. \ \ell \leq j < i \wedge \mathit{a[j]} = e) \\ \operatorname{assume} \ i \leq u \\ \operatorname{assume} \ \mathit{a[i]} = e \\ \mathit{rv} := \mathsf{true}; \\ i := i + 1 \\ \texttt{0} \ \ell \leq i \leq u \wedge (\mathit{rv} \leftrightarrow \exists j. \ \ell \leq j < i \wedge \mathit{a[j]} = e) \end{array}
```

holds if and only if:

$$\ell \leq i \leq u \land (rv \leftrightarrow \exists j. \ \ell \leq j < i \land a[j] = e) \land i \leq u \land a[i] = e$$
  
 $\rightarrow \ell \leq i + 1 \leq u \land (true \leftrightarrow \exists j. \ \ell \leq j < i + 1 \land a[j] = e)$ 

# Motivation (5)

We need an algorithm that decides whether a formula holds.

$$\ell \leq i \leq u \land (rv \leftrightarrow \exists j. \ \ell \leq j < i \land a[j] = e) \land i \leq u \land a[i] = e$$
  
 $\rightarrow \ell \leq i + 1 \leq u \land (true \leftrightarrow \exists j. \ \ell \leq j < i + 1 \land a[j] = e)$ 

If the formula does not hold it should give a counterexample, e.g.:

$$\ell = 0, i = 1, u = 1, rv = false, a[0] = 0, a[1] = 1, e = 1,$$

This counterexample shows that  $i + 1 \le u$  can be violated.

This lecture is about algorithms checking for validity and producing these counterexamples.



## **Topics**



- Propositional Logic
- First-Order Logic
- First-Order Theories
- Quantifier Elimination
- Decision Procedures for Linear Arithmetic
- Decision Procedures for Uninterpreted Functions
- Decision Procedures for Arrays
- Combination of Decision Procedures
- DPLL(T)
- Craig Interpolants