Decision Procedures

Jochen Hoenicke

Software Engineering
Albert-Ludwigs-University Freiburg

Winter Term 2019/2020

Organisation

Organisation

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

Motivation

Motivation

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:

```
for
  @ ℓ ≤ i ≤ u ∧ (rv ↔ ∃j. ℓ ≤ j < i ∧ a[j] = e)
  (int i := ℓ; i ≤ u; i := i + 1) {
  if ((a[i] = e)) {
    rv := true;
  }
}
```

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 ℓ ≤ i ≤ u ∧ (rv ↔ ∃j. ℓ ≤ j < i ∧ a[j] = e)
assume i ≤ u
assume a[i] = e
rv := true;
i := i + 1
@ ℓ ≤ i ≤ u ∧ (rv ↔ ∃j. ℓ ≤ j < i ∧ a[j] = e)
```

```
assume ℓ ≤ i ≤ u ∧ (rv ↔ ∃j. ℓ ≤ j < i ∧ a[j] = e)
assume i ≤ u
assume a[i] ≠ e
i := i + 1
@ ℓ ≤ i ≤ u ∧ (rv ↔ ∃j. ℓ ≤ j < i ∧ a[j] = e)
```

Motivation (4)

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

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

(wp denotes is weakest precondition)
For assignments wp is computed by substitution:

---

assume $\ell \leq i \leq u \land (rv \leftrightarrow \exists j.\ \ell \leq j < i \land a[j] = e)$
assume $i \leq u$
assume $a[i] = e$
$rv := \texttt{true};$
$i := i + 1$
@ $\ell \leq i \leq u \land (rv \leftrightarrow \exists j.\ \ell \leq j < i \land a[j] = e)$

---

holds if and only if:

$$\begin{aligned} &\ell \leq i \leq u \land (rv \leftrightarrow \exists j.\ \ell \leq j < i \land a[j] = e) \land i \leq u \land a[i] = e \\ \to &\ell \leq i + 1 \leq u \land (\texttt{true} \leftrightarrow \exists j.\ \ell \leq j < i + 1 \land a[j] = e) \end{aligned}$$

Motivation (5)

We need an algorithm that decides whether a formula holds.

$$\ell \leq i \leq u \wedge (rv \leftrightarrow \exists j.\, \ell \leq j < i \wedge a[j] = e) \wedge i \leq u \wedge a[i] = e$$
$$\rightarrow \ell \leq i + 1 \leq u \wedge (\texttt{true} \leftrightarrow \exists j.\, \ell \leq j < i + 1 \wedge a[j] = e)$$

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

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

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

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

Contents of Lecture

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