CS 512 Formal Methods, Spring 2018 — Instructor: Assaf Kfoury

Lecture 12: Hoare Logic

March 15, 2018 — Hannah Catabia

(These lecture notes are **not** proofread and proof-checked by the instructor.)

1 A Brief Overview of the Course

Here is a brief list of topics covered in this course:

1. Linear Temporal Logial (LTL)
2. Computation Tree Logic (CTL)
3. CTL*
4. **Hoare Logic ← We are here.**
5. Easy Crypt (applying formal methods to cryptography, lectures by Alley Stoughton)
6. Modal logic
7. Continuation of model checking

We will be spending a couple of weeks on the current topic: Hoare Logic

2 What is Hoare Logic?

It was developed by C.A.R. Hoare in 1967 as a way of tyinr programming languages into propositional and first-order logic. There are a few different subfields of Hoare Logic that have been developed:

- **Probabilistic Hoare Logic (pHL)** - The behavior of the program is probabilistic (as opposed to deterministic).
- **Relational Hoare Logic (RHL)** - Developed into the 2000s to determine if optimized code fulfills the same specifications as the original. It uses Hoare quadruples as opposed to Hoare triples.
- **Probabilistic Relational Hoare Logic (pRHL)** - Combination of the above.

3 Hoare triples

In Hoare Logic, a well-formed formula (WFF) consists of three parts, called a Hoare triple:

$$\{\varphi\}P\{\psi\}$$

The three parts of the triple represent:

- $\{\varphi\}$: the precondition
- P: the program
- $\{\psi\}$: the postcondition

Different sources may use different conventions to name triples, but this is the convention we will be using in CS 512.

It is most informative when precondition φ is as **weak** as possible. In an extreme example, the weakest precondition is `true` (all states satisfy φ) and the strongest precondition is `false` (no states satisfy φ). Also, it is most informative whan postondition ψ is as **strong** as possible.

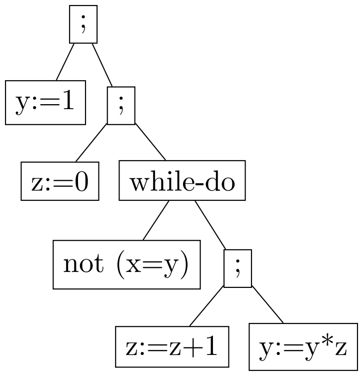
4 Syntax of an imperative programing langauge

Imperitie programming syntax may contain three types of components:

1. integer expressions
2. Boolean expressions
3. program expressions

Here is an example program (also in the handout), as well as its abstract syntax tree:

```
Fact(x):
    y:=1;
    z:=0;
    while not (x=y) do:
        z:=z+1;
        y:=y*z
    od
```



Hoare Logic may be used to determine whether the code above indeed calculates the factorial of the input x. Although this example is simple and we can check it by reading it, that is not always the case for longer and more convoluted programs. This is also complicated by the fact that the meaning of a program is not unique to its particular syntax, and so there may be many ways of programming the same task in a given programming language.

5 PCA vs. TCA

Hoare Logic may be executed with two different correctness assertions:

1. **Partial Correctness Assertion (PCA)**: termination of the program is **not** guaranteed
2. **Total Correctness Assertion (TCA)**: termination is guaranteed

TCA is a stronger condition, but more difficult in practice.