G52DOA - Derivation of Algorithms

Lecture 1

Venanzio Capretta

Nottingham, Thursday 4 February 2010

The correctness of programs

What is the topic of this course?

- Programs written in C, Java, or other imperative languages are often full of bugs;
- Debugging by testing: A finite number of tests can never guarantee absolute correctnes;
- Instead we give a logico/mathematical proof of correctness;
- Specification: Express precisely what the program is supposed to do;
- Verification: Prove formally that the program satisfies the specification;
- Derivation: Start with the specification and build from it a program that is correct by construction.

Dictionary Search

How to search a word W in a dictionary:

- Open at random page;
- Check the first (F) an last (L) word of page;
- If W comes between F and L, we found it;
- Otherwise:
  - If W comes before F: discard pages on the right;
  - If W comes after L: discard pages on the left;
  - Start again from beginning.

Variables of the algorithm

The dictionary is always divided in three parts:

- The discarded left part;
- The discarded right part;
- The active part: where we are searching.

We also need a variable for the randomly chosen page in the searching part.

- Searched word: w;
- Left limit of active part: l;
- Right limit of active part: r;
- Randomly chosen page of the active part: p.

More precise algorithm

```
input w;
l:= first page;
r:= last page;
p:= random page between l and r;
repeat while w is not on p:
    if w comes before the first word of p, r := page before p;
    if w comes after the last word of p, l := page after p;
    p:= random page between l and r;
```

Specification

What does it mean for this algorithm to be correct?
Assuming that, before the execution,
the following proposition (P) is true:

- The dictionary is in alphabetical order,
- w is in the dictionary;

Then, after the execution,
the following proposition (Q) is true:

- w is on page p,
- the value of w has not changed.

P is called a precondition of the program.
Q is called a postcondition of the program.

Write pre- and post- condition in curly brackets before and after the program, respectively:

- input w;
  {dictionary ordered, w in dictionary, w=W}
- l:= first page;
- r:= last page;
- p:= random page between l and r;
- repeat while w is not on p:
  - if w comes before the first word of p, r := page before p;
  - if w comes after the last word of p, l := page after p;
  - p:= random page between l and r;

  {w on p, w=W}

To prove that the program satisfies this specification:
Insert other propositions (assertions) inside the program.

Annotated Program

```
input w;
{dictionary ordered, w in dictionary, w=W}
l:= first page;
r:= last page;
p:= random page between l and r;
{w on p or w in [l..p) or w in (p..r], w=W}
repeat while w is not on p:
{Invariant: w on p or w in [l..p) or w in (p..r], w=W}
    if w comes before the first word of p, r := page before p;
    if w comes after the last word of p, l := page after p;
    p:= random page between l and r;
{w on p, w=W}
```

Invariant: we mean that this proposition is true at the beginning of the loop and it remains true after every iteration of the loop.