Contract-based Software Development

Rasmus Guldborg Pedersen

January 2015

Overview

Program assertion as program logic

What is a program assertion - and in what sense is it a contract? Outline the basic program logic for assignment and some control structures. Show how this program logic can be used to give a formal proof of a fragment of code.

Assertions

- Assertion is a predicate.

Assertions

- Assertion is a predicate.
- A property we *think* is true at that place during execution.

Assertions

- Assertion is a predicate.
- A property we *think* is true at that place during execution.
- An assertion is valid if it's always true.

Assertions

- Assertion is a predicate.
- A property we *think* is true at that place during execution.
- An assertion is valid if it's always true.
- Abort if invalid.

Assertion as a contract

- The prior statement must guarantee that the assertion is true.

Assertion as a contract

- The prior statement must guarantee that the assertion is true.
- The statement that follow can assume that the assertion is true.

Assignment

A form of substitution.

```
// { x < N }
x = x + 1;
// { x ≤ N }
```

Assignment

A form of substitution.

```
// { x < N }
// { x + 1 < N + 1 }
// { x + 1 <= N }
x = x + 1;
// { x <= N }
```

Assignment continued

Swapping values.

$$// \{ (x = X) \wedge (y = Y) \}$$

```
x = x + y
y = x - y
x = x - y
```

$$// \{ (x = Y) \wedge (y = X) \}$$

Assignment continued

$$// \{\ (x = X) \wedge (y = Y)\ \}$$
$$// \{\ (x + y - y = X) \wedge (y = Y)\ \}$$
```
x = x + y ;
```
$$// \{\ (x - x + y = Y) \wedge (x - y = X)\ \}$$
```
y = x - y ;
```
$$// \{\ (x - y = Y) \wedge (y = X)\ \}$$
```
x = x - y
```
$$// \{\ (x = Y) \wedge (y = X)\ \}$$

Selection

```
// { Q }
if (B) {
    // { Q ∧ B } S1 { R }
}
else {
    // { Q ∧ ≠ B } S2 { R }
}
// { R }
```

Selection

$\{Q \wedge B\} S_1 \rightarrow \{R\}$
$\{Q \wedge \neq B\} S_2 \rightarrow \{R\}$

$\{Q\} \rightarrow \{R\}$

The End

"Testing shows the presence, not the absence of bugs."
— Edsger W. Dijkstra