

## B Exercises 3

### Generalised Substitutions and Proof Obligations

The objective of this set of tutorial exercises is to use the semantics of generalised substitutions to compute proof obligations for operations.

The proof obligation for maintaining the machine invariant for the operation:

$$result \leftarrow Op(args) \hat{=} \mathbf{PRE} P \mathbf{THEN} G \mathbf{END}$$

is  $I \wedge P \Rightarrow [G] I$

where  $I$  is the state invariant.

Remember that substitution distributes through conjunction, that is

$$[G] (R_1 \wedge R_2) = [G] R_1 \wedge [G] R_2$$

This allows the computation of separate proof obligations for each conjunct, rather than one larger proof obligation.

1. The **Simple** machine
  - (a) Calculate the proof obligations for each operation of the simple machine.
  - (b) Remove the precondition of the **Decrement** operation and re-compute the proof obligation.  
Comment on the result.

2. The **Bank** machine. Compute the preconditions of the **Deposit** **Withdraw** and **Balance** operations.

Note very carefully that

$$f(x) := y \hat{=} f := f \triangleleft \{x \mapsto y\}$$

3. Assuming that **varA** and **varB** are both integers show that

$$\mathbf{varA} := \mathbf{varB} - \mathbf{varA} ; \mathbf{varB} := \mathbf{varB} - \mathbf{varA} ; \mathbf{varA} := \mathbf{varA} + \mathbf{varB}$$

is equivalent to

$$\mathbf{varA}, \mathbf{varB} := \mathbf{varB}, \mathbf{varA}$$

4. **Traffic lights** Compute the proof obligations for the operations of the simple **SimpleTwoWay** traffic light machine.
5. **Examples from Wordsworth**
  - (a) Section 2.6: exercises 2.3 and 2.4.
  - (b) Section 3.5: exercises 3.6, 3.7 and 3.8.

```

MACHINE Simple
VARIABLES num
INVARIANT  $num \in \mathbb{N}$ 
INITIALISATION  $num := 0$ 

OPERATIONS
  Set ( val )  $\hat{=}$ 
    PRE  $val \in \mathbb{N}$ 
    THEN  $num := val$ 
    END ;
   $val \leftarrow$  Get  $\hat{=}$ 
    BEGIN  $val := num$  END ;
  Increment  $\hat{=}$ 
    BEGIN  $num := num + 1$  END ;
  Decrement  $\hat{=}$ 
    PRE  $1 \leq num$ 
    THEN  $num := num - 1$ 
    END
END

```

```

MACHINE Bank ( maxaccount )
CONSTRAINTS
    maxaccount ∈ ℕ1
SETS
    ACCOUNT
PROPERTIES
    card ( ACCOUNT ) = maxaccount
VARIABLES
    accounts ,
    balance
INVARIANT
    accounts ⊆ ACCOUNT ∧
    balance ∈ accounts → ℕ
INITIALISATION
    accounts , balance := {} , {}

OPERATIONS
account ← NewAccount ≐
    PRE accounts ≠ ACCOUNT
    THEN
        ANY acc
        WHERE acc ∈ ACCOUNT − accounts
        THEN account := acc ||
            accounts := accounts ∪ { acc } ||
            balance ( acc ) := 0
        END
    END ;
Deposit ( account , amount ) ≐
    PRE account ∈ accounts ∧ amount ∈ ℕ
    THEN balance ( account ) := balance ( account ) + amount
    END ;
Withdraw ( account , amount ) ≐
    PRE account ∈ accounts ∧ amount ∈ ℕ ∧ amount ≤ balance ( account )
    THEN balance ( account ) := balance ( account ) − amount
    END ;
bal ← Balance ( account ) ≐
    PRE account ∈ accounts
    THEN bal := balance ( account )
    END ;
holdings ← Holdings ≐
    BEGIN
        holdings := ∑ account . ( account ∈ accounts | balance ( account ) )
    END
END

```

**MACHINE** *SimpleTwoWay*

**SETS**

$DIRECTION = \{ NorthSouth, EastWest \};$

$LIGHT = \{ Red, Green, Amber \}$

**VARIABLES**

*lights*

**INVARIANT**

$lights \in DIRECTION \rightarrow LIGHT \wedge$

$( lights ( NorthSouth ) \in \{ Green, Amber \} \Rightarrow lights ( EastWest ) = Red ) \wedge$

$( lights ( EastWest ) \in \{ Green, Amber \} \Rightarrow lights ( NorthSouth ) = Red )$

**INITIALISATION**

$lights := \{ NorthSouth \mapsto Red, EastWest \mapsto Red \}$

**OPERATIONS**

**ToRed** ( *dir* )  $\hat{=}$

**PRE**  $dir \in DIRECTION \wedge lights ( dir ) = Amber$

**THEN**  $lights ( dir ) := Red$

**END ;**

**ToGreen** ( *dir* )  $\hat{=}$

**PRE**  $dir \in DIRECTION \wedge lights ( dir ) = Red \wedge$

$( dir = NorthSouth \Rightarrow lights ( EastWest ) = Red ) \wedge$

$( dir = EastWest \Rightarrow lights ( NorthSouth ) = Red )$

**THEN**  $lights ( dir ) := Green$

**END ;**

**ToAmber** ( *dir* )  $\hat{=}$

**PRE**  $dir \in DIRECTION \wedge lights ( dir ) = Green$

**THEN**  $lights ( dir ) := Amber$

**END**

**END**