Hoare Logic for Concurrent Programs

Mads Dam

Parallel While Programs

Extend while language of previous lecture:

c ::= skip | x := e | c ; c | if e then c else c | while e do c | cobegin c || c coend

Shared memory!

Issues of interference, atomicity and nondeterminism must be taken into account, e.g.

```
      y := x ; x := y +1
vs    (x,y) := (x + 1,x)
```


Transition Semantics

$$\frac{(c_1,\sigma) \to \sigma'}{(\text{cobegin } c_1 \parallel c_2 \text{ coend},\sigma) \to (c_2,\sigma')}$$

$$\frac{(c_2,\sigma) \to \sigma'}{(\text{cobegin } c_1 \parallel c_2 \text{ coend},\sigma) \to (c_1,\sigma')}$$

$$\frac{(c_1,\sigma) \to (c_1',\sigma')}{(\text{cobegin } c_1 \parallel c_2 \text{ coend},\sigma) \to (\text{cobegin } c_1' \parallel c_2 \text{ coend}, \sigma')}$$

$$\frac{(c_2,\sigma) \to (c_2',\sigma')}{(\text{cobegin } c_1 \parallel c_2 \text{ coend},\sigma) \to (\text{cobegin } c_1 \parallel c_2' \text{ coend}, \sigma')}$$

Rule for cobegin ... coend

Owicki-Gries proof rule:

$$\frac{\{\phi_1\}\ c_1\ \{\psi_1\} \qquad \{\phi_2\}\ c_2\ \{\psi_2\}}{\{\phi_1 \wedge \phi_2\}\ c_1 \parallel c_2\ \{\psi_1 \wedge \psi_2\}}$$

Side condition:

The proofs of $\{\phi_1\}\ c_1\ \{\psi_1\}$ and $\{\phi_2\}\ c_2\ \{\psi_2\}$ must be *interference-free*

Not compositional!

Interference Freedom

Let a proof outline Δ of $\{\phi\}\ c\ \{\psi\}$ be given.

A *critical formula* of Δ is either ψ or a formula ϕ' appearing immediately before some statement in Δ

Let proof outlines Δ_1 of $\{\phi_1\}\ c_1\ \{\psi_1\}$ and Δ_2 of $\{\phi_2\}\ c_2\ \{\psi_2\}$ be given.

Δ_2 does not *interfere* with Δ_1, if for every critical formula ϕ of Δ_1 and triple $\{\phi_2'\}\ c_2'\ \{\psi_2'\}$ appearing in Δ_2, $\{\phi \wedge \phi_2'\}\ c_2'\ \{\phi\}$.

Need consider only those c_2' that are assignments

Then Δ_1 and Δ_2 are interference free, if Δ_1 and Δ_2 do not interfere with each other

Example

```
P: cobegin
                P1: bal := bal + dep
        ||      P2: if bal > 1000
                then credit := 1
                else credit := 0
   coend
```

Proof goal:

$\{bal = B \wedge dep > 0\}$

P

$\{bal = B + dep \wedge dep > 0 \wedge (credit = 1 \to bal > 1000)\}$

Proof of Example

1. Build proof outline Δ_1 of
   $\{bal = B \wedge dep > 0\}\ P_1\ \{bal = B + dep \wedge dep > 0\}$
2. Build proof outline Δ_2 of
   $\{true\}\ P_2\ \{credit = 1 \rightarrow bal > 1000\}$
3. Prove that Δ_1 and Δ_2 are interference-free
4. Conclude by rule for cobegin ... coend

Proof Outline Δ_1

$\{bal = B \wedge dep > 0\}$
$\{bal + dep = B + dep \wedge dep > 0\}$
bal := bal + dep
$\{bal = B + dep \wedge dep > 0\}$

Critical formulas:

- $\phi_{1,1}$: bal + dep = B + dep $\wedge$ dep > 0
- $\phi_{1,2}$: bal = B + dep $\wedge$ dep > 0

Proof Outline Δ_2

```
{true}
if bal > 1000 then
    {true ∧ bal > 1000}
    {1=1 → bal > 1000}
    credit := 1
    {credit=1 → bal > 1000}
else
    {true ∧ bal <= 1000}
    {0=1 → bal > 1000}
    credit := 0
    {credit = 1 → bal > 1000}
fi ;
{credit=1 → bal > 1000}
```

Critical formulas:

- $\phi_{2,1}$: 1=1 $\rightarrow$ bal > 1000
- $\phi_{2,2}$: 0=1 $\rightarrow$ bal > 1000
- $\phi_{2,3}$: credit = 1 $\rightarrow$ bal > 1000

Proving Interference Freedom

Need to prove, for each $i \in \{1,2\}$ and $j \in \{1,2,3\}$:

1. $\{\phi_{1,i} \wedge \phi_{2,1}\}$ credit := 1 $\{\phi_{1,i}\}$
2. $\{\phi_{1,i} \wedge \phi_{2,2}\}$ credit := 0 $\{\phi_{1,i}\}$
3. $\{\phi_{2,j} \wedge \phi_{1,1}\}$ bal := bal + dep $\{\phi_{2,j}\}$

A total of 7 proof goals

Triples of type 1 and 2 hold trivially since no $\phi_{1,i}$ mentions credit

The type 3 goal $\{\phi_{2,2} \wedge \phi_{1,1}\}$ bal := bal + dep $\{\phi_{2,j}\}$ is trivially valid

Remains to prove:

- $\{(1=1 \rightarrow bal > 1000) \wedge bal + dep = B + dep \wedge dep > 0\}$ bal := bal + dep $\{1=1 \rightarrow bal > 1000\}$
- $\{(credit = 1 \rightarrow bal > 1000) \wedge bal + dep = B + dep \wedge dep > 0\}$ bal := bal + dep $\{credit = 1 \rightarrow bal > 1000\}$

Notes

If P_1 had been withdrawal
bal := bal – wdr
where wdr > 0 last step of proof would not have gone through

A program which never grants credit would satisfy the specification!

Would like postcondition of the form
$(credit=1 \rightarrow bal > 1000) \wedge (credit=0 \rightarrow bal <= 1000)$

But this would lead to violation of interference freedom. Why?

Completeness and Compositionality

For completeness need auxillary variables, explicit new variables which record state and history information

Compositional versions exists using ”assumption-guarantee reasoning”:

$$\Gamma_A, \Gamma_G \vdash \{\phi\}\ P\ \{\psi\}$$

Meaning:

- In an environment which always maintain formulas in Γ_A invariant
- When starting in initial state satisfying ϕ
- P will always maintain formulas in Γ_G invariant
- And if and when P terminates, ψ will hold

More info: De Roever et al: Concurrency Verification: Introduction to Compositional and Noncompositional Methods, CUP 2001

Auxillary Variables

Let c be a program and A a set of variables in c

A is a set of auxillary variables of c if

- Variables in A occurs only in assignments
  So: Not in assignment guards or tests in loops or conditionals
- If $x \in A$ occurs in an assignment
  $$(x_1,\ldots,x_n) := (E_1,\ldots,E_n)$$
  then x occurs in E_i only when $x_i \in A$
  So: Variables in A cannot influence variables outside A
- *erase*(c,A): c with all assignments to auxillary variables in A, and all assignments () := () erased

Auxillary Variable Rule

Proof rule:

$$\frac{\{\phi\}\ c\ \{\psi\}}{\{\phi\}\ c'\ \{\psi\}}$$

Side condition:

- There is a set A of auxillary variables of c such that
  c' = *erase*(c)
- ψ does not mention variables in A

Example

```
P: cobegin
          x := x + 1
     ||   x := x + 1
   coend
```

Proof goal:

{x = 0} P {x = 2}

This proof needs auxillary variables!

Idea: Add auxillary variables $done_1$, $done_2$ to catch when each of the assignments have been executed

Proof of Example

Proof outline Δ_1:

$\{\neg done_1 \wedge (\neg done_2 \rightarrow x = 0) \wedge (done_2 \rightarrow x = 1)\}$

$(x,done_1) := (x+1,true)$

$\{done_1 \wedge (\neg done_2 \rightarrow x = 1) \wedge (done_2 \rightarrow x = 2)\}$

Proof outline Δ_2:

$\{\neg done_2 \wedge (\neg done_1 \rightarrow x = 0) \wedge (done_1 \rightarrow x = 1)\}$

$(x,done_2) := (x+1,true)$

$\{done_2 \wedge (\neg done_1 \rightarrow x = 1) \wedge (done_1 \rightarrow x = 2)\}$

Proof of Example, II

Exercise: Check that Δ_1 and Δ_2 are interference free

By the Owicki-Gries rule + rule of consequence we obtain

$\{x=0 \wedge \neg done_1 \wedge \neg done_2\}$ P' {x = 2}

where P' is P with assignments augmented with auxillary variables as on previous slide

By Hoare logic reasoning:

$\{x = 0\}\ (done_1,done_2) := (false,false)$; P' {x = 2}

By the auxillary variable rule:

{x = 0} P {x = 2}