DD1350 Logik för dataloger

Fö 14 – Hoare-logik, forts

1

Partiell korrekthet

$\models_{\text{par}}$ $(\!|\ \phi\ |\!)\ P\ (\!|\ \psi\ |\!)$ gäller om:

om exekveringen av programmet P börjar i ett tillstånd där förvillkoret ϕ är sant,

och exekveringen terminerar,

så är eftervillkoret sant ψ i sluttillståndet

Exempel

Fakultet-programmet $Fac1$

```
y = 1;
z = 0;
while (z != x) {
    z = z + 1;
    y = y * z;
}
```

kan specifieras med Hoare-tripletten

$$(|\ x \geq 0\ |)\ \ Fac1\ (|\ y = x!\ |)$$

Exempel

Fakultet-programmet $Fac2$

```
y = 1;
while (x != 0) {
    y = y * x;
    x = x - 1;
}
```

kan specifieras med Hoare-tripletten

$$(|\ x \geq 0 \wedge x = x_0\ |)\ \ Fac2\ \ (|\ y = x_0!\ |)$$

Programverifiering

Med regler
- över Hoare-trippletter
- vi fokuserar på partiell korrekthet

Bevis
- som bevisträd

  …eller som s.k. ”tablåer”
- reducerar validiteten av Hoare-tripletter till validiteten av predikatlogiska formler över aritmetiken!

Regler: Tilldelning

$$\frac{-}{(\!|\ \psi[E/x]\ |\!)\ \ x = E\ \ (\!|\ \psi\ |\!)}$$

Regler: ”Implied”

$$\frac{\vdash\ \phi' \rightarrow \phi \qquad (|\ \phi\ |)\ C\ (|\ \psi\ |)}{(|\ \phi'\ |)\ C\ (|\ \psi\ |)}$$

$$\frac{(|\ \phi\ |)\ C\ (|\ \psi\ |) \qquad \vdash\ \psi \rightarrow \psi'}{(|\ \phi\ |)\ C\ (|\ \psi'\ |)}$$

Regler: Sammansättning

$$\frac{(|\ \phi\ |)\ C_1\ (|\ \eta\ |) \qquad (|\ \eta\ |)\ C_2\ (|\ \psi\ |)}{(|\ \phi\ |)\ C_1\ ;\ C_2\ (|\ \psi\ |)}$$

inför ett mellanliggande påstående η i kontrollpunkten före C_2

Exempel

Programmet *Swap*

```
z = x;
x = y;
y = z;
```

kan specificeras med

$$(\!|\ x = x_0 \wedge y = y_0 |\!)\ \ Swap\ \ (\!|\ x = y_0 \wedge y = x_0 |\!)$$

Bevistablå

Korrekthetsbevis kan presenteras som ”annoterade” program, där ”annotationerna” är påståenden associerade med kontrollpunkter.

Mer lättläst än bevisträd.

Operationell tolkning:

Om exekveringen börjar i något tillstånd där förvillkoret (dvs första påståendet) är sant,

så gäller att varenda gång exekveringen når en viss kontrollpunkt, så är alla annotationer sanna i den punkten

Bevis i tablåform

$(\!| \; x = x_0 \wedge y = y_0 \; |\!)$ Precondition
$(\!| \; y = y_0 \wedge x = x_0 \; |\!)$ Implied (✓)
```
z = x;
```
$(\!| \; y = y_0 \wedge z = x_0 \; |\!)$ Assignment
```
x = y;
```
$(\!| \; x = y_0 \wedge z = x_0 \; |\!)$ Assignment
```
y = z;
```
$(\!| \; x = y_0 \wedge y = x_0 \; |\!)$ Assignment

Beviset i tablåform (steg 1)

$(\!| \; x = x_0 \wedge y = y_0 \; |\!)$ Precondition

```
z = x;

x = y;

y = z;
```
$(\!| \; x = y_0 \wedge y = x_0 \; |\!)$ Postcondition

Beviset i tablåform (steg 2)

$(\!| \; x = x_0 \wedge y = y_0 \; |\!)$ Precondition

```
z = x;

x = y;
```

$(\!| \; x = y_0 \wedge z = x_0 \; |\!)$

```
y = z;
```

$(\!| \; x = y_0 \wedge y = x_0 \; |\!)$ Assignment

Beviset i tablåform (steg 3)

$(\!| \; x = x_0 \wedge y = y_0 \; |\!)$ Precondition

```
z = x;
```

$(\!| \; y = y_0 \wedge z = x_0 \; |\!)$

```
x = y;
```

$(\!| \; x = y_0 \wedge z = x_0 \; |\!)$ Assignment

```
y = z;
```

$(\!| \; x = y_0 \wedge y = x_0 \; |\!)$ Assignment

Beviset i tablåform (steg 4)

$(\!| \; x = x_0 \wedge y = y_0 \; |\!)$ Precondition
$(\!| \; y = y_0 \wedge x = x_0 \; |\!)$
```
z = x;
```
$(\!| \; y = y_0 \wedge z = x_0 \; |\!)$ Assignment
```
x = y;
```
$(\!| \; x = y_0 \wedge z = x_0 \; |\!)$ Assignment
```
y = z;
```
$(\!| \; x = y_0 \wedge y = x_0 \; |\!)$ Assignment

Beviset i tablåform (steg 5)

Vi har en kontrollpunkt med två påståenden, som ger ett bevisförpliktelse:

$$\vdash \; x = x_0 \wedge y = y_0 \rightarrow \; y = y_0 \wedge x = x_0$$

som uppenbart är ett sant aritmetiskt påstående som enkelt bevisas (t.ex. i naturlig deduktion)

Regler: If

$$\frac{(\!|\ \phi \wedge B\ |\!)\ C_1\ (\!|\ \psi\ |\!)\quad (\!|\ \phi \wedge \neg B\ |\!)\ C_2\ (\!|\ \psi\ |\!)}{(\!|\ \phi\ |\!)\ \mathtt{\textbf{if}}\ B\ \{C_1\}\ \mathtt{\textbf{else}}\ \{C_2\}\ (\!|\ \psi\ |\!)}$$

Exempel

Programmet *Abs*

```
if (x > 0) {
     y = x;
} else {
     y = -x;
}
```

kan specificeras med

$$(\!|\ x = x_0\ |\!)\ Abs\ (\!|\ y = |\ x_0\ |\ |\!)$$

Bevistablå (steg 1)

$(\!| \ x = x_0 \ |\!)$ Precondition

```
if (x > 0) {

    y = x;

} else {

    y = -x;

}
```

$(\!| \ y = |x_0| \ |\!)$ Postcondition

Bevistablå (steg 2)

$(\!| \ x = x_0 \ |\!)$ Precondition

```
if (x > 0) {
```

$(\!| \ x = x_0 \wedge x > 0 \ |\!)$ If

```
    y = x;
```

$(\!| \ y = |x_0| \ |\!)$

```
} else {
```

$(\!| \ x = x_0 \wedge \neg(x > 0) \ |\!)$ If

```
    y = -x;
```

$(\!| \ y = |x_0| \ |\!)$

```
}
```

$(\!| \ y = |x_0| \ |\!)$ Postcondition

Bevistablå (steg 3)

$(\| x = x_0 \|)$	Precondition
`if (x > 0) {`	
$(\| x = x_0 \wedge x > 0 \|)$	If
$(\| x = \| x_0 \| \|)$	
`y = x;`	
$(\| y = \| x_0 \| \|)$	Assignment
`} else {`	
$(\| x = x_0 \wedge \neg(x > 0) \|)$	If
$(\| -x = \| x_0 \| \|)$	
`y = -x;`	
$(\| y = \| x_0 \| \|)$	Assignment
`}`	
$(\| y = \| x_0 \| \|)$	Postcondition

Bevistablå (steg 4)

$(\| x = x_0 \|)$	Precondition
`if (x > 0) {`	
$(\| x = x_0 \wedge x > 0 \|)$	If
$(\| x = \| x_0 \| \|)$	Implied (✓)
`y = x;`	
$(\| y = \| x_0 \| \|)$	Assignment
`} else {`	
$(\| x = x_0 \wedge \neg(x > 0) \|)$	If
$(\| -x = \| x_0 \| \|)$	Implied (✓)
`y = -x;`	
$(\| y = \| x_0 \| \|)$	Assignment
`}`	
$(\| y = \| x_0 \| \|)$	Postcondition

Regler: Partial-while

$$\frac{(|\ \eta \wedge B\ |)\ C\ (|\ \eta\ |)}{(|\ \eta\ |)\ \mathtt{while}\ B\ \{C\}\ (|\ \eta \wedge \neg B\ |)}$$

Slinginvariant (eng: loop invariant)

Slinginvarianter

- En *slinginvariant* till slingan

  $$\mathtt{while}\ B\ \{C\}$$

  är ett påstående η för vilket

  $$\models_{\mathrm{par}} (|\ \eta \wedge \mathrm{B}\ |)\ C\ (|\ \eta\ |)$$

  är sant.
- Till Hoare-trippeln

  $$(|\ \phi\ |)\ \mathtt{while}\ B\ \{C\}\ (|\ \psi\ |)$$

  behöver vi en slinginvariant η för vilken:

  $\vdash\ \phi \rightarrow \eta$ och

  $\vdash\ \eta \wedge \neg B \rightarrow \psi$ (ekvivalent $\vdash \eta \rightarrow \psi \vee B$)

Slinginvarianter

Det finns ett helt ”påståendeintervall” att välja från:

$$\vdash \ \phi \rightarrow \eta \rightarrow \psi \vee B$$

Två omedelbara kandidater att testa:

– ϕ

– $\psi \vee B$

...ifall de är slinginvarianter.

Exempel

- Verifiera fakultet-programmet $Fac1$

```
y = 1;
z = 0;
while (z != x) {
      z = z + 1;
      y = y * z;
}
```

specifierat med Hoare-trippeln

$$(\!|\ x \geq 0 \wedge x = x_0\ |\!)\ Fac1\ (\!|\ y = x_0!\ |\!)$$

Bevistablå (steg 1)

$(\!| \; x \geq 0 \wedge x = x_0 \; |\!)$ — Precondition

```
y = 1;
z = 0;
while (z != x) {

   z = z + 1;
   y = y * z;
}
```

Loop invariant?

$(\!| \; y = x_0! \; |\!)$ — Postcondition

Bevistablå (steg 2)

$(\!| \; x \geq 0 \wedge x = x_0 \; |\!)$ — Precondition

```
y = 1;
z = 0;
```

$(\!| \; \boldsymbol{y = z! \wedge x = x_0 \wedge z \geq 0} \; |\!)$ — Loop invariant

```
while (z != x) {

   z = z + 1;
   y = y * z;
}
```

$(\!| \; y = x_0! \; |\!)$ — Postcondition

Bevistablå (steg 3)

$(\| x \geq 0 \wedge x = x_0 \|)$	Precondition
`y = 1;`	
`z = 0;`	
$(\| y = z! \wedge x = x_0 \wedge z \geq 0 \|)$	Loop invariant
`while (z != x) {`	
$\quad(\| y = z! \wedge x = x_0 \wedge z \geq 0 \wedge z \neq x \|)$	Partial-while
`    z = z + 1;`	
`    y = y * z;`	
$\quad(\| y = z! \wedge x = x_0 \wedge z \geq 0 \|)$	Partial-while
`}`	
$(\| y = z! \wedge x = x_0 \wedge z \geq 0 \wedge \neg(z \neq x) \|)$	Partial-while
$(\| y = x_0! \|)$	Postcondition

Bevistablå (steg 4)

$(\| x \geq 0 \wedge x = x_0 \|)$	Precondition
$(\| 1 = 0! \wedge x = x_0 \wedge 0 \geq 0 \|)$	
`y = 1;`	
$(\| y = 0! \wedge x = x_0 \wedge 0 \geq 0 \|)$	Assignment
`z = 0;`	
$(\| y = z! \wedge x = x_0 \wedge z \geq 0 \|)$	Assignment
`while (z != x) {`	
$\quad(\| y = z! \wedge x = x_0 \wedge z \geq 0 \wedge z \neq x \|)$	Partial-while
$\quad(\| y \cdot (z+1) = (z+1)! \wedge x = x_0 \wedge z+1 \geq 0 \|)$	
`    z = z + 1;`	
$\quad(\| y \cdot z = z! \wedge x = x_0 \wedge z \geq 0 \|)$	Assignment
`    y = y * z;`	
$\quad(\| y = z! \wedge x = x_0 \wedge z \geq 0 \|)$	Assignment
`}`	
$(\| y = z! \wedge x = x_0 \wedge z \geq 0 \wedge \neg(z \neq x) \|)$	Partial-while
$(\| y = x_0! \|)$	Postcondition

Bevistablå (slut)

$(\!\vert\ x \geq 0 \wedge x = x_0\ \vert\!)$	Precondition
$(\!\vert\ 1 = 0! \wedge x = x_0 \wedge 0 \geq 0\ \vert\!)$	Implied (✓)
`y = 1;`	
$(\!\vert\ y = 0! \wedge x = x_0 \wedge 0 \geq 0\ \vert\!)$	Assignment
`z = 0;`	
$(\!\vert\ \mathbf{y = z! \wedge x = x_0 \wedge z \geq 0}\ \vert\!)$	Assignment
`while (z != x) {`	
$\quad(\!\vert\ \mathbf{y = z! \wedge x = x_0 \wedge z \geq 0} \wedge z \neq x\ \vert\!)$	Partial-while
$\quad(\!\vert\ y \cdot (z+1) = (z+1)! \wedge x = x_0 \wedge z+1 \geq 0\ \vert\!)$	Implied (✓)
`    z = z + 1;`	
$\quad(\!\vert\ y \cdot z = z! \wedge x = x_0 \wedge z \geq 0\ \vert\!)$	Assignment
`    y = y * z;`	
$\quad(\!\vert\ \mathbf{y = z! \wedge x = x_0 \wedge z \geq 0}\ \vert\!)$	Assignment
`}`	
$(\!\vert\ \mathbf{y = z! \wedge x = x_0 \wedge z \geq 0} \wedge \neg(z \neq x)\ \vert\!)$	Partial-while
$(\!\vert\ y = x_0!\ \vert\!)$	Implied (✓)

Hoare-logik: Egenskaper

- Hoare-logik är **sund.**
- Hoare-logik är **fullständig...**
  - men det krävs ett ”orakel” (en procedur som avgör) påståenden som används av ”Implied”-regeln
- Hoare-logik är **oavgörbar**
  - pga ovanstående kommentar + att det inte finns någon automatisk procedur för att finna slinginvarianter.