

26

Programbeviser I.

Bevis kontra 'check af assertions' i Eiffel.

Betingelser og bevisregler.

Hvad er programverifikation?

Bevisregel for 'tom kommando'.

Bevisregel for assignment.

Bevisregler for selektive kommandoer.

Bevisregler for iterative kommandoer.

Eksempler.

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Formålet med programbeviser.

```
op1(...) is
  require pre-betingelse
  do
    ...
    En sekvens af kommandoer
    ...
  ensure post-betingelse
end; --op1
```

Løbende **check** af pre-betingelse, post-betingelse og andre assertions.



Bevis af, at hvis pre-betingelsen er opfyldt, så er postbetingelsen også opfyldt.

Vi må have *bevisregler* for forskellige kommandoer i sproget.

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Betingelser, Hoare sætninger og bevisregler.

Hoare sætning:

$$\{P\} \ C \ \{Q\}$$

Hvis P er opfyldt *før* udførelsen af kommandoen C, og hvis C afsluttes normalt, så vil Q være opfyldt *efter* udførelsen af C.

Partiel korrekthed.

Bevisregel:

$$\frac{\vdash H_1, \vdash H_2, \dots, \vdash H_n}{\vdash H}$$

Præmisser

Konklusion

Hvis vi kan vise $H_1, H_2, \dots, H_n$ så kan vi konkludere H

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Modsætningen til **partiel korrekthed** er **total korrekthed**.

Årsagen til, at et programfragment ikke afsluttes normalt kan være, at det kommer ind i en uendelig løkke, eller at programmet aborterer.

Med ovenstående notation kan vi udtrykke rollerne af de (effektive) pre- og postbetingelser i operationer:

{pre-betingelse} operations-krop {post-betingelse}.

Vi er som regel interesserede i de stærkest mulige Hoare sætninger. Man styrker en Hoaresætning ved enten at styrke postbetingelsen Q, eller ved at svække prebetingelsen P.

For at få givne Hoaresætninger til at "fitte" med givne omgivelser, kan man også være interesseret i at svække Hoaresætninger.

Husk, at hvis $P \Rightarrow Q$ siger vi, at P er stærkere en Q, og at Q er svagere en P.

Hvis der indgår variable i et udsagn, er der relativt færre værdier af variabelen, der opfylder et stærkt udsagn end et svagt udsagn.

Program verifikation.

Verifikation af et program P, som består af en kommandosekvens CS:

```
{ pre }  
CS  
{ post }
```

- **Forlæns verifikation** (højrekonstruktion):

Vi antager at *pre* gælder før udførelsen af CS.
Vi skal vise at *post* gælder efter CS.

- **Baglæns verifikation** (venstrekonstruktion):

Vi skal vise at
såfrem *post* skal gælde efter udførelse af CS
er det tilstrækkeligt at forudsætte, at *pre* er opfyldt inden udførelsen af CS.

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Svækkelse af Hoare sætninger.

$$\frac{\vdash P \Rightarrow P_1, \vdash \{P_1\} S \{Q\}}{\vdash \{P\} S \{Q\}}$$

$\{P_1\} S \{Q\}$ svækkes ved en stærkere prebetingelse.

$$\frac{\vdash \{P\} S \{Q_1\}, \vdash Q_1 \Rightarrow Q}{\vdash \{P\} S \{Q\}}$$

$\{P\} S \{Q_1\}$ svækkes ved en svagere postbetingelse.

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Bevisregel for den tomme kommando.

$$\{P\} \quad \{P\}$$

Hvis P er opfyldt før en tom kommando,
er P også opfyldt efter denne.

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Den tomme kommando udtrykkes som bekendt syntaktisk som den tomme streng.

Bevisreglen for den tomme kommando siger, at en betingelse ikke ændres af den tomme kommando.

Bevisregel for assignment.

$$\boxed{\{P_e^x\} \quad x := e \quad \{P\}}$$

$\{P_e^x\}$: Betingelsen P med alle forekomster af x substitueret med e .

Baglæns verifikation (Tilbageføring):

Hvis betingelsen P gælder efter assignmentet, skal P , med alle forekomster af x substitueret med e , gælde før udførelsen af assignmentet.

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Man er umiddelbart fristet til at lave en bevisregel $\{P\} x := e \{Q\}$, hvor Q bl.a. udtrykker at $x = e$. Denne regel vil vi gerne benytte i fremadrettet verifikation.

Som diskuteret i Jan Stage's rapport, giver dette ikke mening hvis x optræder i udtrykket e .

Vi må derfor ty til baglæns verifikation: Givet P ovenfor, hvad er den svageste prebetingelse Q af assignmentet så

$$\{Q\} x := e \{P\}$$

holder. Det er udsagnet P med alle forekomster af x substitueret med e .

Bevisregler for kommando-sekvenser.

$$\frac{\vdash \{P_0\} C_1 \{P_1\}, \vdash \{P_1\} C_2 \{P_2\}}{\vdash \{P_0\} C_1 ; C_2 \{P_2\}}$$

Hvis vi kan vise at

1. Hvis P_0 er opfyldt før C_1 da er P_1 opfyldt efter C_1 , og
2. Hvis P_1 er opfyldt før C_2 da er P_2 opfyldt efter C_2 .

så kan vi konkludere at

- Hvis P_0 er opfyldt før $C_1 ; C_2$ da er P_2 opfyldt efter sekvensen.

$$\frac{\vdash \{P_0\} C_1 \{P_1\}, \vdash \{P_2\} C_2 \{P_3\}, \vdash P_1 \Rightarrow P_2}{\vdash \{P_0\} C_1 ; C_2 \{P_3\}}$$

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Den nederste bevisregel er en variation af den øverste, hvor postbetingelsen af C_1 ikke er identisk med prebetingelsen af C_2 . Istedet for gælder at C_1 's postbetingelse medfører C_2 's prebetingelse.

Eksempel 1.

Ombytning af værdierne af to variable x og y:

$$\{ x = a, y = b \}$$
$$\begin{array}{l} t := x; \\ x := y; \\ y := t \end{array}$$
$$\{ x = b, y = a \}$$

Bevisreglerne for hhv. assignment og kommando-sekvens benyttes til at vise at

- hvis $\{ x = b, y = a \}$ skal gælde efter de tre assignments, da skal $\{ x = a, y = b \}$ nødvendigvis gælde før udførelsen af disse.

(Baglæns verifikation).

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Ved forelæsningen vil vi gennemføre beviset af dette.

Bevisregler for selektive kommandoer (1).

if-then-else:

$$\frac{\vdash \{P \text{ and } B\} \text{ CS1 } \{Q\}, \vdash \{P \text{ and } \neg B\} \text{ CS2 } \{Q\}}{\vdash \{P\} \text{ if } B \text{ then CS1 else CS2 end } \{Q\}}$$

if-then:

$$\frac{\vdash \{P \text{ and } B\} \text{ CS } \{Q\}, \vdash (P \text{ and } \neg B \Rightarrow Q)}{\vdash \{P\} \text{ if } B \text{ then CS end } \{Q\}}$$

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Bevisreglen for if-then formen kan let udledes fra bevisreglen for if-then-else kommandoen. Antag at else-delen er den tomme kommando-sekvens. Benyt den trivielle bevisregel for den tomme kommando, samt reglen for svækkelse af Hoare-sætninger.

Bevisregler for selektive kommandoer (2).

if-then-else ala Eiffel:

$$\frac{\vdash \{P \text{ and } B_0\} \text{ CS } \{Q\}, \vdash X \quad ?}{\vdash \{P\} \text{ if } B_0 \text{ then CS } \text{ elsif } B_1 \text{ then CS}_1 \dots \text{ elsif } B_n \text{ then CS}_n \text{ else CS}_{n+1} \text{ end } \{Q\}}$$

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Den selektive kontrolstruktur, for hvilken der skitseres en bevisregel på denne slide, svarer til if-then-else i Eiffel.

Opgave: Find X så ovenstående bliver en bevisregel for Eiffel's if-then-else kommando.

Eksempel 2.

```
Max(a,b: integer): integer is
  require true
  do
    {P}
    if a >= b
    then {P1} result := a {Q}
    else {P2} result := b {Q}
    end
    {Q}
  ensure result = max(a,b)
end --Max
```

Fremgangsmåde i bevis:

1. Q sættes til postbetingelsen af Max.
2. Find P_1 ved at tilbageføre Q over 'result := a'
3. Find P_2 ved at tilbageføre Q over 'result := b'
4. Gæt P.
5. Vise: $P \text{ and } B \Rightarrow P_1$.
6. Vise: $P \text{ and } \neg B \Rightarrow P_2$.
7. Benyt bevisreglen for if-then-else til at slutte, at Q gælder hvis P er opfyldt.

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Bevisregler for iterative kommandoer.

Pascal:

$$\frac{\vdash \{P \text{ and } B\} \text{ CS } \{P\}}{\vdash \{P\} \text{ while } B \text{ do } \text{CS} \text{ end } \{P \text{ and } \neg B\}}$$

$$\frac{\vdash \{P\} \text{ CS } \{Q\}, \vdash (Q \text{ and } \neg B) \Rightarrow P}{\vdash \{P\} \text{ repeat } \text{CS} \text{ until } B \{Q \text{ and } B\}}$$

Eiffel:

$$\frac{\vdash \{P \text{ and } \neg B\} \text{ CS } \{P\}}{\vdash \{P\} \text{ from until } B \text{ loop } \text{CS} \text{ end } \{P \text{ and } B\}}$$

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Øverst er vist en bevisregel for den traditionelle while-sætning ala Pascal, hvor der itereres over en kommandosekvens.

Dernæst en tilsvarende regel for repeat.

Nederst er vist en regel for Eiffel's loop konstruktion. Initialiseringskommandoerne efter 'from' er udeladt, idet de semantisk set lige så godt kunne være placeret før loop kommandoen (og i sekvens med denne). Invariant og variant-delene er ikke medtaget, idet de ikke bidrager til semantikken af loop'en. Bemærk at P svarer til Eiffel's invariant.

Eksempel 3.

```
myfib(n: integer): INTEGER is
require  $n \geq 0$ 
local a: INTEGER ;
      b: INTEGER ;
      count: INTEGER ;
      br: INTEGER

do
  {P}
from a := 1; b := 0; count := n;
invariant P2
until count = 0
loop
  {P1}
  br := b; b := a;
  a := a + br; count := count - 1
  {P2}
end;
  {P3}
  result := b
  {Q}
ensure result = fib(n)
end; -- myfib
```

Fremgangsmåde i bevis:

1. Q sættes til postbetingelsen af myfib.
2. Gæt på invarianten P₂.
3. Find P₁ ved tilbageføring.
4. Vise at P₂ and $\neg (\text{count} = 0) \Rightarrow P_1$.
5. Benyt bevisreglen for loop til at slutte, at P₃ er P₂ and (count = 0).
5. Tilbagefør P₂ over de tre initiale assignments, og få dermed P₂'.
6. Vise af $P \Rightarrow P_2$ '.
7. Tilbagefør postbetingelsen Q over 'result := b' og få Q'.
8. Vise at P₃ $\Rightarrow$ Q'.

PS1 -- Programbeviser I

© Kurt Nørmark, Aalborg Universitet, 1994.

Noter

Opgave: Gennemfør verifikation af, at ovenstående beregner Fibonacci tal. Vær speciel omhyggelig med at finde P₁ ved tilbageføring af P₂ over de fire assignments.

Bemærk at vi tidligere i noterne har diskuteret netop dette eksempel (se slide "Eksempel på specifikation, implementation og krav til korrekthedsbevis for en funktion", i kapitlet om specifikation med logiske udtryk). Ved denne lejlighed fandt vi invarianten, som betegnes med P₂ ovenfor. Invarianten er givet på sidste side i ovennævnte kapitel.