DD1350 Logik för dataloger

OMTENTAMEN
30 maj 2013, 09.00 - 11.00

Dilian Gurov
KTH CSC
08–790 81 98

Skriv svaren direkt på blanketten. Två formelblad är bifogade. Inga andra hjälpmedel är tillåtna.
Kravet för godkänt på omtentan är att vara godkänd på båda E delar.

Del 1E

Kravet för godkänt på denna del är 7 poäng av 10. Om du är godkänd på kontrollskrivningen, är du automatiskt godkänd på *första uppgiften* (du får alltså 5 poäng och bör inte lösa den uppgiften).

1. Betrakta följande resonemang: 5p

   Om sekventen är bevisbar i bevissystemet och bevissystemet är sund, så gäller sekventen. Men sekventen gäller inte. Därför är sekventen obevisbar i bevissystemet.

   Föreslå en formalisering av resonemanget i form av en satslogisk *sekvent*, och visa att resonemanget är felaktigt genom att hitta en *motvaluering*.

   – Atomer och deras tolkning:

   b : sekventen är bevisbar i bevissystemet
   s : bevissystemet är sunt g : sekventen gäller

   – Sekvent:

   $b \wedge s \rightarrow g, \neg g \vdash \neg b$

   – Motvaluering:

   $\{b : T, s : F, g : F\}$

   – Finns det fler motvalueringar? Om ja, vilka är de?

   Nej

   – Visa en *sanningsvärdestabell* för att motivera dina svar:

	b	s	g	$b \wedge s$	$b \wedge s \rightarrow g$	$\neg g$	$\neg b$
	T	T	T	T	T	F	F
	T	T	F	T	F	T	F
	T	F	T	F	T	F	F
→	T	F	F	F	T	T	F
	F	T	T	F	T	F	T
	F	T	F	F	T	T	T
	F	F	T	F	T	F	T
	F	F	F	F	T	T	T

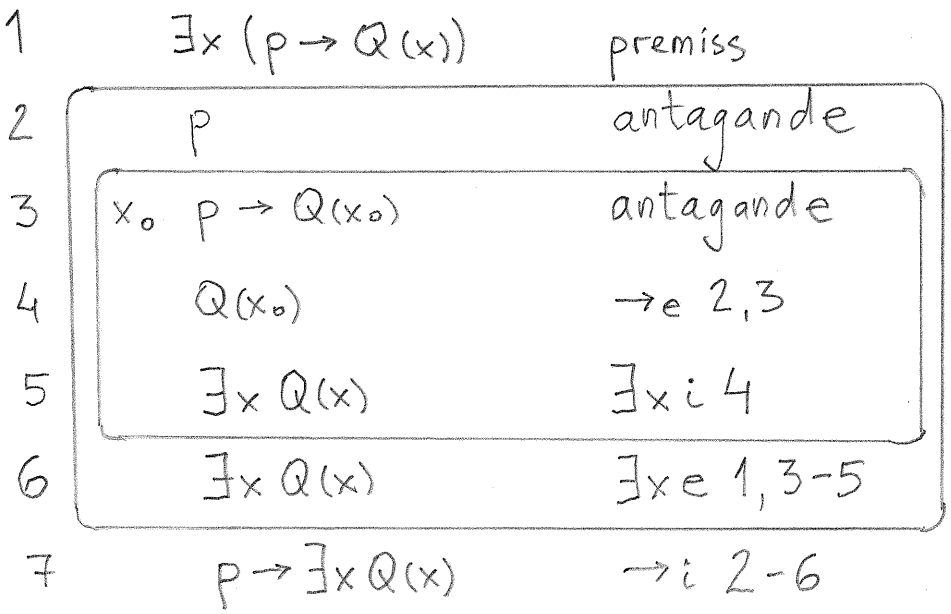
Om du är godkänd på *kontrollskrivningen*, kryssa här: ☐

2. Presentera ett *bevis* i naturlig deduktion till följande sekvent: 5p

$$\exists x\ (p \to Q(x)) \vdash p \to \exists x\ Q(x)$$

Rita tydligt alla boxar för att visa räckvidden för alla antaganden och nya variabler i beviset.

– Bevis:



Del 2E

Kravet för godkänt på denna del är 10 poäng av 15.

1. Binära träd över symboler kan definieras som en termmängd med BNF så här: 5p

$$BSTree \ ::= \ \mathsf{leaf}\,(Letter) \ | \ \mathsf{bstree}\,(BSTree, BSTree)$$

Definiera *induktivt* funktionen **treverse**(t) som vänder om på trädet till dess vänster–höger spegelbild. Notera att du kommer behöva din definition i uppgift 2C.1.

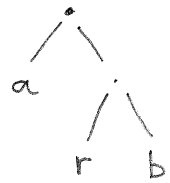
– Induktiv definition:

treverse (leaf(a)) $\stackrel{def}{=}$ leaf(a) 1p

treverse (bstree(t_1, t_2)) $\stackrel{def}{=}$ bstree (treverse(t_2), treverse(t_1)) 2p

Använd din definition för att *stegvist* beräkna:

treverse (bstree (leaf(a), bstree (leaf(r), leaf(b))))



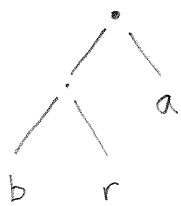
– Stegvis beräkning (OBS: slutresultatet måste vara ett korrekt träd!):

treverse (bstree (leaf(a), bstree (leaf(r), leaf(b))))

= bstree (treverse (bstree (leaf(r), leaf(b))), treverse (leaf(a)))

= bstree (bstree (treverse (leaf(b)), treverse (leaf(r))), leaf(a)) 2p

= bstree (bstree (leaf(b), leaf(r)), leaf(a))



2. Betrakta följande (oönskad) beteendeegenskap för hissar: 5p

Hissen kommer så småningom att stanna och dörrarna att öppnas och förbli öppna.

Föreslå en formalisering av egenskapen i form av en CTL–formel, där du använder atomen move som är sant när hissen är i rörelse och atomen open som är sant när hissens dörr är öppen.

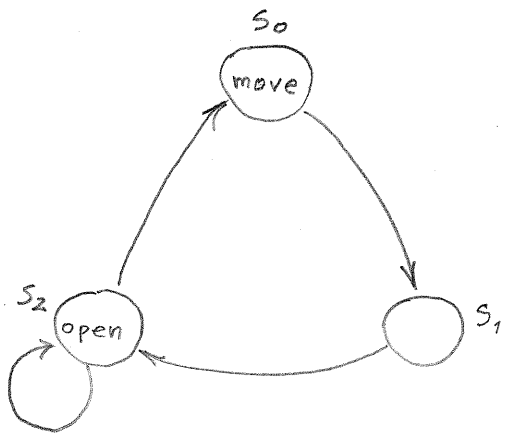
– CTL formel:

$$AF\,(\neg move \wedge AG\,open) \qquad 2p$$

Låt $Atoms \stackrel{\text{def}}{=} \{\mathsf{move}, \mathsf{open}\}$, och låt $\mathcal{M}$ vara en hiss–modell definierad som:

$$\begin{array}{rcl} S & \stackrel{\text{def}}{=} & \{s_0, s_1, s_2\} \\ \rightarrow & \stackrel{\text{def}}{=} & \{(s_0, s_1), (s_1, s_2), (s_2, s_2), (s_2, s_0)\} \\ L & : & s_0 \mapsto \{\mathsf{move}\} \\ & & s_1 \mapsto \{\} \\ & & s_2 \mapsto \{\mathsf{open}\} \end{array}$$

– Presentera modellen grafiskt. Gäller beteendeegenskapen du formaliserade i tillståndet s_0? Motivera ditt svar genom att hänvisa till CTLs semantik (se bifogat formelblad):



Egenskapen gäller inte i s_0 därför att det finns en stig från s_0, nämligen: 1p

$\overline{s_0 s_1 s_2} = s_0 s_1 s_2 s_0 s_1 s_2 \ldots$ 1p

där $\neg move \wedge AG\,open$ inte gäller någonstans i stigen: för varje tillstånd i stigen där $\neg move$ gäller finns det ett senare tillstånd där open inte gäller. 1p

3. Kom ihåg programmeringsspråket som betraktats i kursen i samband med programspecifikation och verifikation. Skriv en *specifikation* i form av en Hoare–trippel till ett program Div som beräknar den vanliga divisionsoperationen över positiva heltal. Specifikationen skall inte använda sig av nya funktionssymboler förutom de som redan finns i språket (tex ingen explicit divisionsoperator). Det skall vara entydigt från specifikationen hur programmet ska användas för att beräkna divisionen m/n av två positiva heltal m och n. 5p

– Specifikation med Hoare–trippel:

$$\{x = x_0 \wedge y = y_0 \wedge x > 0 \wedge y > 0\}\ \text{Div}\ \{z * y_0 \leq x_0 \wedge (z+1) * y_0 > x_0\}$$ 2p

– Förklara intuitivt din specifikation:

Slutvärdet på z är det största talet som multiplicerat med startvärdet på y är mindre eller lika med startvärdet på x, dvs resultatet av heltalsdivisionen av startvärden på x och y. 1p

Implementera Div, dvs skriv kod till Div i programmeringsspråket, så att den uppfyller din specifikation.

– Program:

```
z = 0;
while x≥y {
    x = x-y;
    z = z+1;
}
```
2p

Del 2C

För betyg D måste du ha klarat båda E–delar och fått 6 poäng utav 15 på den här delen, medan 11 poäng krävs för betyg C.

1. Den induktiva BNF-definitionen av symbollistor som termmängder är: 8p

$$\mathit{List} ::= \mathsf{empty} \mid \mathsf{cons}\,(\mathit{Letter}, \mathit{List})$$

Konkatenering $\mathbf{conc}\,(u, v)$ definierade vi induktivt så här:

$$\begin{aligned} \mathbf{conc}\,(\mathsf{empty}, v) &\stackrel{\text{def}}{=} v \\ \mathbf{conc}\,(\mathsf{cons}\,(a, u'), v) &\stackrel{\text{def}}{=} \mathsf{cons}\,(a, \mathbf{conc}\,(u', v)) \end{aligned}$$

medan omvändning $\mathbf{reverse}\,(u)$ så här:

$$\begin{aligned} \mathbf{reverse}\,(\mathsf{empty}) &\stackrel{\text{def}}{=} \mathsf{empty} \\ \mathbf{reverse}\,(\mathsf{cons}\,(a, u')) &\stackrel{\text{def}}{=} \mathbf{conc}\,(\mathbf{reverse}\,(u'), \mathsf{cons}\,(a, \mathsf{empty})) \end{aligned}$$

Dessutom bevisade vi i klassrummet att (lemma A):

$$\forall u\, \forall v\, \mathbf{reverse}\,(\mathbf{conc}\,(u, v)) = \mathbf{conc}\,(\mathbf{reverse}\,(v), \mathbf{reverse}\,(u))$$

Funktionen $\mathbf{leaves}\,(t)$, som samlar alla symboler från löven i binära symbolträdet t i en lista (se uppgift 2E.1), kan definieras induktivt så här:

$$\begin{aligned} \mathbf{leaves}\,(\mathsf{leaf}(a)) &\stackrel{\text{def}}{=} \mathsf{cons}\,(a, \mathsf{empty}) \\ \mathbf{leaves}\,(\mathsf{bstree}\,(t_1, t_2)) &\stackrel{\text{def}}{=} \mathbf{conc}\,(\mathbf{leaves}\,(t_1), \mathbf{leaves}\,(t_2)) \end{aligned}$$

Använd din definition på $\mathbf{treverse}\,(t)$ från uppgift 2E.1 och bevisa med *strukturell induktion* att:

$$\forall t\, \mathbf{leaves}\,(\mathbf{treverse}\,(t)) = \mathbf{reverse}\,(\mathbf{leaves}\,(t))$$

– Bevis med strukturell induktion (OBS: *inte* med fullständig induktion!):

• Basfall t = leaf(a) för någon bokstav a.

Leaves (treverse (leaf(a)))

= Leaves (leaf(a)) {Def treverse}

= cons(a, empty) {Def Leaves}

= conc(empty, cons(a, empty)) {Def conc}

= conc (reverse(empty), cons(a, empty)) {Def reverse}

= reverse (cons(a, empty)) {Def reverse}

= reverse (leaves (leaf(a)) {Def Leaves}

• Fall $t = bstree(t_1, t_2)$ för några delträd t_1 och t_2.

Anta $leaves(treverse(t_1)) = reverse(leaves(t_1))$
och $leaves(treverse(t_2)) = reverse(leaves(t_2))$ } Ind.hyp.

Vi har:

$leaves(treverse(bstree(t_1, t_2)))$

$= leaves(bstree(treverse(t_2), treverse(t_1)))$ {Def treverse}

$= conc(leaves(treverse(t_2)), leaves(treverse(t_1)))$ {Def leaves}

$= conc(reverse(leaves(t_2)), reverse(leaves(t_1)))$ {Ind. hyp.}

$= reverse(conc(leaves(t_1), leaves(t_2)))$ {Lemma A}

$= reverse(leaves(bstree(t_1, t_2)))$ {Def leaves}

• rätta fall : 1p • rätt anv. av ind.hyp : 1p • rätta kommentarer : 1p
• rätt ind.hyp : 1p • rätt anv. av Lemma A : 1p • resten : 1-3p

2. Betrakta följande program Max som beräknar maximala startvärdet på variablerna x, y och z: 7p

```
if (x > y) {
   if (x > z) {
      m = x;
   } else {
      m = z;
   }
} else {
   if (y > z) {
      m = y;
   } else {
      m = z;
   }
}
```

Programmet är specificerat med en Hoare–trippel enligt partiell korrekthet:

$$\{\!|\, x = x_0 \wedge y = y_0 \wedge z = z_0 \,|\!\}\ \mathsf{Max}\ \{\!|\, m = max\,(x_0, y_0, z_0) \,|\!\}$$

Notera att $m = max(x_0, y_0, z_0)$ kan uttryckas som

$$(m = x_0 \vee m = y_0 \vee m = z_0) \wedge (m \geq x_0 \wedge m \geq y_0 \wedge m \geq z_0)$$

Verifiera programmet med Hoare-logik (se formelblad). Presentera beviset som bevistablå.

– Bevistablå:

$\{\!| x=x_0 \wedge y=y_0 \wedge z=z_0 |\!\}$
if $(x>y)$ {
  $\{\!| x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge x>y |\!\}$
  if $(x>z)$ {
    $\{\!| x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge x>y \wedge x>z |\!\}$
    $\{\!| x=\max(x_0,y_0,z_0) |\!\}$ ← ①
    $m=x;$
    $\{\!| m=\max(x_0,y_0,z_0) |\!\}$
  } else {
    $\{\!| x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge x>y \wedge \neg(x>z) |\!\}$
    $\{\!| z=\max(x_0,y_0,z_0) |\!\}$ ← ②
    $m=z;$
    $\{\!| m=\max(x_0,y_0,z_0) |\!\}$
  }
  $\{\!| m=\max(x_0,y_0,z_0) |\!\}$
} else {
  ⋮

} else {
  $\{\!| x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge \neg(x>y) |\!\}$
  if $(y>z)$ {
    $\{\!| x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge \neg(x>y) \wedge y>z |\!\}$
    $\{\!| y=\max(x_0,y_0,z_0) |\!\}$ ← ③
    $m=y;$
    $\{\!| m=\max(x_0,y_0,z_0) |\!\}$
  } else {
    $\{\!| x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge \neg(x>y) \wedge \neg(y>z) |\!\}$
    $\{\!| z=\max(x_0,y_0,z_0) |\!\}$ ← ④
    $m=z;$
    $\{\!| m=\max(x_0,y_0,z_0) |\!\}$
  }
  $\{\!| m=\max(x_0,y_0,z_0) |\!\}$
}
$\{\!| m=\max(x_0,y_0,z_0) |\!\}$

Identifiera alla *bevisförpliktelser* (resulterande från regeln Implied) och motivera varför de gäller.

– Bevisförpliktelser:

Vi har fyra bevisförpliktelser ① – ④ ovan:

$\vdash x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge x>y \wedge x>z \rightarrow x=\max(x_0,y_0,z_0)$

$\vdash x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge x>y \wedge \neg(x>z) \rightarrow z=\max(x_0,y_0,z_0)$

$\vdash x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge \neg(x>y) \wedge y>z \rightarrow y=\max(x_0,y_0,z_0)$

$\vdash x=x_0 \wedge y=y_0 \wedge z=z_0 \wedge \neg(x>y) \wedge \neg(y>z) \rightarrow z=\max(x_0,y_0,z_0)$

som gäller uppenbarligen, enligt meningen av max.

Lycka till!