DD1350 Logik för dataloger

RÄTTNINGSMALL TILL TENTAMEN
20 december 2011, 11.30 - 13.00

Dilian Gurov
KTH CSC

Kravet för godkänt på tentan är att vara godkänd på båda E–delar **och** att ha deltagit i kamraträttningen.

Del 1E

1. Betrakta följande resonemang: 4p

   Om listan är sorterad eller slumpmässigt genererad, så är första elementet i listan större än det sista elementet. Om listan är sorterad, så är första elementet i listan inte större än det sista elementet. Därför är listan osorterad och slumpmässigt genererad.

   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:

   p : listan är sorterad
   q : listan är slumpmässigt genererad
   r : första elementet i listan är större än det sista elementet

   } 1p

   – Sekvent:

   $$p \vee q \rightarrow r,\ p \rightarrow \neg r \vdash \neg p \wedge q$$ 1p

   – Motvaluering:

   $$\{p : F,\ q : F,\ r : T\}$$

   – Finns det fler motvalueringar? Ja, en:

   $$\{p : F,\ q : F,\ r : F\}$$

   } 1p

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

– Sekvent:

$$p \vee q \rightarrow r,\ p \rightarrow \neg r \vdash \neg p \wedge q$$

– Motvaluering:

$$\{p : F,\ q : F,\ r : T\}$$

– Finns det fler motvalueringar? Ja, en:

$$\{p : F,\ q : F,\ r : F\}$$

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

	p	q	r	$p \vee q$	$p \vee q \rightarrow r$	$\neg r$	$p \rightarrow \neg r$	$\neg p$	$\neg p \wedge q$
	T	T	T	T	T	F	F	F	F
	T	T	F	T	F	T	T	F	F
	T	F	T	T	T	F	F	F	F
	T	F	F	T	F	T	T	F	F
	F	T	T	T	T	F	T	T	T
	F	T	F	T	F	T	T	T	T
→	F	F	T	F	T	F	T	T	F
→	F	F	F	F	T	T	T	T	F

1p

2. Presentera ett *bevis* i naturlig deduktion till följande sekvent: [6p]

$$\forall x\ (P(x) \to Q(x)),\ \exists x\ (Q(x) \to \neg P(x)) \vdash \exists x\ \neg P(x)$$

Rita tydligt alla boxar för att visa räckvidden för alla antaganden och nya variabler i beviset. Ett formelblad med alla regler är bifogat.

Bevis:

1	$\forall x\ (P(x) \to Q(x))$	premiss
2	$\exists x\ (Q(x) \to \neg P(x))$	premiss
3	$x_0\ Q(x_0) \to \neg P(x_0)$	antagande
4	$P(x_0) \to Q(x_0)$	$\forall x$ e 1 (x_0)
5	$P(x_0)$	antagande
6	$Q(x_0)$	$\to$e 5,4
7	$\neg P(x_0)$	$\to$e 6,3
8	$\bot$	$\neg$e 5,7
9	$\neg P(x_0)$	$\neg$i 5–8
10	$\exists x\ \neg P(x)$	$\exists x$ i 9 (x_0)
11	$\exists x\ \neg P(x)$	$\exists x$ e 2,3–10

1p var

-1p om fel kommentar

-1p om boxar saknas

Del 2E

1. Betrakta fragmentet av den temporala logiken CTL som vi använde i andra labbuppgiften: 5p

$$\begin{aligned}\phi \;::=\; & p \mid \neg p \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \\ & \mid \mathsf{AX}\,\phi' \mid \mathsf{AG}\,\phi' \mid \mathsf{AF}\,\phi' \\ & \mid \mathsf{EX}\,\phi' \mid \mathsf{EG}\,\phi' \mid \mathsf{EF}\,\phi'\end{aligned}$$

Notera att negation bara förekommer över atomer. Definiera *induktivt* negeringen $\mathsf{neg}(\phi)$ på formler i syntaxen från ovan så att $\mathsf{neg}(\phi)$ ger en formel (i samma syntax!) som gäller i ett tillstånd om ϕ inte gäller i tillståndet, och tvärtom. Så ska tex $\mathsf{neg}(\mathsf{EG}\ \neg\mathsf{p})$ ge $\mathsf{AF}\ p$. (Tips: tänk på de olika De Morgans lagar.)

– Induktiv definition:

$$\begin{aligned}
\mathsf{neg(p)} &\stackrel{\text{def}}{=} \neg p \\
\mathsf{neg}(\neg\mathsf{p}) &\stackrel{\text{def}}{=} p \\
\mathsf{neg}(\phi_1 \wedge \phi_2) &\stackrel{\text{def}}{=} \mathsf{neg}(\phi_1) \vee \mathsf{neg}(\phi_2) \\
\mathsf{neg}(\phi_1 \vee \phi_2) &\stackrel{\text{def}}{=} \mathsf{neg}(\phi_1) \wedge \mathsf{neg}(\phi_2) \\
\mathsf{neg}(\mathsf{AX}\ \phi') &\stackrel{\text{def}}{=} \mathsf{EX}\ \mathsf{neg}(\phi') \\
\mathsf{neg}(\mathsf{AG}\ \phi') &\stackrel{\text{def}}{=} \mathsf{EF}\ \mathsf{neg}(\phi') \\
\mathsf{neg}(\mathsf{AF}\ \phi') &\stackrel{\text{def}}{=} \mathsf{EG}\ \mathsf{neg}(\phi') \\
\mathsf{neg}(\mathsf{EX}\ \phi') &\stackrel{\text{def}}{=} \mathsf{AX}\ \mathsf{neg}(\phi') \\
\mathsf{neg}(\mathsf{EG}\ \phi') &\stackrel{\text{def}}{=} \mathsf{AF}\ \mathsf{neg}(\phi') \\
\mathsf{neg}(\mathsf{EF}\ \phi') &\stackrel{\text{def}}{=} \mathsf{AG}\ \mathsf{neg}(\phi')
\end{aligned}$$

1p, 1p, 1p

3p

-1p om något fall saknas

Använd din definition för att *stegvist* beräkna:

$$\mathsf{neg}(\mathsf{AG}\ (\neg\mathsf{p} \vee \mathsf{AF}\ \mathsf{q}))$$

– Stegvis beräkning (slutresultatet måste vara en formel i syntaxen från ovan):

$$\begin{aligned}
& \mathsf{neg}(\mathsf{AG}\ (\neg\mathsf{p} \vee \mathsf{AF}\ \mathsf{q})) \\
=\ & \mathsf{EF}\ \mathsf{neg}(\neg\mathsf{p} \vee \mathsf{AF}\ \mathsf{q}) \\
=\ & \mathsf{EF}\ (\mathsf{neg}(\neg\mathsf{p}) \wedge \mathsf{neg}(\mathsf{AF}\ \mathsf{q})) \\
=\ & \mathsf{EF}\ (p \wedge \mathsf{EG}\ \mathsf{neg}(\mathsf{q})) \\
=\ & \mathsf{EF}\ (p \wedge \mathsf{EG}\ \neg q)
\end{aligned}$$

1p för 4 steg

1p för korrekt slutresultat

2p

2. Beteendet av en läskförsäljningsautomat kan beskrivas som en modell över atomerna [5p]

$$Atoms \stackrel{\text{def}}{=} \{\mathsf{pengar}, \mathsf{knapp}, \mathsf{lask}, \mathsf{retur}\}$$

med tolkningen:

pengar	:	pengar är instoppade
knapp	:	knappen är intryckt
lask	:	läsk är utlämnad
retur	:	pengarna är returnerade

Betrakta följande beteendeegenskap för sådana automater:

> *Alltid när pengar är instoppade och knappen är intryckt kommer så småningom antingen läsk bli utlämnad eller pengarna bli returnerade.*

Föreslå en formalisering av egenskapen i form av en CTL–formel.

– CTL formel:

$$\mathsf{AG}\ ((\mathsf{pengar} \wedge \mathsf{knapp}) \rightarrow \mathsf{AF}\ (\mathsf{lask} \vee \mathsf{retur}))$$

2p

Låt $\mathcal{M}$ vara modellen definierad som:

$$\begin{array}{rcl} S & \stackrel{\text{def}}{=} & \{s_0, s_1, s_2, s_3, s_4\} \\ \rightarrow & \stackrel{\text{def}}{=} & \{(s_0, s_1), (s_1, s_2), (s_2, s_3), (s_2, s_4), (s_3, s_0), (s_4, s_0)\} \\ L & : & s_0 \mapsto \{\} \\ & & s_1 \mapsto \{\mathsf{pengar}\} \\ & & s_2 \mapsto \{\mathsf{pengar}, \mathsf{knapp}\} \\ & & s_3 \mapsto \{\mathsf{lask}\} \\ & & s_4 \mapsto \{\mathsf{retur}\} \end{array}$$

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

1p → Ja, egenskapen gäller, därför att $(\mathsf{pengar} \wedge \mathsf{knapp}) \rightarrow \mathsf{AF}\ (\mathsf{lask} \vee \mathsf{retur})$ gäller i alla tillstånd som kan nås från s_0:

1p → om $\mathsf{pengar} \wedge \mathsf{knapp}$ inte gäller i något tillstånd, så gäller implikationen;

1p → $\mathsf{pengar} \wedge \mathsf{knapp}$ gäller bara i s_2, och där gäller $\mathsf{AF}\ (\mathsf{lask} \vee \mathsf{retur})$, därför att alla stigar som börjar i s_2 passerar s_3 eller s_4 där $\mathsf{lask} \vee \mathsf{retur}$ gäller.

3p

3. Betrakta följande program Uttag som kan användas för att ta ut 1000 kronor från ett konto om saldot är större än så, där variabeln `flagga` används för att meddela om transaktionen lyckades eller inte: [5p]

```
belopp = 1000;
if (belopp <= saldo) {
   saldo = saldo - belopp;
   flagga = 1;
} else {
   flagga = 0;
}
```

Specificera programmet med en Hoare–trippel $(\!| \phi |\!)$ Uttag $(\!| \psi |\!)$ enligt partiell korrekthet. Det skall vara entydigt från specifikationen hur programmet Uttag kan användas *utan* att känna till själva koden.

Specifikation som Hoare–trippel: 3p

$$(\!| \underbrace{saldo = saldo_0}_{1p} |\!) \text{ Uttag } (\!| \underbrace{\underbrace{(flagga = 0 \wedge saldo = saldo_0 \wedge saldo < 1000)}_{1p} \vee \underbrace{(flagga = 1 \wedge saldo = saldo_0 - 1000 \wedge saldo \geq 0)}_{1p}}_{1p} |\!)$$

Förklara och motivera din specifikation:

1p → Förvillkoret behövs för att kunna relatera slutvärdet på *saldo* till dess startvärde. 2p

1p → Eftervillkoret specifierar att i slutet *flagga* antingen är noll eller ett. I första fallet är saldot mindre än 1000 och (därför) oförändrat. I andra fallet har saldot minskat med 1000 och är icke negativt.

Del 2C

1. Betrakta igen CTL–fragmentet från uppgift 2E.1 och din induktiva definition av negering $\mathsf{neg}(\phi)$. Använd *strukturell induktion* för att bevisa för alla formler ϕ i fragmentet och alla tillstånd s i alla modeller $\mathcal{M}$ att: [9p]

$$\mathcal{M}, s \models \mathsf{neg}(\phi) \leftrightarrow \mathit{inte}\, \mathcal{M}, \mathsf{s} \models \phi$$

Du behöver bara genomföra beviset för följande tre fall: p, $\phi_1 \wedge \phi_2$ och $\mathsf{AG}\ \phi'$. Ett formelblad med CTLs formella semantik är bifogat.

– Bevis:

- Fall $\phi = p$
  Vi har:

$$\begin{aligned} & \mathcal{M}, s \models \mathsf{neg(p)} \\ \leftrightarrow\ & \mathcal{M}, s \models \neg p && \{\text{Def. neg}\} \\ \leftrightarrow\ & \mathit{inte}\, \mathcal{M}, s \models p && \{\text{Def. } \models\} \end{aligned}$$

- Fall $\phi = \phi_1 \wedge \phi_2$
  Antag $\mathcal{M}, s \models \mathsf{neg}(\phi_1) \leftrightarrow \mathit{inte}\, \mathcal{M}, \mathsf{s} \models \phi_1$ och $\mathcal{M}, s \models \mathsf{neg}(\phi_2) \leftrightarrow \mathit{inte}\, \mathcal{M}, \mathsf{s} \models \phi_2$ för alla tillstånd s i alla modeller $\mathcal{M}$ (Induktionshypotes).
  Vi har:

$$\begin{aligned} & \mathcal{M}, s \models \mathsf{neg}(\phi_1 \wedge \phi_2) \\ \leftrightarrow\ & \mathcal{M}, s \models \mathsf{neg}(\phi_1) \vee \mathsf{neg}(\phi_2) && \{\text{Def. neg}\} \\ \leftrightarrow\ & \mathcal{M}, s \models \mathsf{neg}(\phi_1)\ \mathit{eller}\ \mathcal{M}, \mathsf{s} \models \mathsf{neg}(\phi_2) && \{\text{Def. } \models\} \\ \leftrightarrow\ & (\mathit{inte}\, \mathcal{M}, s \models \phi_1)\ \mathit{eller}\ (\mathit{inte}\, \mathcal{M}, s \models \phi_2) && \{\text{Ind.hyp.}\} \\ \leftrightarrow\ & \mathit{inte}\, (\mathcal{M}, s \models \phi_1\ \mathit{och}\ \mathcal{M}, s \models \phi_2) && \{\text{Satslogik}\} \\ \leftrightarrow\ & \mathit{inte}\, \mathcal{M}, s \models \phi_1 \wedge \phi_2 && \{\text{Def. } \models\} \end{aligned}$$

- Fall $\phi = \mathsf{AG}\ \phi'$
  Antag $\mathcal{M}, s \models \mathsf{neg}(\phi') \leftrightarrow \mathit{inte}\, \mathcal{M}, \mathsf{s} \models \phi'$ för alla tillstånd s i alla modeller $\mathcal{M}$ (Induktionshypotes).
  Vi har:

$$\begin{aligned} & \mathcal{M}, s \models \mathsf{neg}(\mathsf{AG}\ \phi') \\ \leftrightarrow\ & \mathcal{M}, s \models \mathsf{EF}\ \mathsf{neg}(\phi') && \{\text{Def. neg}\} \\ \leftrightarrow\ & \mathit{finns\ stig}\ s = s1 \rightarrow s_2 \rightarrow \ldots\ \mathit{där}\ \mathcal{M}, s_i \models \mathsf{neg}(\phi')\ \mathit{för\ något\ } i && \{\text{Def. } \models\} \\ \leftrightarrow\ & \mathit{finns\ stig}\ s = s1 \rightarrow s_2 \rightarrow \ldots\ \mathit{där\ inte}\, \mathcal{M}, s_i \models \phi'\ \mathit{för\ något\ } i && \{\text{Ind.hyp.}\} \\ \leftrightarrow\ & \mathit{inte\ i\ alla\ stigar}\ s = s1 \rightarrow s_2 \rightarrow \ldots\ \mathcal{M}, s_i \models \phi'\ \mathit{för\ alla\ } i && \{\text{Satslogik}\} \\ \leftrightarrow\ & \mathit{inte}\, \mathcal{M}, s_i \models \mathsf{AG}\ \phi' && \{\text{Def. } \models\} \end{aligned}$$

1p för fina kommentarer

1p för läsbar formattering

2. Verifiera programmet Uttag från uppgift 2E.3 relativt din specifikation där. Presentera beviset som en bevistablå. Ett formelblad med alla regler är bifogat. [6p]

$\{\!|\ saldo = saldo_0\ |\!\}$ — Förvillkor

$\{\!|\ saldo = saldo_0 \wedge 1000 = 1000\ |\!\}$ — Implied

```
belopp = 1000;
```

$\{\!|\ saldo = saldo_0 \wedge belopp = 1000\ |\!\}$ — Assignment

```
if (belopp <= saldo) {
```

$\{\!|\ saldo = saldo_0 \wedge belopp = 1000 \wedge belopp \leq saldo\ |\!\}$ — If

$\{\!|\ (1 = 0 \wedge saldo - belopp = saldo_0 \wedge saldo - belopp < 1000) \vee (1 = 1 \wedge saldo - belopp = saldo_0 - 1000 \wedge saldo - belopp \geq 0)\ |\!\}$ — Implied

```
    saldo = saldo - belopp;
```

$\{\!|\ (1 = 0 \wedge saldo = saldo_0 \wedge saldo < 1000) \vee (1 = 1 \wedge saldo = saldo_0 - 1000 \wedge saldo \geq 0)\ |\!\}$ — Assignment

```
    flagga = 1;
```

$\{\!|\ (flagga = 0 \wedge saldo = saldo_0 \wedge saldo < 1000) \vee (flagga = 1 \wedge saldo = saldo_0 - 1000 \wedge saldo \geq 0)\ |\!\}$ — Assignment

```
} else {
```

$\{\!|\ saldo = saldo_0 \wedge belopp = 1000 \wedge \neg\,(belopp \leq saldo)\ |\!\}$ — If

$\{\!|\ (0 = 0 \wedge saldo = saldo_0 \wedge saldo < 1000) \vee (0 = 1 \wedge saldo = saldo_0 - 1000 \wedge saldo \geq 0)\ |\!\}$ — Implied

```
    flagga = 0;
```

$\{\!|\ (flagga = 0 \wedge saldo = saldo_0 \wedge saldo < 1000) \vee (flagga = 1 \wedge saldo = saldo_0 - 1000 \wedge saldo \geq 0)\ |\!\}$ — Assignment

```
}
```

$\{\!|\ (flagga = 0 \wedge saldo = saldo_0 \wedge saldo < 1000) \vee (flagga = 1 \wedge saldo = saldo_0 - 1000 \wedge saldo \geq 0)\ |\!\}$ — Eftervillkor

-1p för felaktig påstående / regelapplicering

-1p om inga kommentarer alls

4p

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

Vi har tre bevisförpliktelser:

a) $\vdash \mathit{saldo} = \mathit{saldo}_0 \rightarrow \mathit{saldo} = \mathit{saldo}_0 \wedge 1000 = 1000$

vilken gäller därför att:

- $\mathit{saldo} = \mathit{saldo}_0$ implicerar $\mathit{saldo} = \mathit{saldo}_0$
- $1000 = 1000$ gäller trivialt

b) $\vdash \mathit{saldo} = \mathit{saldo}_0 \wedge \mathit{belopp} = 1000 \wedge \mathit{belopp} \leq \mathit{saldo} \rightarrow$
$(1 = 0 \wedge \mathit{saldo} - \mathit{belopp} = \mathit{saldo}_0 \wedge \mathit{saldo} - \mathit{belopp} < 1000) \vee$
$(1 = 1 \wedge \mathit{saldo} - \mathit{belopp} = \mathit{saldo}_0 - 1000 \wedge \mathit{saldo} - \mathit{belopp} \geq 0)$

vilken gäller därför att:

- första disjunkten är falskt och kan därför ignoreras
- $1 = 1$ gäller trivialt
- $\mathit{saldo} = \mathit{saldo}_0 \wedge \mathit{belopp} = 1000$ implicerar $\mathit{saldo} - \mathit{belopp} = \mathit{saldo}_0 - 1000$
- $\mathit{belopp} \leq \mathit{saldo}$ implicerar $\mathit{saldo} - \mathit{belopp} \geq 0$

c) $\vdash \mathit{saldo} = \mathit{saldo}_0 \wedge \mathit{belopp} = 1000 \wedge \neg\, (\mathit{belopp} \leq \mathit{saldo}) \rightarrow$
$(0 = 0 \wedge \mathit{saldo} = \mathit{saldo}_0 \wedge \mathit{saldo} < 1000) \vee$
$(0 = 1 \wedge \mathit{saldo} = \mathit{saldo}_0 - 1000 \wedge \mathit{saldo} \geq 0)$

vilken gäller därför att:

- andra disjunkten är falskt och kan därför ignoreras
- $0 = 0$ gäller trivialt
- $\mathit{saldo} = \mathit{saldo}_0$ implicerar $\mathit{saldo} = \mathit{saldo}_0$
- $\mathit{belopp} = 1000 \wedge \neg\, (\mathit{belopp} \leq \mathit{saldo})$ implicerar $\mathit{saldo} < 1000$

-1p för felaktig bevisförpliktelse

-1p om ingen motivation alls

2p