Inleiding logica

Inleveropgave 3

Lientje Maas

30 september 2013

Ik (Rijk) heb verbeteringen in rood vermeld. Deze verbeteringen meegenomen zijn dit correcte uitwerkingen van de derde inleveropgaven.

1 Practicum opdrachten

Bij de practicum opdrachten maken we gebruik van het de vervulbaarheidschecker yices (1).

Opgave 6.6.9 (iii)

In deze opgave gaan we onderzoeken of de verzameling $\{\neg(p \vee \neg\neg q), r \rightarrow q, \neg p \rightarrow r\}$ vervulbaar of strijdig is. Hiervoor gebruiken we een yices bestand waarmee we yices vragen een oplossing te vinden voor de formule die bestaat uit de conjunctie van de drie elementen van onze verzameling. Dit yices bestand ziet er als volgt uit:

```
; test for checking satisfiability of
; (~(p | ~~ q )) & (r -> q) & (~ p -> r)
(benchmark whodunnit.smt
:extrapreds ((p) (q) (r))
:formula (and
(not (or p (not(not q))))
(implies r q)
(implies (not p) r)
))
```

Dat wat achter een ; staat is slechts commentaar en wordt niet door yices meegenomen. Dit is puur ter verduidelijking voor onszelf. Als we nu het commando `yices -e -smt bestandsnaam.txt` invoeren, krijgen we de volgende uitvoer: `unsat.`
Dit betekent dat yices geen oplossing heeft kunnen vinden voor de formule bestaande uit de conjunctie van de elementen van onze verzameling. Dit betekent dat de verzameling niet vervulbaar is, ofwel dat deze strijdig is.

Opgave 6.6.14 (iv)

In deze opgave gaan we bewijzen dat de als volgt opgebouwde formule een tautologie is:
$(A \rightarrow (B \rightarrow C)) \leftrightarrow ((A \wedge B) \rightarrow C)$. Als we dit zouden invoeren in yices en we vinden een

oplossing, betekent dit natuurlijk niet direct dat het een tautologie is. Yices geeft namelijk een *mogelijke* oplossing. Als we echter een negatie voor de gehele formule plaatsen en yices dan *geen* oplossing vindt, weten we dat we te maken hebben met een tautologie. Als yices 'unsat' geeft betekent dit dat er geen enkele valuatie te vinden is waarvoor de negatie van onze formule waar is. Hieruit volgt dat voor alle valuaties onze originele formule (zonder negatie) waar is. Dit is de definitie van een tautologie. Ons yices bestand ziet er als volgt uit:

```
; test for checking satisfiability of
; ~((A -> (B -> C)) <-> ((A & B) -> C))
(benchmark whodunnit.smt
:extrapreds ((A) (B) (C))
:formula (not
(and
  (implies (implies A (implies B C)) (implies (and A B) C))
  (implies (implies (and A B) C) (implies A (implies B C))) )
))
```

We zien dat de bi-implicatie is geconstrueerd door een conjunctie van twee implicaties. Als we nu het commando `yices -e -smt bestandsnaam.txt` invoeren, krijgen we de volgende uitvoer: `unsat`.
We hebben al gezien dat we dan kunnen concluderen dat $(A \to (B \to C)) \leftrightarrow ((A \wedge B) \to C)$ een tautologie is.

Opgave 6.6.8 (ii)

In deze opgave laten we zien dat $\neg(p \vee q) \models p \leftrightarrow q$ geldt. Dit betekent dat voor alle valuaties $V \models \neg(p \vee q)$ geldt dat $V \models p \leftrightarrow q$. Als $\neg(p \vee q) \models p \leftrightarrow q$ *niet* geldt, moet er dus een valuatie zijn zodat $V \models \neg(p \vee q)$, maar $V \not\models p \leftrightarrow q$. Als hier geen oplossing voor is, hebben we bewezen dat $\neg(p \vee q) \models p \leftrightarrow q$ geldt. We laten yices naar een oplossing zoeken. Ons bestand ziet er nu als volgt uit:

```
; test for checking satisfiability of
; (~(p | q) & ~(p <-> q))
(benchmark whodunnit.smt
:extrapreds ((p) (q))
:formula (and
  (not (or p q))
  (not (and (implies p q) (implies q p)))
))
```

Als we nu het commando `yices -e -smt bestandsnaam.txt` invoeren, krijgen we de volgende uitvoer: `unsat`.
Zoals eerder beschreven betekent dit dat $\neg(p \vee q) \models p \leftrightarrow q$ geldt.

Formule construeren

In deze opgave gaan we op zoek naar een formule in p_1, p_2, q_1, q_2, r_1, r_2 en r_3 die uitdrukt dat $r_1 r_2 r_3$ (als binair getal) de som van (de binaire getallen) $p_1 p_2$ en $q_1 q_2$ is. Deze formule zullen we gebruiken om waarden van p_1, p_2, q_1 en q_2 te laten vinden door yices, zodanig dat hun som 3 is. Dat wil zeggen: r_1 is onwaar en r_2 en r_3 zijn waar. We zullen nu eerst een waarheidstafel maken voor p_1, p_2, q_1 en q_2. In deze tafel zullen we vervolgens de verkregen binaire getallen p_1p_2 en q_1q_2 bij elkaar optellen en deze binair weergeven in de tafel met behulp van $r_1r_2r_3$.

p_1	p_2	q_1	q_2	r_1	r_2	r_3
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Uit deze tabel kunnen we de volgende formules halen, die we direct een naam geven:

- Formule A:
  $r_1 \leftrightarrow ((\neg p_1 \wedge p_2 \wedge q_1 \wedge q_2) \vee (p_1 \wedge \neg p_2 \wedge q_1 \wedge \neg q_2) \vee (p_1 \wedge \neg p_2 \wedge q_1 \wedge q_2) \vee (p_1 \wedge p_2 \wedge \neg q_1 \wedge q_2) \vee (p_1 \wedge p_2 \wedge q_1 \wedge \neg q_2) \vee (p_1 \wedge p_2 \wedge q_1 \wedge q_2))$
- Formule B:
  $r_2 \leftrightarrow ((\neg p_1 \wedge \neg p_2 \wedge q_1 \wedge \neg q_2) \vee (\neg p_1 \wedge \neg p_2 \wedge q_1 \wedge q_2) \vee (\neg p_1 \wedge p_2 \wedge \neg q_1 \wedge q_2) \vee (\neg p_1 \wedge p_2 \wedge q_1 \wedge \neg q_2) \vee (p_1 \wedge \neg p_2 \wedge \neg q_1 \wedge \neg q_2) \vee (p_1 \wedge \neg p_2 \wedge \neg q_1 \wedge q_2) \vee (p_1 \wedge p_2 \wedge \neg q_1 \wedge \neg q_2) \vee (p_1 \wedge p_2 \wedge q_1 \wedge q_2))$
- Formule C:
  $r_3 \leftrightarrow ((\neg p_1 \wedge \neg p_2 \wedge \neg q_1 \wedge q_2) \vee (\neg p_1 \wedge \neg p_2 \wedge q_1 \wedge q_2) \vee (\neg p_1 \wedge p_2 \wedge \neg q_1 \wedge \neg q_2) \vee (\neg p_1 \wedge p_2 \wedge q_1 \wedge \neg q_2) \vee (p_1 \wedge \neg p_2 \wedge \neg q_1 \wedge q_2) \vee (p_1 \wedge \neg p_2 \wedge q_1 \wedge q_2) \vee (p_1 \wedge p_2 \wedge \neg q_1 \wedge \neg q_2) \vee (p_1 \wedge p_2 \wedge q_1 \wedge \neg q_2))$

Ook weten we al dat r_1 onwaar is (dus $\neg r_1$ is waar) en dat r_2 en r_3 waar zijn. Als we een conjunctie maken van deze zes formules en deze in yices voegen, zal yices ons geschikte waarden voor p_1, p_2, q_1 en q_2 geven. Deze conjunctie wordt dus $(A \wedge B \wedge C \wedge \neg r_1 \wedge r_2 \wedge r_3)$. Ons yices bestand ziet er nu als volgt uit:

```
(benchmark whodunnit.smt
:extrapreds ((p1) (p2) (q1) (q2) (r1) (r2) (r3))
:formula (and

(and
(implies (or (and (not p1) p2 q1 q2) (and p1 (not p2) q1 (not q2)) (and p1 (not
p2) q1 q2) (and p1 p2 (not q1) q2) (and p1 p2 q1 (not q2)) (and p1 p2 q1 q2)) r1)
```

```
(implies r1 (or (and (not p1) p2 q1 q2) (and p1 (not p2) q1 (not q2)) (and p1 (not
p2) q1 q2) (and p1 p2 (not q1) q2) (and p1 p2 q1 (not q2)) (and p1 p2 q1 q2))))

(and
(implies (or (and (not p1) (not p2) q1 (not q2)) (and (not p1) (not p2) q1 q2) (and
(not p1) p2 (not q1) q2) (and (not p1) p2 q1 (not q2)) (and p1 (not p2) (not q1)
(not q2)) (and p1 (not p2) (not q1) q2) (and p1 p2 (not q1) (not q2)) (and p1 p2
q1 q2)) r2)

(implies r2 (or (and (not p1) (not p2) q1 (not q2)) (and (not p1) (not p2) q1 q2)
(and (not p1) p2 (not q1) q2) (and (not p1) p2 q1 (not q2)) (and p1 (not p2) (not
q1) (not q2)) (and p1 (not p2) (not q1) q2) (and p1 p2 (not q1) (not q2)) (and p1
p2 q1 q2))))

(and

(implies (or (and (not p1) (not p2) (not q1) q2) (and (not p1) (not p2) q1 q2) (and
(not p1) p2 (not q1) (not q2)) (and (not p1) p2 q1 (not q2)) (and p1 (not p2) (not
q1) q2) (and p1 (not p2) q1 q2) (and p1 p2 (not q1) (not q2)) (and p1 p2 q1 (not
q2))) r3)

(implies r3 (or (and (not p1) (not p2) (not q1) q2) (and (not p1) (not p2) q1 q2)
(and (not p1) p2 (not q1) (not q2)) (and (not p1) p2 q1 (not q2)) (and p1 (not p2)
(not q1) q2) (and p1 (not p2) q1 q2) (and p1 p2 (not q1) (not q2)) (and p1 p2 q1
(not q2)))))

(not r1)

r2

r3

))
```

Als we nu het commando `yices -e -smt bestandsnaam.txt` invoeren, krijgen we de volgende uitvoer:

```
sat
<= r1 false>
<= r2 true>
<= r3 true >
<= p1 false>
<= p2 false>
<= q1 true>
<= q2 true>
```

We hebben nu dus geschikte waarden voor p_1, p_2, q_1 en q_2 gevonden, namelijk $p_1 = 0$, $p_2 = 0$,

$q_1 = 1$ en $q_2 = 1$.

Bonusopgave bij Formule construeren

De oplossing die we zojuist hebben gevonden, is slechts één van de mogelijke oplossingen. Als we nu onze conjunctie van de vorige opgave uitbreiden tot $(A \wedge B \wedge C \wedge \neg r_1 \wedge r_2 \wedge r_3) \wedge \neg(\neg p_1 \wedge \neg p_2 \wedge q_1 \wedge q_2)$, zal yices op zoek gaan naar een oplossing ongelijk aan de zojuist gevonden oplossing. Omdat de inhoud van het yices bestand erg veel is om weer te geven, zullen we dit niet nog een keer in zijn geheel doen. We hebben enkel het volgende element toegevoegd aan de conjunctie: `(not (and (not p1) (not p2) q1 q2))`.

Als we nu het commando `yices -e -smt bestandsnaam.txt` invoeren, krijgen we de volgende uitvoer:

```
sat
<= r1 false>
<= r2 true>
<= r3 true >
<= p1 false>
<= p2 true>
<= q1 true>
<= q2 false>
```

Een andere oplossing voor ons probleem is dus $p_1 = 0$, $p_2 = 1$, $q_1 = 1$ en $q_2 = 0$.

Op dezelfde manier kunnen we nu op zoek naar nog een andere oplossing. We voegen nu $\neg(\neg p_1 \wedge p_2 \wedge q_1 \wedge \neg q_2)$, ofwel `(not (and (not p1) p2 q1 (not q2)))`, toe aan onze conjunctie. Yices vindt nu een derde oplossing, namelijk $p_1 = 1$, $p_2 = 0$, $q_1 = 0$ en $q_2 = 1$.

Zo kunnen we weer verder. Nu moeten we zorgen dat yices de zojuist gevonden oplossing óók weglaat. Daarvoor voegen we $\neg(p_1 \wedge \neg p_2 \wedge \neg q_1 \wedge q_2)$, ofwel `(not (and p1 (not p2) (not q1) q2))`, toe aan de conjunctie. Yices geeft ons nu de oplossing $p_1 = 1$, $p_2 = 1$, $q_1 = 0$ en $q_2 = 0$.

De volgende stap is natuurlijk het weglaten van nóg een oplossing: de oplossing die we net gevonden hebben. We voegen $\neg(p_1 \wedge p_2 \wedge \neg q_1 \wedge \neg q_2)$, ofwel `(not (and p1 p2 (not q1) (not q2)))`, toe aan de conjunctie. We krijgen de volgende uitvoer: `unsat.`Dit betekent dat yices geen andere oplossing heeft gevonden, dus dat de vier gevonden oplossingen alle oplossingen zijn.
De oplossingen zijn dus:

- $p_1 = 0$, $p_2 = 0$, $q_1 = 1$ en $q_2 = 1$;
- $p_1 = 0$, $p_2 = 1$, $q_1 = 1$ en $q_2 = 0$;
- $p_1 = 1$, $p_2 = 0$, $q_1 = 0$ en $q_2 = 1$;
- $p_1 = 1$, $p_2 = 1$, $q_1 = 0$ en $q_2 = 0$.

2 Werkgroep opdrachten

Opgave 6.6.1 (ix)

In deze opgave berekenen we $V((r \leftrightarrow \neg p) \leftrightarrow (p \leftrightarrow s))$ gegeven de valuatie V met $V(p) = V(q) = 1$ en $V(r) = V(s) = 0$. Hierbij gebruiken we de semantische regels.
Omdat $V(p) = 1$ volgt met sem$\neg$ dat $V(\neg p) = 0$. Samen met $V(r) = 0$ volgt nu met sem$\leftrightarrow$ dat $V((r \leftrightarrow \neg p)) = 1$. Omdat $V(p) = 1$ en $V(s) = 0$ volgt met sem$\leftrightarrow$ dat $V((p \leftrightarrow s)) = 0$.
We hebben nu $V((r \leftrightarrow \neg p)) = 1$ en $V((p \leftrightarrow s)) = 0$. Nu volgt met sem$\leftrightarrow$ dat $V((r \leftrightarrow \neg p) \leftrightarrow (p \leftrightarrow s)) = 0$.

Opgave 6.6.4 (vi)

In deze opgave gaan we bepalen of de uitspraak $V \models q \to ((\bot \to r) \to s)$ waar is, gegeven een valuatie V met $V \models p$, $V \models q$, $V \not\models r$ en $V \not\models s$.
Met sem$\bot$ volgt $V \not\models \bot$, dus met sem$\to$ volgt dat $V \models (\bot \to r)$, onafhankelijk van de waarde van $V(r)$. Omdat $V \not\models s$, volgt nu met sem$\to$ dat $V \not\models ((\bot \to r) \to s)$. Omdat $V \models q$, volgt nu met sem$\to$ dat $V \not\models q \to ((\bot \to r) \to s)$. De uitspraak is dus onwaar.

Opgave 8.6.1 (v)

In deze opgave gaan we de geldigheid van het redeneerschema $(p \to \neg q), (p \to q) \;/\; r$ toetsen. Een redeneerschema is geldig als iedere valuatie die de premissen waar maakt, ook de conclusie waar maakt. We maken nu gebruik van een waarheidstafel.

p	q	r	$\neg q$		$(p \to \neg q)$	$(p \to q)$		r	
0	0	0	1	V_1	1	1	$*$	0	$\longleftarrow$
0	0	1	1	V_2	1	1	$*$	1	
0	1	0	0	V_3	1	1	$*$	0	$\longleftarrow$
0	1	1	0	V_4	1	1	$*$	1	
1	0	0	1	V_5	1	0		0	
1	0	1	1	V_6	1	0		1	
1	1	0	0	V_7	0	1		0	
1	1	1	0	V_8	0	1		1	

De gevallen waarbij beide premissen waar zijn, zijn aangegeven met een $*$. We zien dat bij twee van deze gevallen de conlusie onwaar is, aangegeven met een $\longleftarrow$. Het is dus niet altijd zo dat als beide premissen waar zijn, de conlusie ook waar is. Hieruit kunnen we concluderen dat het redeneerschema ongeldig is.

Opgave 9.4.7 (iii)

In deze opgave maken we een natuurlijke deductie voor de bewering $\vdash \bot \to A$. Bij het afleiden hiervan werken we van onder naar boven. Omdat het hoofdvoegteken $\to$ is, zal er waarschijnlijk sprake zijn geweest van $\to$-introductie. Hierdoor krijgen we de nieuwe hypothese $\bot$. Omdat uit het onware alles volgt, mogen we hier met behulp van $\bot$-eliminatie A uit concluderen. Als we nu onze hypothese $\bot$ opheffen, zijn we klaar met ons bewijs. Dit ziet er als volgt uit:

$$\dfrac{\dfrac{[\bot]^1}{A}\,{\scriptstyle \bot\mathrm{E}}}{\bot \to A}\,{\scriptstyle \to\mathrm{I},1}$$

Opgave 9.4.7 (ix)

In deze opgave maken we een natuurlijke deductie voor de bewering $\vdash \neg A \leftrightarrow (A \to \bot)$. We werken weer van onder naar boven. Omdat het hoofdvoegteken $\leftrightarrow$ is, zal er waarschijnlijk sprake zijn geweest van een $\leftrightarrow$-introductie. Hierdoor krijgen we de nieuwe hypothesen $\neg A$ en $(A \to \bot)$.
Laten we nu verder gaan met de linkertak, waar we $(A \to \bot)$ willen afleiden uit de nieuwe hypothese $\neg A$. Het hoofdvoegteken van $A \to \bot$ is $\to$, dus hier is waarschijnlijk sprake geweest van een $\to$-introductie. Hierdoor krijgen we de nieuwe hypothese A, die we alleen maar in de linkertak mogen gebruiken. Omdat we nu A en $\neg A$ hebben, kunnen we met behulp van $\neg$-eliminatie $\bot$ concluderen. Als we nu de hypothesen A en $\neg A$ opheffen, zijn we klaar met de linkerkant van het bewijs.
Dan de rechtertak, waar we $\neg A$ willen afleiden uit $(A \to \bot)$. Bij $\neg A$ is er waarschijnlijk sprake geweest van $\neg$-introductie, wat de nieuwe hypothese A geeft, die moet leiden tot $\bot$. Omdat we nu de hypothesen A en $(A \to \bot)$ hebben, kunnen we met behulp van $\to$-eliminatie concluderen dat dit inderdaad leidt tot $\bot$. Nu moeten we nog de hypothesen A en $(A \to \bot)$ opheffen en dan is ons bewijs compleet.
Dit ziet er als volgt uit:

$$\dfrac{\dfrac{\dfrac{[\neg A]^1 \qquad [A]^2}{\bot}\,{\scriptstyle \neg\mathrm{E}}}{A \to \bot}\,{\scriptstyle \to\mathrm{I},2} \qquad \dfrac{\dfrac{[A \to \bot]^1 \qquad [A]^3}{\bot}\,{\scriptstyle \to\mathrm{E}}}{\neg A}\,{\scriptstyle \neg\mathrm{I},3}}{\neg A \leftrightarrow (A \to \bot)}\,{\scriptstyle \leftrightarrow\mathrm{I},1}$$

Opgave 3.1.8 (c) (Inductie syllabus)

In deze opgave definiëren we de formule waarin alle negaties ($\neg$) weggelaten zijn met recursie op FOR. Deze functie zullen we neg noemen. We gebruiken $\Box$ als staande voor de binaire connectieven $\vee$, $\wedge$, $\to$ en $\leftrightarrow$.

Recursieve definitie (van de functie neg op FOR):

i. $\mathrm{neg}(p_i) := p_i$ voor alle i;

ii. $\mathrm{neg}(\bot) := \bot$;

iii. $\mathrm{neg}(\neg A) := \mathrm{neg}(A)$;

iv. $\mathrm{neg}((A \Box B)) := (\mathrm{neg}(A) \Box \mathrm{neg}(B))$.