MI-FME Cvičení 9

Tomáš Chvosta

Březen 2020

Převeďte každou z následujících základních cest na formu SSA (Static Single Assignment Form) a zapište si její podmínku ověření. Zkontrolujte, zda platí podmínka ověření. Pokud platí, napište krátký důkaz. Pokud tomu tak není, najděte protipříklad, to znamená proměnné přiřazení, které ukazuje, že vzorec neplatí, a které zároveň představuje počáteční stav, který ale po následném provedení vede k chybě.

Cvičení 9a

Zadání:

assume $x \geq 0$
$y \leftarrow x$
$z \leftarrow y$
@ $z \geq 0$

Static Single Assignment forma:

V tomto případě není potřeba zavádět nové názvy proměnných, jelikož žádná z proměnných nenabývá podruhé nové hodnoty. SSA forma má tedy následující tvar:

assume $x \geq 0$
$y \leftarrow x$
$z \leftarrow y$
@ $z \geq 0$

Logická formule z SSA:

Logická formule z SSA vypadá následovně:

$$(\forall x, y, z)([x \geq 0 \wedge y = x \wedge z = y] \Rightarrow z \geq 0)$$

Ověřovací podmínky:

Pro důkaz použijeme metodu ručního dokazování z přednášky. Předpokládáme $x \geq 0$, $y = x$, $z = y$ a dokážeme $z \geq 0$. Kvůli předpokladu $z = y$ stačí dokázat $y \geq 0$. Kvůli předpokladu $y = x$ stačí dokázat $x \geq 0$, což je ale zároveň jeden z předpokladů, takže formule platí a můžeme tvrdit, že program je napsán korektně.

Cvičení 9b

Zadání:

assume $x \geq 0$
$z \leftarrow y$
$y \leftarrow x$
@ $z \geq 0$

Static Single Assignment forma:

Můžeme si všimnout, že ve druhém kroku přiřadíme hodnotu y do z a následně přiřadíme do y novou hodnotu x. Je tedy potřeba zavést nový název proměnné pro toto přiřazení. SSA forma má tedy následující tvar:

assume $x \geq 0$
$z \leftarrow y$
$y_1 \leftarrow x$
@ $z \geq 0$

Logická formule z SSA:

Logická formule z SSA vypadá následovně:

$$(\forall x, y, y_1, z)([x \geq 0 \wedge z = y \wedge y_1 = x] \Rightarrow z \geq 0)$$

Ověřovací podmínky:

Jelikož je na první pohled zřejmé, že formule nebude logicky platná, pokusíme se najít protipříklad a tím dokázat, že formule neplatí. Můžeme využít předpokladu $z = y$ a dosadit ho do pravé strany formule. Získáme tedy závěr, že $y \geq 0$. Tedy pro $y < 0$ závěr očividně neplatí. Protipříklad tedy může vypadat například takto:

$$\{x \mapsto 8, y \mapsto -5, y_1 \mapsto 8, z \mapsto -5\}$$

Toto ohodnocení proměnných splňuje levou stranu formule, avšak nesplňuje pravou stranu. Z toho můžeme usoudit, že formule neplatí. Toto ohodnocení klidně může být i počátečním stavem. Můžeme si pro představu ukázat běh programu:

Krok	x	y	z
assume $x \geq 0$	8	-5	
$z \leftarrow y$	8	-5	-5
$y \leftarrow x$	8	8	-5
@ $z \geq 0$	8	8	-5

Z tabulky si můžeme všimnout, že ve chvíli, kdy má být hodnota proměnné $z \geq 0$ je $z = -5$. Program tedy není korektní.

Cvičení 9c

Zadání:

assume $x < 0$
$x \leftarrow x - k$
assume $k \leq 1$
@ $x \geq 0$

Static Single Assignment forma:

Můžeme si všimnout, že na začátku předpokládáme nějakou hodnotu $x < 0$ a poté ve druhém kroku přiřadíme hodnotu $x - k$ do x. Je tedy potřeba zavést nový název proměnné pro toto přiřazení. SSA forma má tedy následující tvar:

assume $x < 0$
$x_1 \leftarrow x - k$
assume $k \leq 1$
@ $x_1 \geq 0$

Logická formule z SSA:

Logická formule z SSA vypadá následovně:

$$(\forall x, x_1, k)([x < 0 \wedge x_1 = x - k \wedge k \leq 1] \Rightarrow x_1 \geq 0)$$

Ověřovací podmínky:

Při dokazování, že platí $x_1 \geq 0$ můžeme využít předpokladu $x_1 = x - k$ a dokazovat $x - k \geq 0$. To můžeme pomocí základních matematických pravidel upravit na tvar $x \geq k$. To znamená, že stačí najít protipříklad, který bude splňovat $x < k$. Protipříklad tedy může vypadat například takto:

$$\{x \mapsto -5, x_1 \mapsto -1, k \mapsto -4\}$$

Toto ohodnocení proměnných splňuje levou stranu formule, avšak nesplňuje pravou stranu. Z toho můžeme usoudit, že formule neplatí. Můžeme si pro představu ukázat běh programu:

Krok	x	k
assume $x < 0$	-5	-4
$x \leftarrow x - k$	-1	-4
assume $k \leq 1$	-1	-4
@ $x \geq 0$	-1	-4

Z tabulky si můžeme všimnout, že ve chvíli, kdy má být hodnota proměnné $x \geq 0$ je $x = -1$. Program tedy není korektní.

Cvičení 9d

Zadání:

assume $k \leq x$
$x \leftarrow x - k$
@ $x \geq 0$

Static Single Assignment forma:

Můžeme si všimnout, že ve druhém kroku přiřazujeme proměnné x novou hodnotu. Je tedy potřeba zavést nový název proměnné pro toto přiřazení. SSA forma má tedy následující tvar:

assume $k \leq x$
$x_1 \leftarrow x - k$
@ $x_1 \geq 0$

Logická formule z SSA:

Logická formule z SSA vypadá následovně:

$$(\forall x, x_1, k)([k \leq x \wedge x_1 = x - k] \Rightarrow x_1 \geq 0)$$

Ověřovací podmínky:

Pro důkaz použijeme metodu ručního dokazování z přednášky. Předpokládáme $k \leq x$, $x_1 = x - k$ a dokážeme $x_1 \geq 0$. Kvůli předpokladu $x_1 = x - k$ stačí dokázat $x - k \geq 0$, tedy $x \geq k$. To však triviálně dokazuje předpoklad $k \leq x$. Můžeme tedy tvrdit, že program je napsán korektně.

Cvičení 9e

Zadání:

$x \leftarrow x - k$
assume $k \leq x$

@ $x \geq 0$

Static Single Assignment forma:

Můžeme si všimnout, že v prvním kroku přiřazujeme proměnné x novou hodnotu. Je tedy potřeba zavést nový název proměnné pro toto přiřazení. SSA forma má tedy následující tvar:

$x_1 \leftarrow x - k$
assume $k \leq x_1$
@ $x_1 \geq 0$

Logická formule z SSA:

Logická formule z SSA vypadá následovně:

$$(\forall x, x_1, k)([x_1 = x - k \wedge k \leq x_1] \Rightarrow x_1 \geq 0)$$

Ověřovací podmínky:

Při dokazování, že platí $x_1 \geq 0$ můžeme využít předpokladu $x_1 = x - k$ a dokazovat $x - k \geq 0$. To můžeme pomocí základních matematických pravidel upravit na tvar $x \geq k$. Zároveň můžeme získat nový předpoklad složení předpokladů $x_1 = x - k$, $k \leq x_1$. Získáme předpoklad $k \leq x - k$ tedy $2k \leq x$. Máme tedy předpoklad $2k \leq x$ a závěr $x \geq k$. Můžeme si však všimnout, že nejspíše bude existovat nějaké x a nějaké k, pro které bude splněn předpoklad, ale nebude platit závěr. Pojďmě ho najít. Hledáme tedy protipříklad, kdy $2k \leq x \wedge x < k$ tedy $2k \leq x < k$. Pro úpravě získáme, že $k < 0$. Zvolme tedy například $k = -1$ a x tak, aby platilo $-2 \leq x < -1$, tedy například $x = -2$. Protipříklad tedy může vypadat například takto:

$$\{x \mapsto -2, x_1 \mapsto -1, k \mapsto -1\}$$

Toto ohodnocení proměnných splňuje levou stranu formule, avšak nesplňuje pravou stranu. Z toho můžeme usoudit, že formule neplatí. Můžeme si pro představu ukázat běh programu:

Krok	x	k
initial state	-2	-1
$x \leftarrow x - k$	-1	-1
assume $k \leq x$	-1	-1
@ $x \geq 0$	-1	-1

Z tabulky si můžeme všimnout, že ve chvíli, kdy má být hodnota proměnné $x \geq 0$ je $x = -1$. Program tedy není korektní.

Cvičení 9f

Zadání:

assume $k \geq 0$
$x \leftarrow x - k$
assume $k \leq x$
@ $x \geq 0$

Static Single Assignment forma:

Můžeme si všimnout, že ve druhém kroku přiřazujeme proměnné x novou hodnotu. Je tedy potřeba zavést nový název proměnné pro toto přiřazení. SSA forma má tedy následující tvar:

assume $k \geq 0$
$x_1 \leftarrow x - k$
assume $k \leq x_1$
@ $x_1 \geq 0$

Logická formule z SSA:

Logická formule z SSA vypadá následovně:

$$(\forall x, x_1, k)([k \geq 0 \wedge x_1 = x - k \wedge k \leq x_1] \Rightarrow x_1 \geq 0)$$

Ověřovací podmínky:

Předpokládáme $k \geq 0$, $x_1 = x - k$, $k \leq x_1$ a dokážeme $x_1 \geq 0$. Spojením prvního a třetího předpokladu získáme předpoklad $0 \leq k \leq x_1$. Závěr $x_1 \geq 0$ je ekvivalentní s $\neg\neg(x_1 \geq 0)$. Jelikož dokazujeme negaci, můžeme předpokládat $\neg(x_1 \geq 0)$ tedy $x_1 < 0$ a najít spor. Spojením předpokladu $0 \leq k \leq x_1$ s předpokladem $x_1 < 0$ získáme předpoklad $0 \leq k \leq x_1 < 0$ tedy po úpravě $0 < 0$, což je spor. Formule tedy platí a můžeme tvrdit, že program je napsán korektně.

Cvičení 9g

Zadání:

assume $x \geq 0$
$y \leftarrow x$
input x
@ $x \geq 0$

Static Single Assignment forma:

Jelikož ve třetím kroku načítáme do x novou hodnotu, je potřeba zavést nový název proměnné. SSA forma má tedy následující tvar:

assume $x \geq 0$
$y \leftarrow x$
input x_1
@ $x_1 \geq 0$

Logická formule z SSA:

Logická formule z SSA vypadá následovně:

$$(\forall x, x_1, y)([x \geq 0 \wedge y = x \wedge \top] \Rightarrow x_1 \geq 0)$$

Ověřovací podmínky:

Na pravé straně formule máme závěr $x_1 \geq 0$, nicméně v předpokladech není x_1 nijak omezeno. Je to způsobeno tím, že bezprostředně po načtení nové hodnoty do x_1 uživatelským vstupem, má být $x_1 \geq 0$. Můžeme tedy velmi snadno najít protipříklad, pro který formule neplatí, například:

$$\{x \mapsto 8, y \mapsto 8, x_1 \mapsto -1\}$$

Toto ohodnocení proměnných splňuje levou stranu formule, avšak nesplňuje pravou stranu. Z toho můžeme usoudit, že formule neplatí. Můžeme si pro představu ukázat běh programu:

Krok	x	y
assume $x \geq 0$	8	
$y \leftarrow x$	8	8
input x	-1	8
@ $x \geq 0$	-1	8

Z tabulky si můžeme všimnout, že ve chvíli, kdy má být hodnota proměnné $x \geq 0$ je $x = -1$. Program tedy není korektní.

Cvičení 9h

Zadání:

$y \leftarrow x$
input x
assume $y \geq 0$
@ $y \geq 0$

Static Single Assignment forma:

Jelikož ve druhém kroku načítáme do x novou hodnotu, je potřeba zavést nový název proměnné. SSA forma má tedy následující tvar:

$y \leftarrow x$
input x_1
assume $y \geq 0$
@ $y \geq 0$

Logická formule z SSA:

Logická formule z SSA vypadá následovně:

$$(\forall x, x_1, y)([y = x \wedge \top \wedge y \geq 0] \Rightarrow y \geq 0)$$

Ověřovací podmínky:

Máme předpoklady $y = x$, $y \geq 0$ a máme dokázat $y \geq 0$. Vidíme, že $y \geq 0$ je zároveň předpoklad i závěr, formule je tedy triviálně dokázána a můžeme tvrdit, že program je napsán korektně.

Cvičení 9i

Zadání:

$y \leftarrow x$
input x
assume $x \geq 0$
@ $y \geq 0$

Static Single Assignment forma:

Jelikož ve druhém kroku načítáme do x novou hodnotu, je potřeba zavést nový název proměnné. SSA forma má tedy následující tvar:

$y \leftarrow x$
input x_1
assume $x_1 \geq 0$
@ $y \geq 0$

Logická formule z SSA:

Logická formule z SSA vypadá následovně:

$$(\forall x, x_1, y)([y = x \wedge \top \wedge x_1 \geq 0] \Rightarrow y \geq 0)$$

Ověřovací podmínky:

Máme předpoklady $y = x$, $x_1 \geq 0$ a máme dokázat $y \geq 0$. Kvůli předpokladu $y = x$ dokazujeme $x \geq 0$. Dále však nemáme žádný předpoklad, který by nějak omezoval x. Stačí tedy najít protipříklad takový, že $x < 0$. Protipříklad tedy může vypadat například takto:

$$\{x \mapsto -1, y \mapsto -1, x_1 \mapsto 8\}$$

Toto ohodnocení proměnných splňuje levou stranu formule, avšak nesplňuje pravou stranu. Z toho můžeme usoudit, že formule neplatí. Můžeme si pro představu ukázat běh programu:

Krok	x	y
$y \leftarrow x$	-1	-1
input x	8	-1
assume $x \geq 0$	8	-1
@ $y \geq 0$	8	-1

Z tabulky si můžeme všimnout, že ve chvíli, kdy má být hodnota proměnné $y \geq 0$ je $y = -1$. Program tedy není korektní.