Semantyka i Weryfikacja Programów
Praca domowa # 1
MIMUW 2018/19

Michał Szafraniuk
219673

29 listopada 2018

Idea rozwiązania

Idea rozwiązania opiera się na tym, aby w stanach pamiętać dodatkowo "wiążący budżet" oraz dotychczas poniesiony koszt. Wiążący budżet jest najmniejszym limitem spośród limitów instrukcyj `limit` okalających rozpatrywaną instrukcję a dotychczasowy koszt jest kosztem wszystkich instrukcyj poniesionym od wejścia w pierwszą zewnętrzną dla danej instrukcji instrukcję `limit`. W ten sposób będziemy w stanie rozpoznawać sytuacje, w których wymagane jest przerwanie wykonywania instrukcji (ignorowanie).

Modyfikacje

Modyfikujemy standardową definicję stanów poprzez dodanie specjalnych, dziwnych (na tyle dziwnych, żeby nie były elementami zbioru **Var**) znaczników $\square$ oraz $\triangle$, które symbolizują odpowiednio "wiążący budżet" (w myśl znaczenia opisanego powyżej) oraz dotychczas poniesiony koszt przypisań. Dokładniej:

$$\mathbf{State} = \mathbf{Var} \cup \{\square, \triangle\} \to \mathbb{Z}_\perp$$

gdzie $\mathbb{Z}_\perp$ reprezentuje zbiór liczb całkowitych powiększony o symbol $\perp$, który będzie służył do oznaczania budżetu nieustawionego/nieograniczonego w sytuacji, gdy wykonanie programu znajduje się poza jakąkolwiek instrukcją `limit`.

Przyjmujemy, że na początku $s\square = \perp$.

W oczywisty sposób rozszerzamy relację $\leqslant$ z $\mathbb{Z}$ na $\mathbb{Z}_\perp$ jako

$$\leqslant_\perp \doteq \leqslant \cup \{\langle a, \perp\rangle : a \in \mathbb{Z}_\perp\}$$

Ale dalej aby nie zaciemniać używam po prostu $\leqslant$ w obydwu znaczeniach.

Semantyka naturalna instrukcji

Przez $\mathcal{A}$ i $\mathcal{B}$ oznaczam standardowe funkcje semantyczne dla wyrażeń, odpowiednio, arytmetycznych i boolowskich.

Instrukcja `skip`

$$\langle \texttt{skip}, s\rangle \to s$$

Instrukcja przypisania

Do rozważenia są trzy przypadki: przypisanie nie narusza budżetu, przypisanie po raz pierwszy narusza budżet lub budżet był już naruszony wcześniej.

Jeśli koszt przypisania mieści się w budżecie to aktualizujemy wartość zmiennej oraz dotychczas poniesiony koszt:

$$\langle \texttt{x:=e}, s\rangle \to s[x \mapsto \mathcal{A}[\![e]\!]s][\triangle \mapsto s\triangle + |\mathcal{A}[\![e]\!]s|] \quad \text{if} \quad s\triangle + |\mathcal{A}[\![e]\!]s| \leqslant s\square$$

Jeśli dotychczas mieściliśmy się w budżecie ale obecne przypisanie przekroczy budżet to powiększamy koszt:

$$\langle \texttt{x:=e}, s\rangle \to s[\triangle \mapsto s\triangle + |\mathcal{A}[\![e]\!]s|] \quad \text{if} \quad s\triangle \leqslant s\square,\ s\triangle + |\mathcal{A}[\![e]\!]s| > s\square$$

Jeśli budżet był już naruszony wcześniej to ignorujemy:

$$\langle \texttt{x:=e}, s\rangle \to s \quad \text{if} \quad s\triangle > s\square$$

Instrukcja złożenia

Jeśli jesteśmy w stanie przekroczenia budżetu to ignorujemy

$$\langle I_1\,\texttt{;}\,I_2, s\rangle \to s \quad \text{if} \quad s\triangle > s\square$$

a w przeciwnym przypadku standardowo

$$\frac{\langle I_1, s\rangle \to s' \qquad \langle I_2, s'\rangle \to s''}{\langle I_1\,\texttt{;}\,I_2, s\rangle \to s''} \quad \text{if} \quad s\triangle \leqslant s\square$$

Instrukcja `if`

Jeśli jesteśmy w stanie przekroczenia budżetu to ignorujemy

$$\langle \texttt{if}\ b\ \texttt{then}\ I_1\ \texttt{else}\ I_2, s\rangle \to s \quad \text{if} \quad s\triangle > s\square$$

a w przeciwnym przypadku standardowo

$$\frac{\langle I_1, s\rangle \to s'}{\langle \texttt{if}\ b\ \texttt{then}\ I_1\ \texttt{else}\ I_2, s\rangle \to s'} \quad \text{if} \quad s\triangle \leqslant s\square,\ \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

oraz

$$\frac{\langle I_2, s\rangle \to s'}{\langle \texttt{if}\ b\ \texttt{then}\ I_1\ \texttt{else}\ I_2, s\rangle \to s'} \quad \text{if} \quad s\triangle \leqslant s\square,\ \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

Instrukcja `while`

Jeśli jesteśmy w stanie przekroczenia budżetu to ignorujemy

$$\langle \texttt{while}\ b\ \texttt{do}\ I, s\rangle \to s \quad \text{if} \quad s\triangle > s\square$$

a w przeciwnym przypadku standardowo

$$\frac{\langle I, s\rangle \to s' \qquad \langle \texttt{while}\ b\ \texttt{do}\ I, s'\rangle \to s''}{\langle \texttt{while}\ b\ \texttt{do}\ I, s\rangle \to s''} \quad \text{if} \quad s\triangle \leqslant s\square,\ \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

oraz

$$\langle \texttt{while}\ b\ \texttt{do}\ I, s\rangle \to s \quad \text{if} \quad s\triangle \leqslant s\square,\ \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

Instrukcja `limit`

Jeśli jesteśmy w stanie przekroczenia budżetu lub `e` wylicza się do wartości ujemnej to ignorujemy:

$$\langle \texttt{limit}\ e\ \texttt{in}\ I, s\rangle \to s \quad \text{if} \quad \mathcal{A}[\![e]\!]s < 0 \quad \text{or} \quad s\triangle > s\square$$

W przeciwnym przypadku, jeśli $s\square = \perp$ to rozważana instrukcja `limit` nie jest zagnieżdżona w żadnej innej instrukcji `limit` więc musimy odpowiednio zainicjalizować znaczniki, w przeciwnym przypadku znacznika kosztu nie aktualizujemy a znacznik budżetu aktualizujemy tylko jeśli aktualny budżet rozważanej instrukcji `limit` jest mniejszy od odziedziczonego minimum.

Aby wyrazić to dokładniej rozpatrzmy możliwe przypadki:

- rozważany `limit` jest już okalany przez co najmniej jedną instrukcję `limit`
  - sukces I, aktualny limit mniejszy od wiążącego budżetu
    $\rightsquigarrow$ przekazujemy wgłąb budżet ustawiony na nowe minimum, odtwarzamy budżet

    $$\frac{\langle I, s[\square \mapsto \mathcal{A}[\![e]\!]s]\rangle \to s'}{\langle \texttt{limit}\ e\ \texttt{in}\ I, s\rangle \to s'[\square \mapsto s\square]} \quad \text{if} \quad s\triangle \leqslant s\square < \perp, 0 \leqslant \mathcal{A}[\![e]\!]s < s\square, s'\triangle \leqslant s'\square$$

  - sukces I, ale aktualny limit większy lub równy minimum
    $\rightsquigarrow$ standardowo

    $$\frac{\langle I, s\rangle \to s'}{\langle \texttt{limit}\ e\ \texttt{in}\ I, s\rangle \to s'} \quad \text{if} \quad s\triangle \leqslant s\square < \perp, 0 \leqslant s\square \leqslant \mathcal{A}[\![e]\!]s, s'\triangle \leqslant s'\square$$

  - porażka I, aktualny limit niemniejszy od wiążącego budżetu
    $\rightsquigarrow$ przywracamy wartość zmiennych i budżetu, ale propagujemy poniesiony koszt

    $$\frac{\langle I, s\rangle \to s'}{\langle \texttt{limit}\ e\ \texttt{in}\ I, s\rangle \to s[\triangle \mapsto s'\triangle]} \quad \text{if} \quad s\triangle \leqslant s\square < \perp, 0 \leqslant s\square \leqslant \mathcal{A}[\![e]\!]s, s'\triangle > s'\square$$

  - porażka I, aktualny limit mniejszy od wiążącego budżetu
    $\rightsquigarrow$ przekazujemy wgłąb budżet ustawiony na nowe minimum, propagujemy poniesiony koszt ale nie propagujemy aktualnego limitu

    $$\frac{\langle I, s[\square \mapsto \mathcal{A}[\![e]\!]s]\rangle \to s'}{\langle \texttt{limit}\ e\ \texttt{in}\ I, s\rangle \to s[\triangle \mapsto s'\triangle]} \quad \text{if} \quad s\triangle \leqslant s\square < \perp, 0 \leqslant \mathcal{A}[\![e]\!]s < s\square, s'\triangle > s'\square$$

- rozważany `limit` jest pierwszy
  $\rightsquigarrow$ podobnie jak wyżej, z tą różnicą, że musimy troszkę inaczej zadbać o znaczniki

  – sukces I: przekazujemy wgłąb nowy budżet i dotychczasowy koszt równy zeru, propagujemy budżet równy $\perp$

  $$\frac{\langle I, s[\Box \mapsto \mathcal{A}[\![e]\!]s][\triangle \mapsto 0]\rangle \to s'}{\langle \texttt{limit}\ e\ \texttt{in}\ I, s\rangle \to s'[\Box \mapsto \perp]} \quad \text{if} \quad s\Box = \perp \geqslant \mathcal{A}[\![e]\!]s \geqslant 0, s'\triangle \leqslant s'\Box$$

  – porażka I: udajemy, że nic się nie stało

  $$\frac{\langle I, s[\Box \mapsto \mathcal{A}[\![e]\!]s][\triangle \mapsto 0]\rangle \to s'}{\langle \texttt{limit}\ e\ \texttt{in}\ I, s\rangle \to s} \quad \text{if} \quad s\Box = \perp \geqslant \mathcal{A}[\![e]\!]s \geqslant 0, s'\triangle > s'\Box$$

Instrukcja `allow`

Pododbnie jak dla `limit` - jeśli jesteśmy w stanie przekroczenia budżetu lub `e` wylicza się do wartości ujemnej to ignorujemy:

$$\langle \texttt{allow}\ e\ \texttt{in}\ I, s\rangle \to s \quad \text{if} \quad \mathcal{A}[\![e]\!]s < 0 \quad \text{or} \quad s\triangle > s\Box$$

W przeciwnym przypadku, bez względu na to jak głęboko jesteśmy zagnieżdżeni w inne instrukcje `limit/allow` to przekazujemy znacznik budżetu ustawiony na wyliczoną wartość e oraz zerujemy znacznik kosztu.

W przypadku:

- sukcesu I koszt wykonania `allow` wynosi zero a więc przywracamy "stare" wartości poniesionego kosztu oraz budżetu:

  $$\frac{\langle I, s[\Box \mapsto \mathcal{A}[\![e]\!]s][\triangle \mapsto 0]\rangle \to s'}{\langle \texttt{allow}\ e\ \texttt{in}\ I, s\rangle \to s'[\triangle \mapsto s\triangle][\Box \mapsto s\Box]} \quad \text{if} \quad s\triangle \leqslant s\Box, \mathcal{A}[\![e]\!]s \geqslant 0, s'\triangle \leqslant s'\Box$$

- porażki I koszt wykonania `allow` równy jest przekroczonemu budżetowi:

  $$\frac{\langle I, s[\Box \mapsto \mathcal{A}[\![e]\!]s][\triangle \mapsto 0]\rangle \to s'}{\langle \texttt{allow}\ e\ \texttt{in}\ I, s\rangle \to s[\triangle \mapsto (s\triangle + s'\triangle - \mathcal{A}[\![e]\!]s)]} \quad \text{if} \quad s\triangle \leqslant s\Box, \mathcal{A}[\![e]\!]s \geqslant 0, s'\triangle > s'\Box$$