MI-FME Cvičení 15

Tomáš Chvosta

Duben 2020

Zadání

Přidejte logické formule představující assertace na řádky programu začínající symbolem @ tak, aby platily všechny ověřovací podmínky. Není potřeba cokoliv dokazovat, neformální argumenty bohatě postačí. Pokud se stane, že najdete chybu v programu (předpokládám, že ji najdete), opravte ji.

$found \leftarrow \bot$
for $i \leftarrow 0$ **to** $n - k$ **do**
    @
    $p \leftarrow \top$
    for $j \leftarrow 0$ **to** $k - 1$ **do**
        @
        if $a[i + j] \neq s[j]$ **then**
            $p \leftarrow \bot$
        @
    if p **then** $found \leftarrow \top$
    @
@ $found \Leftrightarrow \exists r \ . \ substr(a, n, s, k, r)$

Je použitá následující definice:

$$(\forall a \in \mathcal{A}, n \in \mathcal{N}, s \in \mathcal{A}, k \in \mathcal{N}, p \in \mathcal{N})$$
$$(substr(a, n, s, k, p) :\Leftrightarrow (\forall i \in \{0, \ldots, k - 1\})(p + i < n \wedge a[p + i] = s[i]))$$

Řešení

Můžeme si všimnout, že program obsahuje dva for cykly a assertace, které máme za úkol doplnit, jsou vždy na začátku a na konci těchto cyklů. Vnější for cyklus se snaží najít index i v poli a, který značí výskyt pole s. Pokud ho najde změní hodnotu proměnné $found$ na $\top$. Vnitřní cyklus kontroluje, zda se na daném indexu i v poli a všechny prvky z pole s skutečně vyskytují. Pokud ne, uloží do proměnné p hodnotu $\bot$.

V assertaci na začátku vnitřního cyklu by tedy mělo platit, že p platí, právě když pro všechny doposud zkontrolované j se prvky polí shodovali. K tomu můžeme využít definovaný predikát *substr*. To samé bude i na konci cyklu, akorát zde si musíme uvědomit, že jsme zkontrolovali o jednu hodnotu j navíc.

V assertaci na začátku vnějšího cyklu by zase mělo platit, že $found$ platí, právě když mezi doposud projitými indexi i existuje takový, který představuje počáteční index v poli a, kde se vyskytuje s. Opět můžeme využít definici predikátu *substr*.

Nesmíme opomenout ještě jednu věc. Do assertací bychom měli doplnit podmínky, že $i \leq n - k$ a také $j \leq k - 1$, abychom měli zaručeno, že ve všech základních cestách programu máme v proměnných korektní hodnoty. Program s doplněnými assertacemi bude vypadat následovně:

$found \leftarrow \bot$
for $i \leftarrow 0$ **to** $n - k$ **do**
  @ $(found \Leftrightarrow ((\exists r \in \{0, \ldots, i-1\})(substr(a, n, s, k, r)))) \wedge i \leq n - k$
  $p \leftarrow \top$
  for $j \leftarrow 0$ **to** $k - 1$ **do**
    @ $(p \Leftrightarrow substr(a, n, s, j, i)) \wedge i \leq n - k \wedge j \leq k - 1$
    if $a[i + j] \neq s[j]$ **then**
      $p \leftarrow \bot$
    @ $(p \Leftrightarrow substr(a, n, s, j + 1, i)) \wedge i \leq n - k \wedge j \leq k - 1$
  if p **then** $found \leftarrow \top$
  @ $(found \Leftrightarrow ((\exists r \in \{0, \ldots, i\})(substr(a, n, s, k, r)))) \wedge i \leq n - k$
@ $found \Leftrightarrow \exists r \,.\, substr(a, n, s, k, r)$

Po podrobnějším zkoumání programu si můžeme všimnout, že program nefunguje korektně v případě, že $n = 0$ a zároveň $k = 0$. Proměnná $found$ je triviálně $\top$, nicméně neexistuje r takové, že by splňovalo $substr(a, n, s, k, r)$. To však dokážeme jednoduše opravit upravením podmínky v assertaci na konci programu. Program pak bude vypadat následovně:

$found \leftarrow \bot$
for $i \leftarrow 0$ **to** $n - k$ **do**
  @ $(found \Leftrightarrow ((\exists r \in \{0, \ldots, i-1\})(substr(a, n, s, k, r)))) \wedge i \leq n - k$
  $p \leftarrow \top$
  for $j \leftarrow 0$ **to** $k - 1$ **do**
    @ $(p \Leftrightarrow substr(a, n, s, j, i)) \wedge i \leq n - k \wedge j \leq k - 1$
    if $a[i + j] \neq s[j]$ **then**
      $p \leftarrow \bot$
    @ $(p \Leftrightarrow substr(a, n, s, j + 1, i)) \wedge i \leq n - k \wedge j \leq k - 1$
  if p **then** $found \leftarrow \top$
  @ $(found \Leftrightarrow ((\exists r \in \{0, \ldots, i\})(substr(a, n, s, k, r)))) \wedge i \leq n - k$
@ $(found \Leftrightarrow \exists r \,.\, substr(a, n, s, k, r)) \vee ((n = 0) \wedge (k = 0))$

Po vypsání všech základních cest programu a všech logických formulí vycházejí-

cích z těchto základních cest lze snadno zjistit, že všechny ověřovací podmínky platí. Dle zadání to však už není součástí tohoto úkolu.