Částečná korektnost

Petr Štěpánek

S využitím materialu Krysztofa R. Apta

2007

Částečná korektnost je vlastností programu a znamená, že program vydává korektní výsledky pro dané dotazy. Vzhledem k tomu, dotazy pro logické programy a programy v čistém Prologu mohou být formulovány tak, že očekáváme jednu, nebo více odpovědí, je třeba pojem částečné korektnosti vyjádřit dvojím způsobem.

Motivace. Program *APPEND* může ilustrovat oba případy.

Pro dotaz `app([1,2],[3,4],Zs])` bychom chtěli dokázat, že pokud výpočet zakončí úspěšně, proměnné `Zs` bude přiřazen seznam `[1,2,3,4]`, t.j. vypočtená odpovědní substituce `{Zs/[1,2,3,4]}.`

Na druhé straně, pro dotaz `app(Xs,Ys,[1,2,3,4])` bychom rádi dokázali, že jako odpovědi dostaneme všechna možná rozdělení seznamu `[1,2,3,4]` na prefix a sufix.

To znamená, že dokážeme, že každá substituce

```
{Xs/[],Ys/[1,2,3,4]}
{Xs/[1],Ys/[2,3,4]}
{Xs/[1,2],Ys/[3,4]}
{Xs/[1,2,3],Ys/[4]}
{Xs/[1,2,3,4],Ys/[]}
```

je možná vypočtená odpověď na dotaz `app(Xs,Ys,[1,2,3,4]).`

Navíc bychom měli dokázat, že žádná jiná substituce nebude vypočtena. To je totéž jako tvrzení, že uvedená množina substitucí je totožná s množinou všech vypočtených substitucí k uvedenému dotazu.

Podobné zesílení je možné i u prvního dotazu. Zde bychom měli dokázat, že dotaz `app([1,2],[3,4],Zs])` připouští právě jednu vypočtenou odpovědní substituci `{Zs/[1,2,3,4}.`

Původní formulace zaručovala jenom, že tento dotaz připouští nejvýše jednu odpověď `{Zs/[1,2,3,4]}.`

Oba popsané typy částečné korektnosti lze formalizovat pomocí odpovědních instancí dotazu.

Definice. (částečná korektnost)

Mějme program P, dotaz $\{Q\}$ a množinu dotazů $\boldsymbol{Q}$.

- Píšeme $\{Q\}\ P\ \boldsymbol{Q}$ abychom vyjádřili fakt, že všechny vypočtené instance dotazu Q jsou prvky množiny $\boldsymbol{Q}$.
- Množinu všech vypočtených instancí Q označíme $sp(Q,P)$.

Cílem první formulace částečné korektnosti je určit formule $\{Q\}\ P\ \boldsymbol{Q}$, cílem druhé, je vypočíst množiny $sp(Q,P)$. Přitom $\{Q\}\ P\ \boldsymbol{Q}$ platí, právě když $sp(Q,P) \subseteq \boldsymbol{Q}$. První pojem korektnosti pracuje s inkluzemi dvou množin dotazů, zatímco druhý pracuje s rovností dvou množin dotazů.

Oba pojmy korektnosti se vztahují k úspěšným SLD-derivacím a ne k LD-derivacím. Nejde o významný rozdíl, protože Věta o nezávislosti na výběrovém pravidle říká, že vypočtená instance nezávisí na výběrovém pravidle.

V dalším popíšeme metody, které dovolí dokázat obě formy korektnosti.

Tedy pro program *APPEND*

$\{Q\}$ `{app([1,2],[3,4],Zs])}`

P `APPEND`

$\boldsymbol{Q}$ `{app([1,2],[3,4],[1,2,3,4]}` .

a

sp(`app(Xs,Ys,[1,2,3,4]),APPEND`) =

```
{app([],[1,2,3,4],[1,2,3,4]),
 app([1],[2,3,4],[1,2,3,4]),
 app([1,2],[3,4],[1,2,3,4]),
 app([1,2,3],[4],[1,2,3,4]),
 app([1,2,3,4],[],[1,2,3,4])
}
```

Vyjmenovány jsou všechny odpovědní instance pro program *APPEND* a dotaz `app(Xs,Ys,[1,2,3,4])`.

Podmíněné dotazy a programy

Nejprve se budeme zabývat metodou, která dovoluje dokázat částečnou korektnost prvního druhu, tedy formule $\{Q\}\ P\ \mathbf{Q}$.

Názvosloví: Atomické formuli říkáme p-atom , jestliže její predikátový symbol je p . Predikátový symbol v atomické formuli A jsme již dříve označovali jako $rel(A)$. Tedy A je $rel(A)$-atom .

Následující definice jsou velmi obecné, ale pro začátek budou vyhovovat.

Definice. (podmínky a specifikace predikátů). Mějme predikát p .

- *podmínkou* (*assertion*) *pro* p je množina p-atomů uzavřená na substituce,
- *podmínka* je podmínka pro nějaký predikátový symbol,
- říkáme, že *podmínka* $\boldsymbol{A}$ *platí pro atom* A jestliže $A \in \boldsymbol{A}$.

- *specifikace pro* p je dvojice podmínek pre_p a $post_p$ pro p .
Říkáme, že pre_p je *před-podmínka* (*pre-assertion*) a $post_p$ je *po-podmínka* (*post-assertion*) přiřazená k predikátu p ,
- *specifikace* je nějaká množina specifikací pro různé predikátové symboly.

Příklad. Uvažujme predikát `member` a jeho specifikaci

$pre_{\texttt{member}}$ = {`member(s,t)`| `t` je seznam}

$post_{\texttt{member}}$ = {`member(s,t)`| `s` je prvek seznamu `t`}

Obě podmínky jsou uzavřeny na substituce a

$pre_{\texttt{member}}$ platí pro `member(s,t)`, právě když `t` je seznam,

$post_{\texttt{member}}$ platí pro `member(s,t)`, právě když `s` je prvek seznamu `t`.

Naproti tomu množina {`member(s,t)`|`s` je proměnná a `t` je seznam} není podmínka, protože není uzavřena na substituce.

V dalším budeme předpokládat že *každý predikátový symbol má přiřazenu pevně zvolenou specifikaci.*

Tento předpoklad dovoluje mluvit o před- a po-podmínkách predikátových symbolů, kolizi specifikací dvou výskytů stejného predikátu lze řešit přejmenováním.

Volbou specifikace vyjadřujeme záměr, aby po-podmínky přiřazené k predikátovým symbolům platily pro každou vypočtenou instanci odpovídajících atomických dotazů.

V případě programu *APPEND* mimo jiné chceme, aby po-podmínka pro predikát `app` platila pro atom `app([1,2],[3,4],[1,2,3,4])`.

Obecněji, chceme, aby po-podmínka zahrnovala všechny vypočtené instance app-atomů tvaru `app(s,t,u)`kde `s,t` jsou seznamy.

Protože každá specifikace nemusí vyhovovat uvedenému záměru, klademe na specifikace další omezezující podmínky.

Tato omezení vyžadují, aby platily silnější podmínky, zejména aby před-podmínky platily pro všechny atomy vybrané pro LD-derivace a aby po-podmínky pla-tily pro vypočtené instance všech vybraných atomů.

Takové omezující podmínky jsou zobecněním pojmů dobře modovaného dotazu a dobře modované klauzule.

Definice. (Tranzitivnost)
Jsou-li dány atomy $A_1, A_2, \ldots, A_n, A_{n+1}$ a odpovídající podmínky $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_n, \boldsymbol{A}_{n+1}$, kde $n \geq 0$, píšeme

$$\models A_1 \in \boldsymbol{A}_1, \ldots, A_n \in \boldsymbol{A}_n \Rightarrow A_{n+1} \in \boldsymbol{A}_{n+1}$$

abychom vyjádřili fakt, že pro každou substituci θ

$$A_1\theta \in \boldsymbol{A}_1, \ldots, A_n\theta \in \boldsymbol{A}_n \quad \text{implikuje} \quad A_{n+1}\theta \in \boldsymbol{A}_{n+1}.$$

Notace. Zkracujeme $A \in pre_{\mathrm{rel}(A)}(A)$ na slovo $pre(A)$ a podobně slovo $post(A)$ je zkratkou za $A \in post_{\mathrm{rel}(A)}(A)$.

Tedy pro atom $p(s)$ a před-podmínku pre_p , píšeme $pre(p(s))$ jestliže $p(s) \in pre_p$.

Definice. (podmíněné dotazy a podmíněné klauzule)

(i) Dotaz $A_1 , A_2 , \ldots , A_n$ je podmíněný jestliže pro j , $1 \leq j \leq n$

$$|= post(A_1) , \ldots , post(A_{j-1}) \Rightarrow pre(A_j) .$$

(ii) Klauzule

$$H \ \text{<-}\ B_1 , B_2 , \ldots , B_n \qquad (1)$$

je podmíněná jestliže pro j , $1 \leq j \leq n+1$

$$|= pre(H) , post(B_1) , \ldots , post(B_{j-1}) \Rightarrow pre(B_j)$$

kde $pre(B_{n+1}) = post(H).$

(iii) Program je podmíněný, je-li podmíněná každá jeho klauzule.

Jinými slovy: dotaz je podmíněný, jestliže

• před-podmínka každého atomu je "implikována" konjunkcí po-podmínek předcházejících atomů.

Klauzule (1) je podmíněná , jestliže

• (j , $1 \leq j \leq n$) každá před-podmínka každého atomu B_j je „implikována" konjunkcí před-podmínky hlavy H a po-podmínkami předchozích atomů v těle.

• ($j = n + 1$) po-podmínka hlavy je „implikována" konjunkcí před-podmínkou hlavy H a po-podmínkami všech atomů v těle (1) .

Speciálně atomický dotaz A je promíněný, je-li $|= pre(A)$ a jednotková klauzule A <- je podmíněná, je-li $|= pre(A) \Rightarrow post(A)$.

Lemma. (o perzistenci podmíněnosti)

SLD rezolventa podmíněného dotazu a podmíněného programu je také podmíněná.

Důkaz postupuje stejně jako u perzistence dobré modovanosti. Opírá se o dvě pozorování.

Pozorování 1. Instance podmíněného dotazu (klauzule) je podmíněná.

Stačí si uvědomit, že podmínky jsou množiny atomů uzavřené na substituce.

Pozorování 2. Je-li **A**, H, **C** podmíněný dotaz a (1) je podmíněná klauzule, potom **A**, **B**, **C** je také podmíněný dotaz. (**B** je tělo (1).)

Důkaz Pozorování 2. Označme

$$\mathbf{A} = A_1, A_2, \dots, A_k,$$
$$\mathbf{B} = A_{k+1}, \dots, A_{k+l},$$
$$\mathbf{C} = A_{k+l+1}, \dots, A_{k+l+m}.$$

Pro každé i, $1 \le i \le k + l + m$ chceme dokázat

$$|= post(A_1), \dots, post(A_{j\text{-}1}) \Rightarrow pre(A_j). \qquad (2)$$

Uvažujeme tři případy. (a) i, $1 \le i \le n$ potom **A** je podmíněný dotaz, protože **A**, H, **C** je podmíněný dotaz.

(b) i, $k+1 \le i \le k + l$ podle předpokladu je H <- **B** podmíněná klauzule, odkud

$$|= pre(H), post(A_{k+1}), \dots, post(A_{j\text{-}1}) \Rightarrow pre(A_j).$$

Navíc **A**, H, **C** je podmíněný dotaz, tedy

$$|= post(A_1), \dots, post(A_k) \Rightarrow pre(H)$$

složením obou "implikací" dostaneme (2).

(c) $k+l+1 \leq i \leq k+l+m$. Protože **A**, H, **C** je podmíněný dotaz, platí

$|= \ post(A_1)\ , \dots , post(A_k),\ post(H)\ , post(A_{k+l+1})\ , \dots , post(A_{i\text{-}1}) \Rightarrow\ pre(A_j)$

a

$|= \ post(A_1)\ , \dots , post(A_k) \Rightarrow\ pre(H)$.

Navíc H <- **B** je podmíněná klauzule, odkud

$|= \ pre(H)\ , post(A_{k+1})\ , \dots , post(A_{k+l}) \Rightarrow\ post(H)$.

Složením posledních dvou „implikací“ dostáváme

$|= \ post(A_1)\ , \dots , post(A_k), post(A_{k+1})\ , \dots , post(A_{k+l}) \Rightarrow\ post(H)$.

Nakonec kombinací s první “implikací” v (c) dostaneme

$|= \ post(A_1)\ , \dots , post(A_{k+l})\ , post(A_{k+l+1})\ , \dots , post(A_{i\text{-}1}) \Rightarrow\ pre(A_j)$

Tím je lemma dokázáno.

Důsledek. (perzistence podmíněnosti v SLD-derivacích)

Jsou-li P a Q podmíněné, potom všechny rezolventy ve všech SLD-derivacích pro $P \cup \{Q\}$ jsou podmíněné.

Důsledek. (před-podmínky)

Jsou-li P a Q podmíněné a ξ je LD-derivace pro $P \cup \{Q\}$ potom $|= \ pre(A)$ platí pro každý atom A vybraný v ξ .

Důkaz. Pro podmíněný dotaz $A_1, A_2, \dots , A_k$, platí $|= \ pre(A_1)$. Zbytek vyplývá z předchozího důsledku o perzistenci použitého na LD-derivace.

Důsledek. (po-podmínky)

Nechť P a Q jsou podmíněné . Potom pro každou vypočtenou odpovědní instanci $A_1, A_2, \dots , A_n$, dotazu Q platí

$$|= post(A_j) \text{ pro každé } j ,\ 1 \leq j \leq n .$$