KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

JAN LAŠTOVIČKA
https://www.inf.upol.cz/lide/jan-lastovicka

LOGICKÉ PROGRAMOVÁNÍ

6. SLD-rezoluce

poznámky k přednášce, verze z 12. listopadu 2026

1 Neformální úvod

Pokud r ne relační symbol s četností nula, budeme místo $r()$ psát jen r. Chceme rozhodnout, zda platí

$$\{fire_detected, alarm_on \leftarrow fire_detected\} \models alarm_on,$$

což je ekvivalentní tvrzení, že teorie

$$T = \{fire_detected, alarm_on \leftarrow fire_detected, \square \leftarrow alarm_on\}$$

nemá model. Pro důkaz sporem předpokládejme, že $\mathcal{I}$ je jejím modelem. Protože formule $alarm_on \leftarrow fire_detected$ a $\square \leftarrow alarm_on$ jsou pravdivé, je pravdivá i formule $\square \leftarrow fire_detected$. Z pravdivosti atomu $fire_detected$ plyne, že musí být pravdivá i nepravda $\square$, což je zřejmý spor. Teorie T tedy nemá model a proto původní vyplývání platí. Důkaz lze stručně zachytit následující posloupností definitních cílů:

1. $\square \leftarrow alarm_on$
2. $\square \leftarrow fire_detected$
3. $\square$

ukončené prázdným cílem. Pro upřesnění průběhu můžeme uvést použitou posloupnost definitních klauzulí:

1. $alarm_on \leftarrow fire_detected$
2. $fire_detected$

Obecně můžeme stanovit pravidlo, že z pravdivosti definitního cíle $\leftarrow A_1, \ldots, A_n$ a definitní klauzule $A_i \leftarrow B_1, \ldots, B_m$, kde $1 \leq i \leq n$, lze odvodit pravdivost definitního cíle $\leftarrow A_1, \ldots, A_{i-1}, B_1, \ldots, B_m, A_{i+1}, \ldots, A_n$.

Uvažujme nyní definitní program

$$room_locked.$$
$$move_detected.$$
$$alarm_on \leftarrow room_locked, move_detected.$$

a definitní cíl $\leftarrow alarm_on$. Používáním právě uvedeného pravidla obdržíme posloupnost definitních cílů:

1. $\leftarrow alarm_on$

2. $\leftarrow room_locked, move_detected$
3. $\leftarrow move_detected$
4. $\square$

Vezměme program

$$
\begin{aligned}
&fire_detected.\\
&move_detected.\\
&alarm_on \leftarrow fire_detected\\
&alarm_on \leftarrow room_locked, move_detected.
\end{aligned}
$$

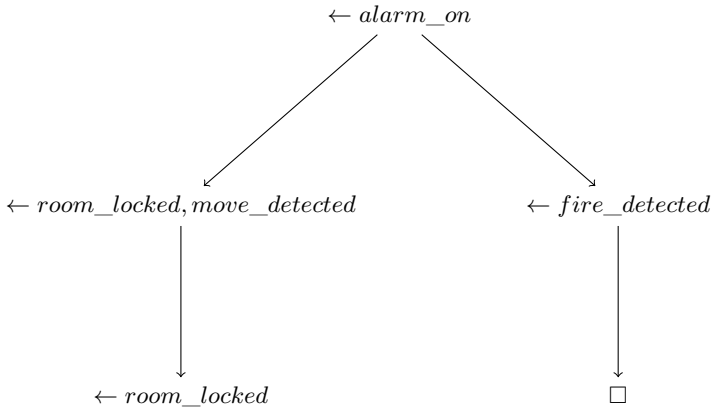
následující posloupnost k prázdnému cíli nevede:

1. $\leftarrow alarm_on$
2. $\leftarrow room_locked, move_detected$
3. $\leftarrow room_locked$

přestože jiná k němu vede:

1. $\leftarrow alarm_on$
2. $\leftarrow fire_detected$
3. $\square$

Obě posloupnosti lze přehledně zachytit stromem:



Obraťme nyní pozornost k definitnímu programu s proměnnými:

$$
\begin{aligned}
&parent(adam, abel).\\
&male(adam).\\
&father(X, Y) \leftarrow parent(X, Y), male(X).
\end{aligned}
$$

Pokud je pravdivá definitní klauzule

$$father(X, Y) \leftarrow parent(X, Y), male(X),$$

pak bude pravdivá i definitní klauzule

$$father(adam, abel) \leftarrow parent(adam, abel), male(adam).$$

Proto se dostaneme ke sporu i následující posloupností definitních cílů:

1. $\leftarrow father(adam, abel)$
2. $\leftarrow parent(adam, abel), male(adam)$
3. $\leftarrow male(adam)$
4. $\square$

Neformální úvod zakončeme příkladem definitního programu:

$$parent(anna, bert).$$
$$parent(bert, cyril).$$
$$grandparent(X, Y) \leftarrow parent(X, Z), parent(Z, Y).$$

Uvažujme definitní cíl $\leftarrow grandparent(anna, cyril)$. Nalezení sporu zachycuje posloupnost definitních cílů:

1. $\leftarrow grandparent(anna, cyril)$
2. $\leftarrow parent(anna, Z), parent(Z, cyril)$
3. $\leftarrow parent(bert, cyril)$
4. $\square$

Při přechodu z druhého k třetímu definitnímu cíli bylo nutné aplikovat substituci $\{Z/bert\}$ na $parent(Z, cyril)$.

2 SLD-odvození

Substituci $\theta = \{X_1/Y_1, \ldots, X_n/Y_n\}$ nazveme **přejmenování**, jestliže θ je vzájemně jednoznačné zobrazení. Například $\{X/X_1, Y/Y_1, X_1/X, Y_1/Y\}$ je přejmenování. **Variantou** formule φ rozumíme formuli $\varphi\theta$, kde θ je přejmenování. Například $(\forall Y)leq(zero, Y)$ je variantou definitní klauzule $(\forall X)leq(zero, X)$.

Pojmy unifikace a nejobecnějšího unifikátoru lze přímočaře definovat i pro atomické formule. Unifikační algoritmus lze jednoduše rozšířit tak, aby pracoval i s atomy. Notace $\mathrm{mgu}(A, B) = \theta$ (*the most general unifier*) znamená, že A a B jsou atomy, které lze unifikovat a θ je jejich nejobecnější unifikátor.

Nechť G je definitní cíl tvaru $\neg\exists(A_1 \wedge \cdots \wedge A_m)$ a C je definitní klauzule tvaru $\forall(B_0 \leftarrow (B_1 \wedge \cdots \wedge B_n))$ taková, že žádná proměnná se nevyskytuje současně v G i v C (množiny $\mathrm{var}(G)$ a $\mathrm{var}(C)$ jsou disjunktní). Dále nechť $1 \leq i \leq n$. Řekneme, že definitní klauzuli C **lze aplikovat** na atom A_i definitního cíle G, jestliže atomy A_i a B_0 lze unifikovat. **Aplikací** vznikne definitní cíl

$$G' = \neg\exists((A_1 \wedge \cdots \wedge A_{i-1} \wedge B_1 \wedge \cdots \wedge B_n \wedge A_{i+1} \wedge \cdots \wedge A_m)\theta)$$

nazývaný **resolventa**, kde θ je nejobecnější unifikátor A_i a B_0. Rezolventa G' značíme $\mathrm{res}(G, A_i, C, \theta)$. Odvození rezolventa se také nazývá **odvozovacím** nebo **rezolučním krokem** a můžeme jej vyjádřit pravidlem:

$$\frac{\neg\exists(A_1 \wedge \cdots \wedge A_m) \quad \forall(B_0 \leftarrow (B_1 \wedge \cdots \wedge B_n))}{\neg\exists((A_1 \wedge \cdots \wedge A_{i-1} \wedge B_1 \wedge \cdots \wedge B_n \wedge A_{i+1} \wedge \cdots \wedge A_m)\theta)} \quad \mathrm{mgu}(A_i, B_0) = \theta$$

Totéž vyjádřeno v syntaxi definitních klauzulí a cílů:

$$\frac{\leftarrow A_1, \ldots, A_m \quad B_0 \leftarrow B_1, \ldots, B_n}{\leftarrow A_1\theta, \ldots, A_{i-1}\theta, B_1\theta, \ldots, B_n\theta, A_{i+1}\theta, \ldots, A_m\theta} \quad \mathrm{mgu}(A_i, B_0) = \theta$$

Například na jediný podcíl definitního cíle $(\forall X)leq(succ(zero), X)$ lze aplikovat pravidlo

$$(\forall Z)(\forall Y)(leq(succ(Z), succ(Y)) \leftarrow leq(Z, Y)).$$

Nejobecnější unifikátor $leq(succ(zero), X)$ a $leq(succ(Z), succ(Y))$ je substituce $\{Z/zero, X/succ(Y)\}$. Aplikací vznikne cíl $(\forall Y)leq(zero, Y)$, na který lze aplikovat fakt $(\forall X)leq(zero, X)$, protože

$$\mathrm{mgu}(leq(zero, Y), leq(zero, X)) = \{Y/X\}.$$

Aplikací obdržíme prázdný cíl $\square$.

Výběrovou funkcí R rozumíme funkci, která z každého neprázdného definitního cíle vybere jeden podcíl. Například můžeme uvažovat výběrovou funkci, která vždy vybere první podcíl z neprázdného cíle. Pokud je dána výběrová funkce R, pak odvozovací krok můžeme též zapsat jako

$$G \xrightarrow[C]{\theta} G'.$$

Tedy $G' = \mathrm{res}(G, R(G), C, \theta)$.

Nechť G_0 je definitní cíl, P definitní program a R výběrová funkce. Poté **odvození** (**SLD-odvození** nebo **SLD-derivací**) definitního cíle G_0 s použitím P a R rozumíme maximální posloupnost cílů $G_0, G_1, \ldots$ spolu s posloupností $C_1, C_2, \ldots$ variant definitních klauzulí z P a posloupností $\theta_1, \theta_2, \ldots$ substitucí takovou, že G_i vznikla aplikací C_i na podcíl $R(G_{i-1})$ definitního cíle G_{i-1} s použitím θ_i. SLD-derivaci zapisujeme:

$$G_0 \xrightarrow[\theta_1]{C_1} G_1 \xrightarrow[\theta_2]{C_2} G_2 \cdots$$

nebo stručněji:

$$G_0 \xrightarrow{C_1} G_1 \xrightarrow{C_2} G_2 \cdots$$

nebo nejstručněji:

$$G_0 \to G_1 \to G_2 \cdots .$$

Název SLD-rezoluce pochází z anglického *Selective Linear Definite clause resolution*. Například:

$$\leftarrow leq(succ(zero), X) \xrightarrow[\{Z/zero, X/succ(Y)\}]{leq(succ(Z),succ(Y))\leftarrow leq(Z,Y)} \leftarrow leq(zero, Y) \xrightarrow[\{Y/X\}]{leq(zero,X)} \square.$$

Odvození, které končí prázdným cílem, nazýváme **vyvrácení** (**SLD-vyvrácení** nebo **SLD-refutace**). Například výše uvedené odvození je vyvrácení. **Zúžením substituce** θ na množinu proměnných V rozumíme substituci $\{X_1/X_1\theta, \ldots, X_n/X_n\theta\}$, kde $\{X_1, \ldots, X_n\} = \mathrm{Dom}(\theta) \cap V$. Skládání substitucí lze přirozeně zobecnit na konečnou posloupnost substitucí. Konkrétně nechť $\theta_1, \ldots, \theta_n$ jsou substituce. Pak jejich složení je substituce $\theta_1 \ldots \theta_n$, když $n > 0$, jinak je složení rovno prázdné substituci. Nechť $\theta_1, \ldots, \theta_n$ je posloupnost substitucí vyvrácení G_0. Složená substituce $\theta_1 \ldots \theta_n$ zúžená na $\mathrm{var}(G_0)$ se nazývá **vypočítanou odpovědí**. Například složme substituce u výše uvedeného vyvrácení $\{Z/zero, X/succ(Y)\}\{Y/X\} = \{Z/zero, X/succ(X), Y/X\}$, zúžením substituce na $\{X\}$ obdržíme vypočítanou odpověď $\{X/succ(X)\}$. Další příklad vyvrácení:

$$\leftarrow leq(X, succ(zero)) \xrightarrow[\{X/succ(Z),Y/zero\}]{leq(succ(Z),succ(Y))\leftarrow leq(Z,Y)} \leftarrow leq(Z, zero) \xrightarrow[\{Z/zero,X/zero\}]{leq(zero,X)} \square.$$

Protože $\{X/succ(Z), Y/zero\}\{Z/zero, X/zero\} = \{X/succ(zero), Y/zero, Z/zero\}$, je vypočítaná odpověď $\{X/succ(zero)\}$.

Konečné odvození, které není vyvrácení, se nazývá **neúspěšné odvození** (**neúspěšná derivace**). Příklad neúspěšného odvození:

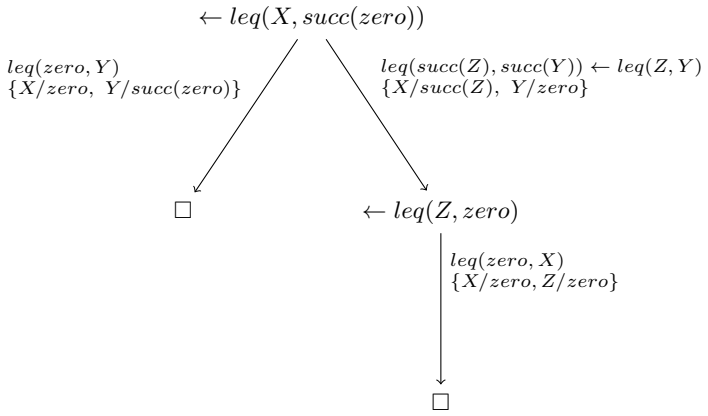
$$\leftarrow leq(succ(succ(zero)), succ(zero)) \xrightarrow[\{X/succ(zero),Y/zero\}]{leq(succ(X),succ(Y))\leftarrow leq(X,Y)} \leftarrow leq(succ(zero), zero).$$

Příklad nekonečného odvozování:

$$\begin{aligned} \leftarrow leq(X, Y) &\xrightarrow[\{X/succ(X_1),Y/succ(Y_1)\}]{leq(succ(X_1),succ(Y_1))\leftarrow leq(X_1,Y_1)} \leftarrow leq(X_1, Y_1) \\ &\xrightarrow[\{X_1/succ(X),Y/succ(Y)\}]{leq(succ(X),succ(Y))\leftarrow leq(X,Y)} \leftarrow leq(X, Y) \to \cdots \end{aligned}$$

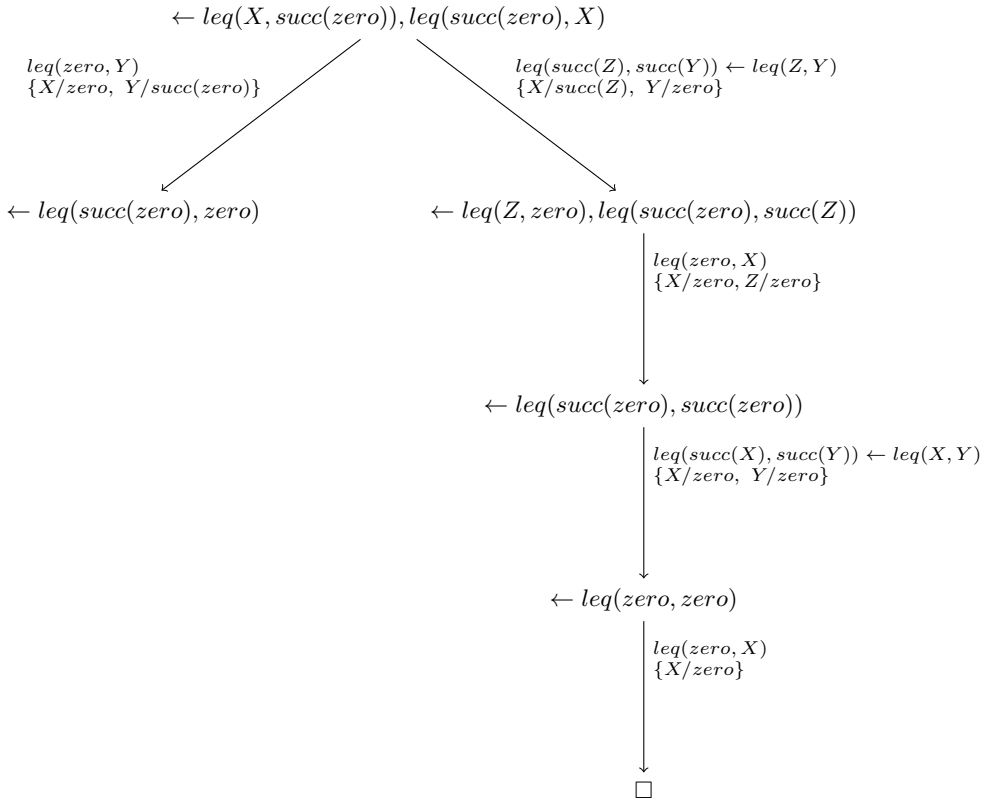
Nechť G_0 je definitní cíl, P je definitní program a S je výběrová funkce. **Odvozovací strom** (též **derivační strom** nebo **SLD-strom**) pro G_0 vzhledem k P a S je hranově i vrcholově ohodnocený strom, kde kořen je ohodnocen G_0. Následníci každého vrcholu ohodnoceného G jednoznačně odpovídají pravidlům z P, jejichž varianty lze aplikovat na G s použitím výběrové funkce S. Vrchol odpovídající variantě C pravidla z programu P je ohodnocen výsledkem aplikace C na vybraný podcíl cíle G s použitím substituce θ, hrana je ohodnocena C a θ. Odvození G_0 vzájemně jednoznačně korespondují s větvemi odvozovacího stromu.

Odvozovací strom pro $\leftarrow leq(X, succ(zero))$:



Vypočítané odpovědi jsou $\{X/zero\}$ a $\{X/succ(zero)\}$.

Dovozovací strom se zamítací větví:

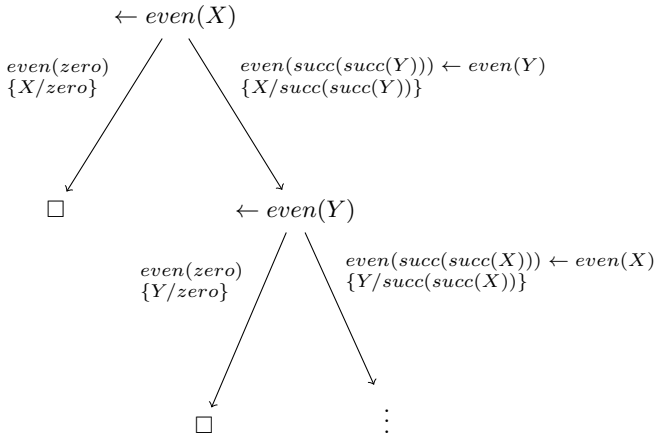


Jediná vypočítaná odpověď je $\{X/succ(zero)\}$.

V následujícím příkladu používáme definitní program:

$$even(zero).$$
$$even(succ(succ(X))) \leftarrow even(X).$$

Odvozovací strom s nekonečnou větví:



Odvozovací strom obsahuje nekonečně mnoho odpovědí: $\{X/zero\}, \{X/succ(succ(zero))\}, \ldots$