ZÁKLADY MATEMATICKEJ TEÓRIE PROGRAMOV

Časť 1.2: Vlastnosti programových schém

Igor Prívara
Inštitút informatiky a štatistiky

FMFI UK Bratislava, Katedra informatiky

Základy teórie programovania

Programové schémy – abstrakcia programov

- základné pojmy, vlastnosti programových schém
- **rozhodnuteľnosť vlastností programových schém**
- porovnávanie tried programových schém

Správnosť programov – vzhľadom na špecifikácie

- Floydova metóda a Hoareova metóda
- indukčné techniky použité pri dokazovaní správnosti
- (systematická) konštrukcia správnych programov
- dokazovanie správnosti rekurzívnych programov

Sémantika programov – formálny význam programov

- princípy operačnej, denotačnej a axiomatickej sémantiky
- algebraická štruktúra sémantických domén
- formálna definícia denotačného a operačného významu imperatívnych a rekurzívnych programov
- porovnanie denotačného a operačného významu programov

Typy a sémantika – využitie typov pri definícii sémantiky

Rozhodovacie problémy

pre triedu štandardných programových schém $\mathcal{S}$

- problém zastavenia
- problém divergencie
- problém ekvivalencie
- problém izomorfizmu

Existuje algoritmus, ktorý sa pre ľubovoľnú schému (dvojicu schém) zastaví s odpoveďou či daná vlastnosť je alebo nie je splnená?

- áno – rozhodnuteľný problém
- nie – nerozhodnuteľný problém

Existuje procedúra, ktorá sa pre ľubovoľnú schému (dvojicu schém) zastaví s odpoveďou áno keď daná schéma spĺňa danú vlastnosť, v opačnom prípade sa výpočet nemusí zastaviť (ak zastaví musí odpovedať, že vlastnosť neplatí)?

- áno – čiastočne rozhodnuteľný problém
- nie – problém nie je ani čiastočne rozhodnuteľný

Príklad:

S_1: **begin** $[y] := [a]$

1: **if** $p(y)$ **then goto** 7
2: **if** $p(y)$ **then goto** 2
3: $[y] := [f(y)]$
4: **if** $p(y)$ **then goto** 6
5: **goto end**
6: $[y] := [a]$
7: **if** $p(y)$ **then goto end**
8: $[y] := [f(y)]$
9: **goto** 7

end $[z] := [y]$

S_2: **begin** $[y] := [a]$

1: **if** $p(y)$ **then goto** 6
2: $[y] := [f(y)]$
3: **if** $p(y)$ **then goto** 5
4: **goto end**
5: $[y] := [a]$
6: **if** $p(y)$ **then goto end**
7: $[y] := [f(y)]$
8: **goto** 6

end $[z] := [y]$

Príklad:

S_3: **begin** $[y] := [x]$

1: **if** $p(y)$ **then goto end**
2: $[y] := [f(y)]$
3: **goto** 1

end $[z] := [y]$

S_4: **begin** $[y] := [x]$

1: **if** $p(y)$ **then goto end**
2: $[y] := [f(y)]$
3: **if** $p(y)$ **then goto end**
4: $[y] := [f(y)]$
5: **goto** 3

end $[z] := [y]$

S_5: **begin** $[y] := [x]$

1: **if** $p(y)$ **then goto end**
2: **if** $p(f(y))$ **then goto** 5
3: $[y] := [f^2(y)]$
4: **goto** 1
5: $[y] := [f(y)]$

end $[z] := [y]$

- S_3, S_4 sú kompatibilné, ekvivalentné a izomorfné schémy.
- S_3, S_5 sú kompatibilné, ekvivalentné ale nie sú izomorfné.

Problém divergencie

Veta – Problém divergencie v triede štandardných programových schém nie je ani čiastočne rozhodnuteľný.

Myšlienka dôkazu:

- Uvažujme triedu dvojpáskových konečných automatov $\mathcal{A}$.

  Problém "$Accept(A) = \emptyset$" pre $A \in \mathcal{A}$ nie je ani čiastočne rozhodnuteľný.

- Redukujeme triedu automatov $\mathcal{A}$ do podtriedy štandardných schém $\mathcal{S}$.

  Pre každý automat $\mathcal{A}$ skonštruujeme schému S_A z podtriedy štandardných schém $\mathcal{S}_1 \subset \mathcal{S}$ tak, že platí

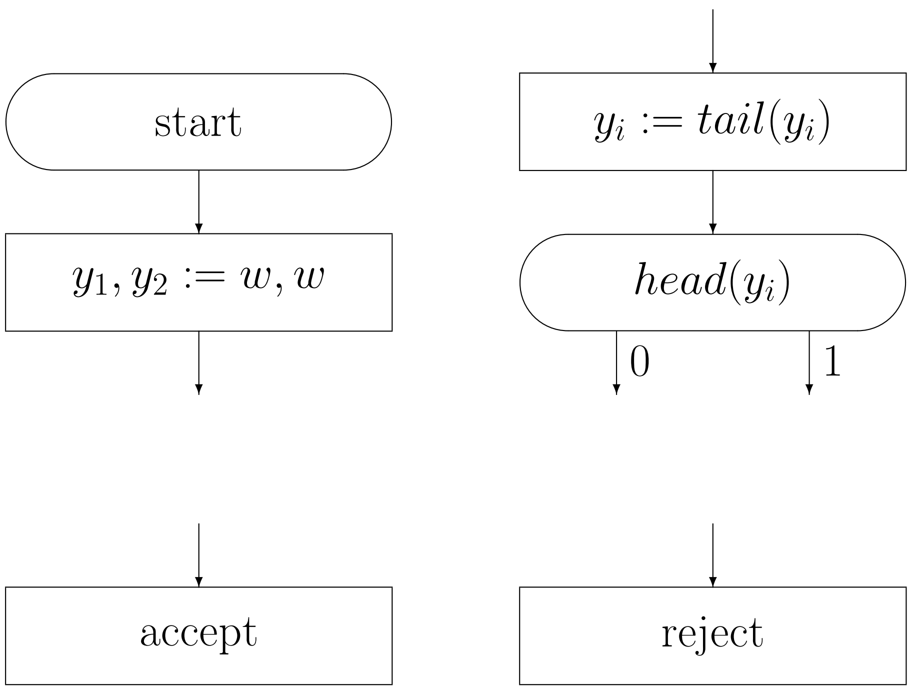
  – ku každému vstupnému slovu $w \in \{0,1\}^*$ automatu existuje Herbrandova interpretácia $\mathcal{I}_H^w$, tak že pre všetky k $(1 \leq k \leq n)$ platí
  $$i_H(p)(\text{“}f^k(a)\text{”}) = w_k,$$
  – odvodenie automatu A so vstupným slovom w je zladené s výpočtom schémy S_A s Herbrandovskou interpretáciou $\mathcal{I}_H^w$.

- Pre každý automat A a schému S_A platí

  "$Accept(A) = \emptyset$" práve vtedy, keď pre všetky Herbrandove interpretácie $\mathcal{I}_H$ program $(S_A, \mathcal{I}_H)$ diverguje.

Trieda dvojpáskových konečných automatov

- vstupy: nekonečné reťazce znakov – $\{0,1\}^\infty$
- operácie: $tail(xw) = w$, $head(xw) = x$
- stavy: počiatočný, prechodový, akceptujúci, odmietací



- $Accept(A)$ – množina akceptovaných slov
- $Reject(A)$ – množina odmietnutých slov
- $Loop(A)$ – množina slov, pre ktoré automat cyklí

Vlastnosti triedy – $\Sigma^\infty = Accept(A) \cup Reject(A) \cup Loop(A)$

- Problém “$Accept(A) = \emptyset$" nie je ani čiastočne rozhodnuteľný
- Problém “$Loop(A) \neq \emptyset$" nie je ani čiastočne rozhodnuteľný

Redukcia automatov $\mathcal{A}$ do schém $\mathcal{S}$

Podtrieda štandardných schém $\mathcal{S}_1$

Symboly – neprázdne len F^0, F^1, B^1, X_y, X_z

- $F^0 = \{a\}, F^1 = \{f\}, F^i = \emptyset$ pre $i \geq 2$
- $B^1 = \{p\}, B^i = \emptyset$ pre $i \geq 2$
- $X_x = \emptyset, X_y = \{y_1, y_2\}, X_z = \{z\}$

Príkazy – počiatočný a koncový príkaz, dvojpríkaz, večný cyklus

- **begin** $[y_1, y_2] := [a, a]$
- l: $[y_i] := [f(y_i)]$
  l+1: **if** $p(y_i)$ **then** st
- l: **goto** l
- **end** $[z] := [a]$

Herbrandove interpretácie – univerzum – $\{$"a", "$f^i(a)$"$\}$

Redukcia – ku každému dvojpáskovému konečnému automatu $A \in \mathcal{A}$ skonštruujeme schému $S_A \in \mathcal{S}_1$:

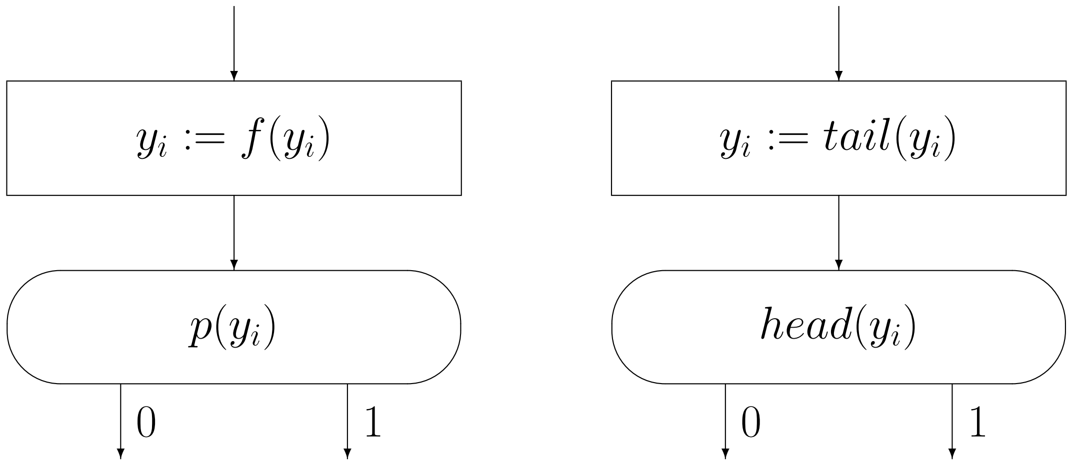
- počiatočný stav $\Longrightarrow$ počiatočný príkaz
- prechodový stav $\Longrightarrow$ dvojpríkaz
- akceptujúci stav $\Longrightarrow$ koncový príkaz
- odmietací stav $\Longrightarrow$ večný cyklus

Vlastnosti redukcie automatov na schémy

Zodpovedajúce si objekty –

- automat A so vstupným slovom w
- schéma S_A s interpretáciou $I_H^w(p)$ zladenou s w

Zladenie "výpočtu" – indukcia vzhľadom na počet modifikácií y_i



inicializácia pracovnej premennej y_i:

$$y_i := a \qquad\qquad y_i := w_0 w_1 w_2 \cdots$$

stav po prvej modifikácii pracovnej premennej y_i:

$$y_i := f(a) \qquad\qquad y_i := w_1 w_2 \cdots$$
$$i_H(p)(\text{“}f(a)\text{”}) = w_1 \qquad\qquad head(w_1 w_2 \cdots) = w_1$$

stav premennej y_i po $(k-1)$–modifikáciách:

$$y_i := f^{k-1}(a) \qquad\qquad y_i := w_{k-1} w_k w_{k+1} \cdots$$

stav po k–tej modifikácii pracovnej premennej y_i:

$$y_i := f^k(a) \qquad\qquad y_i := w_k w_{k+1} \cdots$$
$$i_H(p)(\text{“}f^k(a)\text{”}) = w_k \qquad\qquad head(w_k w_{k+1} \cdots) = w_k$$

Problém divergencie pre nejakú interpetáciu

Veta – Problém divergencie pre nejakú interpertáciu nie je v triede štandardných programových schém ani čiastočne rozhodnuteľný.

Myšlienka dôkazu:

- Uvažujme triedu dvojpáskových konečných automatov $\mathcal{A}$.

  Problém "$\mathit{Loop}(A) \neq \emptyset$" pre $A \in \mathcal{A}$ nie je ani čiastočne rozhodnuteľný.

- Redukujeme triedu automatov $\mathcal{A}$ do podtriedy štandardných schém $\mathcal{S}$.

  Pre každý automat $\mathcal{A}$ skonštruujeme schému S_A z podtriedy štandardných schém $\mathcal{S}_1 \subset \mathcal{S}$ tak, že platí

  - odmietací stav $\Longrightarrow$ koncový príkaz,
  - ku každému vstupnému slovu $w \in \{0,1\}^*$ automatu existuje Herbrandova interpretácia $\mathcal{I}_H^w$, tak že pre všetky k $(1 \leq k \leq n)$ platí
    $$i_H(p)(\text{“}f^k(a)\text{”}) = w_k,$$
  - odvodenie automatu A so vstupným slovom w je zladené s výpočtom schémy S_A s Herbrandovskou interpretáciou $\mathcal{I}_H^w$.

- Pre každý automat A a schému S_A platí

  "$\mathit{Loop}(A) \neq \emptyset$" práve vtedy, keď existuje Herbrandova interpretácia $\mathcal{I}_H$ tak, že program $(S_A, \mathcal{I}_H)$ diverguje.

Zastavenie, ekvivalencia a izomorfizmus

Problém zastavenia – Problém zastavenia štandardných schém je nerozhodnuteľný (je však čiastočne rozhodnuteľný).

Keby bol problém zastavenia rozhodnuteľný, bol by rozhodnuteľný aj problém divergencie pre nejakú interpretáciu (komplementárny problém).

Problém ekvivalencie – Problém ekvivalencie štandardných schém nie je ani čiastočne rozhodnuteľný.

Nech S_1 je ľubovoľná schéma a S_2 divergujúca schéma. Z rozhodnuteľnosti problému ekvivalencie schém by vyplývala aj rozhodnuteľnosť divergencie S_1.

Problém izomorfizmu – Problém izomorfizmu štandardných schém nie je ani čiastočne rozhodnuteľný.

Nech S_1 je ľubovoľná schéma a S_2 schéma, ktorá vznikne nahradením koncového príkazu príkazom **end goto end**. S_1 je izomorfná s S_2 práve vtedy, keď S_1 diverguje. Z rozhodnuteľnosti problému izomorfizmu schém by vyplývala aj rozhodnuteľnosť divergencie.

Podtriedy štandardných schém

Syntaktické ohraničenia: dodatočné syntaktické ohraničenia

stromové schémy – graf schémy neobsahuje cyklus,

Janovove schémy – monadické symboly nad jedinou pracovnou premennou,

konzervatívne schémy – premenná z ľavej strany priradenia sa vyskytuje aj na pravej strane priraďovacieho príkazu,

progresívne schémy – premenná z ľavej strany priraďovacieho príkazu je použitá na pravej strane nasledujúceho príkazu priradenia.

Sémantické ohraničenia: vlastnosti prípustných interpretácií

liberálne schémy – pri ľubovoľnej interpretácii $\mathcal{I}_H$ sa ten istý výraz počíta nanajvýš jeden krát,

voľné schémy – keď pre každú cestu vedúcu z počiatočného príkazu existujú interpretácia a ohodnotenie vstupných premenných také, že výpočet sleduje túto cestu,

dosiahnuteľné schémy – obsahuje len dosiahnuteľné príkazy, t.j. pre každý príkaz v schéme existuje interpretácia, pri ktorej sa príkaz vykoná,

priechodné schémy – obsahuje len dosiahnuteľné hrany, t.j. pre každú hranu v schéme existuje interpretácia, pri ktorej sa hranou "prejde".

Voľné schémy

Definícia – Schéma $S \in \mathcal{S}$ je voľná práve vtedy, keď pre každú cestu vedúcu z počiatočného príkazu existujú interpretácia $\mathcal{I}$ a ohodnotenie vstupných premenných v také, že výpočet $(S, \mathcal{I}, v)$ sleduje túto cestu.

Tvrdenie – Schéma nie je voľná práve vtedy, keď v priebehu výpočtu $(S, \mathcal{I}, v)$ sa (aspoň) dvakrát testuje ten istý predikát.

neprípustná cesta: $\cdots\ p^0(\bar{t}) \cdots\ p^1(\bar{t}) \cdots$

S_6: **begin** $[y_1, y_2] := [a, a]$
1: **if** $p(y_1)$ **then goto** 4
2: $[y_1] := [f(y_1)]$
3: **goto** 1
4: **if** $p(y_2)$ **then goto end**
5: $[y_1, y_2] := [g(y_1), f(y_2)]$
6: **goto** 4
end $[z] := [y_1]$

Vlastnosti voľných schém

- Problém "voľnosti" schémy nie je ani čiastočne rozhodnuteľný.
- Problémy divergencie, zastavenia a izomorfizmu sú rozhodnuteľné.
- Problém ekvivalencie dvoch schém je otvorený.
- Existuje schéma S, ku ktorej neexistuje ekvivalentná voľná schéma.

  Dôsledok – neexistuje algoritmus, transformujúci danú schému S do ekvivalentnej voľnej schémy.

Postov problém priradenia

Postov systém – u_i, v_i sú slová nad abecedou $\Sigma = \{a, b\}$:

$$C = \{(u_1, v_1), \cdots, (u_n, v_n)\}.$$

Riešenie Postovho systému – k–tica $i_1, i_2, \cdots, i_k$ (kde $1 \leq i_j \leq n$, $k \geq 1$) taká, že

$$u_{i_1} u_{i_2} \cdots u_{i_k} = v_{i_1} v_{i_2} \cdots v_{i_k}.$$

Príklad:

1. Štvorica $i_1 = 2, i_2 = 1, i_3 = 1, i_4 = 3$ je riešením systému

   $$C = \{(b, bbb), (babbb, ba), (ba, a)\},$$

   pretože $babbb\ b\ b\ ba = ba\ bbb\ bbb\ a$.
2. Systém $C = \{(ab, bbb), (a, ba), (b, bb)\}$ nemá riešenie, pretože $|v_i| > |u_i|$.

Postov problém priradenia – Post correspondence problem:

Má ľubovoľný Postov systém nad abecedou Σ riešenie?

Veta: Postov problém priradenia je nerozhodnuteľný (je však čiastočne rozhodnuteľný).

Dôsledok: Duálny problém nie je ani čiastočne rozhodnuteľný.

Voľnosť schémy je nerozhodnuteľná

Redukcia na Postov problém priradenia –
ku každému Postovmu systému $C_n = \{(u_1, v_1), \cdots, (u_n, v_n)\}$ nad abecedou $\Sigma = \{a, b\}$ skonštruujeme schému S_{C_n} takú, že S_{C_n} nie je voľná práve vtedy keď C_n má riešenie, t.j.

S_{C_n} je voľná práve vtedy, keď C_n nemá riešenie.

Problém voľnosti nie je ani čiastočne rozhodnuteľný –
pretože problém duálny k Postovmu problému priradenia nie je ani čiastočne rozhodnuteľný.

Redukcia –

- symboly, použité pri konštrukcii schémy S_{C_n}:
  $F^0 = \{a, b\}$, $F^1 = \{f, f_a, f_b\}$, $B^1 = \{p\}$, $X_y = \{y_1, y_2, y_3\}$,
- premenná y_3 kontroluje generovanie nejakej k–tice $i_1, \cdots i_k$,
- Herbrandovská interpretácia predikátu p zodpovedá jednej k–tici,
- pre ľubovolné slovo $w = \sigma_1 \cdots \sigma_m \in \Sigma^*$ je označenie $[y_i]{:=}[w(y_i)]$ skratkou pre priradenie $[y_i]{:=}[f_{\sigma_1}(f_{\sigma_2}(\cdots f_{\sigma_m}(y_i)\cdots))]$,
- priradenie do y_1 (y_2) simuluje postupnú kompozíciu u_i (v_i),
- “nevoľnosť" vzniká len vtedy, keď po skončení konštrukcie slov platí $y_1 = y_2$.

Schéma k Postovmu systému C_4

S_{C_4}: **begin** $[y_1, y_2, y_3] := [a, a, b]$

1: **if** $p(y_3)$ **then goto** 8
2: $[y_3] := [f(y_3)]$
3: **if** $p(y_3)$ **then goto** 10
4: $[y_3] := [f(y_3)]$
5: **if** $p(y_3)$ **then goto** 12
6: $[y_1, y_2] := [u_4(y_1), v_4(y_2)]$
7: **goto** 13
8: $[y_1, y_2] := [u_1(y_1), v_1(y_2)]$
9: **goto** 13
10: $[y_1, y_2] := [u_2(y_1), v_2(y_2)]$
11: **goto** 13
12: $[y_1, y_2] := [u_3(y_1), v_3(y_2)]$
13: $[y_3] := [f(y_3)]$
14: **if** $p(y_3)$ **then goto** 17
15: $[y_3] := [f(y_3)]$
16: **goto** 1
17: **if** $p(y_1)$ **then goto end**
18: **if** $p(y_2)$ **then goto end**
19: $[y_3] := [f(y_3)]$

end $[z] := [y_3]$

- interpretácia $p^0(b)$, $p^1(fb)$, $p^0(f^2b)$, $p^1(f^3b)$, $p^0(f^4b)$, $p^1(f^5b)$, $p^1(f^6b)$ zodpovedá trojici (2,1,1) – generuje slová $u_2u_1u_1$ a $v_2v_1v_1$,
- cesta 17–18–**end** je neprípustná ak $y_1 = y_2$, t.j. C_4 má riešenie.

Voľné schémy – divergencia, zastavenie

Divergencia – Problém divergencie voľných schém je rozhodnuteľný.

Myšlienka dôkazu – voľná schéma diverguje práve vtedy, keď v grafe schémy neexistuje cesta z počiatočného do koncového príkazu.

Ak takáto cesta existuje, musí byť "prípustná", t.j. existujú interpretácia a ohodnotenie, pri ktorých sa výpočet zastaví (spor s predpokladom).

Zastavenie – Problém zastavenia voľných schém je rozhodnuteľný.

Myšlienka dôkazu – voľná schéma sa zastaví, ak v grafe schémy neexistuje cyklus, ku ktorému existuje cesta z počiatočného príkazu.

Ak takýto cyklus existuje, existujú interpretácia a ohodnotenie, pri ktorých sa výpočet diverguje (spor s predpokladom).

Voľné schémy – izomorfizmus

Izomorfizmus – Problém izomorfizmu voľných schém je rozhodnuteľný (história – postupnosť vykonaných príkazov).

Myšlienka dôkazu – ku každej voľnej schéme priradíme regulárnu gramatiku nasledujúcou konštrukciou:

- neterminálne symboly – stavy s_i zodpovedajúce návestiam v schéme, s_b a s_e sú stavy pre návestia **begin** a **end**, počiatočný stav – s_b
- terminálne symboly – ku každému priradeniu priradíme terminálny symbol a_i (rovnaké príkazy – rovnaké symboly), ku každej podmienke p priradíme dvojicu terminálnych symbolov p^+, p^-
- pravidlá zostrojíme z príkazov schémy

begin $[\overline{y}] := [t_1(\overline{x}), \cdots, t_n(\overline{x})]$	$s_b \to a_1 s_1$
i: $[\overline{y}] := [t_1(\overline{x},\overline{y}), \cdots, t_n(\overline{x},\overline{y})]$	$s_i \to a_2 s_{i+1}$
i: **goto** k	$s_i \to s_k$
i: **if** $p(\overline{x},\overline{y})$ **then** $[\overline{y}] := [t_1(\overline{x},\overline{y}), \cdots, t_n(\overline{x},\overline{y})]$	$s_i \to p^+ a_3 s_{i+1}$
	$s_i \to p^- s_{i+1}$
i: **if** $p(\overline{x},\overline{y})$ **then goto** k	$s_i \to p^+ s_k$
	$s_i \to p^- s_{i+1}$
end $[\overline{z}] := [t_1(\overline{x},\overline{y}), \cdots, t_n(\overline{x},\overline{y})]$	$s_e \to a_4$

- $L(S)$ regulárny jazyk "generovaný" voľnou schémou S, ku každému (symbolickému) výpočtu zodpovedá práve jedno slovo v jazyku $L(S)$.
- Dve kompatibilné voľné schémy S_1, S_2 sú izomorfné práve vtedy, keď jazyky $L(S_1)$ a $L(S_2)$ generujú tie isté množiny slov.
- Z rozhodnuteľnosti problému ekvivalencie regulárnych jazykov vyplýva rozhodnuteľnosť izomorfizmu schém.

Janovove schémy

Trieda Janovových schém – syntaktické ohraničenia

- monadické schémy – $F^0 = \emptyset$ a $F^i = \emptyset, B^i = \emptyset$ pre $i > 1$
- $X_x = \{x\}, X_y = \{y\}, X_z = \{z\}$
- symboly sa aplikujú len na y

Štandardné vs. Janovove schémy – rôzna úroveň abstrakcie

- štandardné schémy umožňujú prístup k "zložke" stavu (pracovnej premennej
- Janovove schémy majú "monolitný" stav, reprezentovaný jedinou premmenou

S: **begin** $[y_1, y_2] := [x, a]$
 1: **if** $p(y_1)$ **then goto end**
 2: $[y_1, y_2] := [f(y_1), g(y_1, y_2)]$
 3: **goto** 1
end $[z] := [y_2]$

J: **begin** $[y] := [g(x)]$
 1: **if** $p(y)$ **then goto end**
 2: $[y] := [f(y)]$
 3: **goto** 1
end $[z] := [y]$

Vlastnosti Janovových schém

Vzťah Janovových a voľných schém

- Problém voľnosti Janovovej schémy je rozhodnuteľný.

  Janovova schéma nie je voľná práve vtedy, keď v nej existuje cesta prechádzajúca cez dve rovnaké testovacie podmienky, medzi ktorými nie je priradenie.

- Každá Janovova schéma sa dá preložiť do ekvivalentnej voľnej Janovovej schémy.

  Existuje tzv. úplná množina transformácií, ktorá transformuje ľubovoľnú Janovovu schému do normálneho tvaru – Janovove schémy v normálnom tvare sú voľné.

  Základná transformácia odstraňuje dva za sebou nasledujúce testy pracovnej premennej predikátom p, medzi ktorými premenná nie je modifikovaná.

Vlastnosti Janovových schém – všetky problémy rozhodnuteľné

- problém divergencie, zastavenia a izomorfizmus – dôsledok preložiteľnosti Janovových schém do voľných Janovových schém.

- problém ekvivalencie - dôkaz sa opiera o nasledujúcu vlastnosť:

  Dve Janovove schémy sú ekvivalentné práve vtedy, keď pre každú Herbrandovskú interpretáciu s ohodnotením predikátov sa buď oba výpočty nezastavia, alebo sa oba zastavia a "generujú" rovnaké reťazce $f_k \cdots f_1 f_0 x$.

Rozhodnuteľnosť vs. nerozhodnuteľnosť

$\mathcal{S}_1$ – divergencia nerozhodnuteľná: $\{y_1, y_2\}$, $\{a, f^1\}$, $\{p^1\}$

- **begin** $[y_1, y_2] := [a, a]$, **end** $[z_1, z_2] := [y_1, y_2]$
- $p(y_1)$, $p(y_2)$, $[y_1] := [f(y_1)]$, $[y_2] := [f(y_2)]$

$\mathcal{S}_2$ – divergencia rozhodnuteľná: $\{y_1, y_2\}$, $\{a, b, f^1\}$, $\{p^1\}$

- **begin** $[y_1, y_2] := [a, b]$, **end** $[z_1, z_2] := [y_1, y_2]$
- $p(y_1)$, $p(y_2)$, $[y_1] := [f(y_1)]$, $[y_2] := [f(y_2)]$

$\mathcal{S}_3$ – divergencia nerozhodnuteľná: $\{y_1, y_2\}$, $\{x\}$, $\{f^1\}$, $\{p^1\}$

- **begin** $[y_1, y_2] := [f(x), f(x)]$, **end** $[z_1, z_2] := [y_1, y_2]$
- $p(y_1)$, $p(y_2)$, $[y_1] := [f(y_1)]$, $[y_2] := [f(y_2)]$

$\mathcal{S}_4$ – divergencia nerozhodnuteľná: $\{y_1, y_2\}$, $\{x_1, x_2\}$, $\{f^1\}$, $\{p^1\}$

- **begin** $[y_1, y_2] := [x_1, x_2]$, **end** $[z_1, z_2] := [y_1, y_2]$
- $p(y_1)$, $p(y_2)$, $[y_2] := [y_1]$, $[y_1] := [f(y_1)]$, $[y_2] := [f(y_2)]$

$\mathcal{S}_5$ – divergencia rozhodnuteľná: $\{y_1, y_2\}$, $\{x_1, x_2\}$, $\{f^1\}$, $\{p^1\}$

- $p(y_1)$, $p(y_2)$, $[y_1] := [f(y_1)]$, $[y_2] := [f(y_2)]$

$\mathcal{S}_6$ – divergencia nerozhodnuteľná: $\{y_1, y_2\}$, $\{x_1, x_2\}$, $\{f^1\}$, $\{p^1\}$

- $p(y_1)$, $[y_1] := [y_2]$, $[y_1] := [f(y_1)]$, $[y_2] := [f(y_2)]$

$\mathcal{S}_7$ – divergencia rozhodnuteľná: $\{y_1, y_2\}$, $\{x_1, x_2\}$, $\{f^1, g^1\}$, $\{p^1\}$

- $p(y_1)$, $p(y_2)$, $[y_2] := [y_1]$, $[y_1] := [f(y_1)]$, $[y_2] := [g(y_2)]$

$\mathcal{S}_8$ – divergencia rozhodnuteľná: $\{y_1, y_2\}$, $\{x_1, x_2\}$, $\{f^1\}$, $\{p^2\}$

- $p(y_1, y_2)$, $[y_2] := [y_1]$, $[y_1] := [y_2]$, $[y_1] := [f(y_1)]$, $[y_2] := [f(y_2)]$