I. cast

1) Sformulujte a dokazte vysledky o (ne)rozhodnutelnosti problemu |zastavenia| v S, V, D (stand, volne a doslahnutelne schemy)

S – časť: prehľadávanie stromu výpočtu do šírky
V, D – rozhod: V – ak ∃ cesta do endu; D – end je doslanuteľný

2). Ktoru z uvedenych konstrukcii je (nie je) mozne prelozit do |ekvivalentnej schemy| z W? Preco?

W1: while b1 ⋀ b2 do S od	*nie: b1 = true => W1 zastaví, W diverguje*
W2: while b1 ⋁ b2 do S od	*áno: while b1 do S od; while b2 do S; while b1 do S od od*
W3: while not b do S od	*nie: nevieš po koľkých otočeniach máš skončiť*

3. Urcite a zdovodnite vztahy medzi triedami S, V, D a P (priechodne sch.) vzhladom na relacie podmozina a prelozitelnost.

⊆	S	V	D	P	prelož	S	V	D	P
S	T	F	F	F	S	T	F	T	F
V	T	T	F	F	V	T	T	T	T
D	T	F	T	F	D	T	F	T	T
P	T	F	T	T	P	T	F	T	T

4) Sformulovat a dokazat tvrdenie o (ne)rozhodnutelnosti prislusnosti standartnych schem k triede !volnych schem!.

Ani časť: redukciou PKP na problém voľnosti.
Princíp demonštrujeme na schéme S_{C4} zodpovedajúcej PKP C_4
*S_{C4}: **begin** $[y_1, y_2, y_3] := [a,a,b]$*
*1: **if** $p(y_1)$ **then goto 8***
2: $[y_1] := [f(y_1)]$
*3: **if** $p(y_1)$ **then goto 10***
4: $[y_3] := [f(y3)]$
*5: **if** p(y3) **then goto 12***
6: $[y_1, y_2] := [u_4(y1), v_4(y2)]$
*7: **goto 13***
8: $[y_1, y_2] := [u_1(y1), v_1(y2)]$
*9: **goto 13***
10: $[y_1, y_2] := [u_2(y1), v_2(y2)]$
*11: **goto 13***
12: $[y_1, y_2] := [u_3(y1), v_3(y2)]$
13: $[y_3] := [f(y_3)]$
*14: **if** p(y3) 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** $[z] := [y_3]$*

Premennou y_3 sa kontroluje generovanie nejakej k-tice $(l_1 \dots l_k)$. Priradenie do y_1 a y_2 simuluje postupnú kompozíciu u_i a v_i. Nevoľnosť schémy vzniká jednine v ceste 17 – 18 – end a to práve vtedy keď $y_1 = y_2$, t.j. keď má C_4 riešenie.

5) Uvazujme triedu schem L s !rozhodnutelnym! problemom doslahnutelnosti prikazu (napr. liberalne schemy). Treba dokazat rozhodnutelnost divergencie priechodnych schem z L.

Ak ∃ hrana do endu, tak nediverguje, inak diverguje (lebo je doslahnuteľná, t.j. sa ňou prejde).

6) Sformulujte a dokazte tvrdenia umoznujuce vyuzit Herbrandove interpretacie pri dokazovani zakladnych vlastnosti programovych schem.

Lema: Ku každej interpretácii I symbolov danej schémy S a každému ohodnoteniu vstupných premenných v existuje voľná interpretácia I_H, ktorá je s ňou zladená (a naopak).

Dôkaz: Nech ∀I, ∀p ∈ B': $T = T^0_{p,I} + T^1_{p,I}$, kde $T^k_{p,I} = \{(t_1,\dots,t_n): I(p)(I'(t_1),\dots,I'(t_n)) = k\}$. Interpretáciu I_H zladenú s I, definuje predpis: $I_H(p)('t_1',\dots,'t_n') = k$, ak $(t_1,\dots,t_n) \in T^k_{p,I}$. Výpočet a zladený symbolický výpočet prechádzajú tou istou cestou v diagrame danej schémy.
Lema: Nech S je schéma, I jej interpretácia, v ohodnotenie vstupov a I_H voľná interpretácia zladená s I a v. Potom pre j-te konfigurácie $[d_1,\dots,d_n]_s$ a $['t_1',\dots,'t_n']_{s^\wedge}$ výpočtov (S,I,v), $(S, I_H, 'x^\wedge')$ platí $s = s^\wedge$ a $I'(t_i) = d_i$ pre všetky i ∈ <1,n>.
Dôkaz: Indukciou vzhľadom na výpočet. V každom indukčnom kroku treba brať do úvahy všetky možné príkazy.
Veta: 1.Schéma S sa zastaví práve vtedy, keď sa zastaví výpočet pre každú Herbrandovu interpretáciu schémy S.
2.Dve kompatibilné schémy S_1, S_2 sú ekvivalentné vtedy a len vtedy, keď pre každú Herbrandovu interpretáciu I_H sa programy (S_1, I_H), (S_2, I_H) buď oba zacyklia alebo sa oba zastavia, pričom platí $val(S_1, I_H, 'x^\wedge') = val(S_2, I_H, 'x^\wedge')$.
3.Dve kompatibilné schémy S_1, S_2 sú izomorfné práve vtedy, keď pre každú Herbrandovu interpretáciu je výpočtová postupnosť stavov S_1 identická s výpočtovou postupnosťou stavov S_2.
Dôkaz: Zrejme ak platí vlastnosť pre ∀ interpretácie, platí aj pre Herbrandove. Opačná implikácia: 1.Zastavenie: Dôsledok uvedených liem. 2.Ekvivalencia: Nech $S_1 \equiv S_2$ na triede voľných interpretácií, t.j. $\forall I_H: val(S_1, I_H, 'x^\wedge') = val(S_2, I_H, 'x^\wedge')$. Potom pre ∀ interpretácie I a ohodnotenia v existuje zladená interpretácia I_H taká, že $(S_1, I), (S_1, I_H)$ alebo oba cyklia alebo $val(S_1, I, v) = I'(val(S_1, I_H, x^\wedge))$. Analogická vlastnosť platí pre S_2 (dôsledok druhej lemy). Takže programy $(S_1, I), (S_2, I)$ buď oba cyklia alebo sa oba zastavia s rovnakým výsledkom. 3.Izomorfizmus: Analógia s ekvivalenciou.

7) Sformulujte a dokazte tvrdenie o (ne)rozhodnutelnosti problemu izomorfizmu volnych schem. Uvedte definiciu historie vypoctu, pre ktore vase tvrdenie plati.

Problém izomorfizmu voľných schém je rozhodnuteľný.
Ku každej voľnej schéme S priradíme regulárnu gramatiku nasledujúcou konštrukciou. Neterminály sú stavy s_i, zodpovedajúce jednotlivým návestiam v schéme S. Terminály definujeme tým, že ku každému priradeniu priradíme terminálny symbol a_i a ku každej podmienke p priradíme dvojicu terminálov p^+, p^-. Počiatočným stavom je s_b a pravidlá zostrojíme takto:

***begin** $[y^\wedge] := [t_1(x^\wedge),\dots,t_n(x^\wedge)]$*	$s_b \to a_b s_1$
l: $[y^\wedge] := [t_1(x^\wedge,y^\wedge),\dots,t_n(x^\wedge,y^\wedge)]$	$s_l \to a_l s_{l+1}$
*l: **goto** k*	$s_l \to s_k$
*l: **if** $p(x^\wedge,y^\wedge)$ **then** $[y^\wedge] := [t_1(x^\wedge,y^\wedge),\dots,t_n(x^\wedge,y^\wedge)]$*	$s_l \to p^+ a_l s_{l+1}$
	$s_l \to p^- s_{l+1}$
*l: **if** $p(x^\wedge,y^\wedge)$ **then goto** k*	$s_l \to p^+ s_k$
	$s_l \to p^- s_{l+1}$
***end** $[z^\wedge] := [t_1(x^\wedge,y^\wedge),\dots,t_n(x^\wedge,y^\wedge)]$*	$s_e \to a_e$

Označme L(S) regulárny jazyk, generovaný uvedenou gramatikou. Je zrejmé, že v prípade voľných schém ku každej Herbrandovskej interpretácii zodpovedá práve jedno slovo v L(S). Dve kompatibilné voľné schémy S_1 a S_2 sú izomorfné práve vtedy, keď $L(S_1)$ a $L(S_2)$ generujú tie isté množiny slov. Tento problém je rozhodnuteľný => aj izomorfizmus voľných schém je rozhodnuteľný.
Použili sme históriu postupnosti stavov.

8) Dokazte, ze trieda standardnych schem S sa neda prelozit do triedy strukturovanych schem W.

Napr. voľná schéma S nemá ekvivalentnú štruktúrovanú.
*S: **begin** [y] := [x]*
*1: **if** $p_1(y)$ **then goto 3***
*2: **goto end***
*3: **if** $p_2(y)$ **then goto 5***
*4: **goto end***
5: [y] := [f(y)]
*6: **goto 1***
***end** [z] := [y]*
*Ak by existovala ekviv. štrukt. schéma W, musela by evidentne obsahovať cyklus s podmienkou p_1 (alebo p_2). Ak predpokladáme W ≡ S, tak existujú voľné interpretácie I_H pri ktorých sa cyklus **while** p_1 do ... od vykoná. Uvažujme I_H: 1. $I(p_1)(t) = 1$ pre všetky $t \in H$ 2. $I(p_2)(t) = 1$ pre tie $t \in H$, ktoré sa vyskytujú pri výpočte $(W, I_H, 'x')$, kým sa nezačne výpočet cyklu s predikátom p_1. Potom sa výpočet $(S, I_H, 'x')$ zastaví, ale $(W, I_H, 'x')$ sa zacyklí. Spor.*

9) Divergencia v, standartnych, volnych a doslahnutelnych schemach.

S – nerozhod. V – rozhod. D – rozhod.

10). Z nerozhodnutelnosti zastavenia nejakej interpretacie satndartnej schemy urcit, ci je rozhodnutelna doslahnutelnost lubovolneho priradenia.

Nerozhod. – pridáme priradenie tesne pred end. Potom ak by sme vedeli rozhodnúť dosiahnuteľnosť tohto priradenia, tak vieme rozhodnúť aj zastavenie. Spor.

11) W s countrom . Efektivna prelozitelnost S->WC

Ano, S sa da efektivne prelozit do W^1c. Problem robia prikazy goto, ktore mozu skocit uplne hocikam (vid kontrapriklad na preklad S -> W).
Schemu z S: begin

```
1: if p1(y) then goto 3
2: goto end
3: if p2(y) then goto 5
4: goto end
5: [y]:=[f(y)]
6: goto 1
end
```

Prelozime takto do W^1c:

```
W: begin
  c=1
  while c<=6 do
    if c=1 then
      if p1(y) then c=3
           else c=2
    else if c=2 then
      c=7
    else if c=3 then
      if p2(y) then c=5
                 else c=4
    else if c=4 then
      c=7
    else if c=5 then
      [y]:=[f(y)]
      c=6
    else if c=6 then
      c=1
  od
```

Skratka v c si pamatame riadok v scheme S, ktory prave vykonavame. c najprv nastavime na c=1 a vbehneme do cyklu while. Kym je c<=6 (este sme sa nedostali na koncovy prikaz), tak postupne otestujeme c a vykoname prislusny prikaz (riadok) zo schemy S. A nastavime c na dalsi riadok, ktory sa ma vykonat.

12). Dok., ze dve Janov. sch. su ekviv. prave vtedy, ked sa pre kazdu herb. interpr. bud obe nezastavia alebo sa zastavia a plati val(S1,I) = val(S2,I) = f_k f_{k-1} ... f_1 x. (pozooooooor v skriptach je to sice povedane ale nie dokazane)

13) Porovnajte triedy schem S a R.

S je (efekt.) prelož. do R (ostrá podmožina). Naspäť to nejde, lebo

```
R_A: begin [y] := [a]
       φ(y) <= if p(y) then f(y) else h(φ(g1(y)),φ(g2(y)))
     end [z] := [φ(a)]   sa nedá preložiť do S.
```

14). Bola zadana nasledujeca trieda schem:
P ::= {A | P1;P2 | if b then P1 else P2 | while b do P od }
pricom b boli predikaty intepretovane nad mnozinou stavou, A boli elementerne prikazy - funkcie interpretovane na mnozine stavov. Bolo treba urcit ci je problem divergencie rozhodnutelny v tejto triede.

Áno je, sú to vlastne W. (vraj)

15). problem zastavenia pre standardne, volne a dosiahnutelne schemy
Sformulujte a dokazte vysledky o (ne)rozhodnutelnosti problemu |divergencie|
v triede standartnych, volnych a dosiahnutelnych schem.

S – nerozhod. V – rozhod. D – rozhod.

16). (15 bodov)
Uvazujme triedu schem L s rozhodnutelnym problemom dosiahnutelnosti prikazu (napr. liberalne schemy). Dokazte, alebo vyvratte tvrdenie: Problem, ci je lubovolna schema z L priechodna je rozhodnutelny.

Je rozhod. (vraj) – ale čo ak idú do príkazu dve hrany? Ja si skôr myslím, že nie.

17) mame nejaky >loop S1 when b break S2 pool< ... skonstruujte k tomu ekvivalentnu zalezitost v strukturovanych schemach a inferencne pravidlo... dokazte ze je zdrave

S1;while not b do S2;S1 od

18) Zistite, ci je nasledujuca schema volna, resp. ci sa zastavi

```
S: begin [y1,y2,y3]:=[x,f(x),f(f(x))]
    1: [y2]:=[f(y2)]
    2: if p(y2) then goto 4
    3: goto end
    4: [y3]:=[f(y3)]
    5: [y2]:=[f(y2)]
    6: if p(y2) then goto end
    7: [y1]:=[f(y1)]
    8: if p(y1) then goto 10
    9: goto 5
   10: [y3]:=[f(y3)]
   11: if p(y3) then goto end
   12: goto 1
  end [z]:=[a]
```

19) Najdite nekonecnu mnozinu M standardnych schem taku, ze pre jej lubovolnu konecnu podmnozinu M' existuje interpretacia, pre ktoru vsetky schemy z M' zastavia, ale neexistuje interpretacia pre ktoru zastavia vsetky schemy z mnoziny M.

20) Ukazte, ze problem volnosti Janovovych schem je rozhodnutelny.

Evidentne, daná Janovova schéma nie je voľná práve vtedy, keď v nej existuje cesta obsahujúca dva rovnaké testovacie prikazy (t.j. s rovnakým predikátovým symbolom), medzi ktorými nie je priradenie.

21) Nech S je lubovolna volna schema. Tvoria vystupne termy (po odstraneni zatvoriek) cez vsetky mozne Herbrandovske interpretacie regularnu mnozinu? Aku podmienku staci pridat, aby regularnu mnozinu tvorili?

22) Keby sme pri vypocte rekurzivneho programu prepisovali term zodpovedajuci najpravejsiemu najvnutornejsiemu vyskytu funkcnej premennej mohli by sme dostat iny vysledok ako pri normalnom vyhodnocovani?

Áno, napr.
$\phi(x,y)$ <= if x<1 then 1 else $\phi(0,\phi(x,y))$
normálne vždy vráti 1, ale spomínaný postup vráti ⊥

23) Uvazujme nasledovnu rekurzivnu schemu:

```
begin
  φ(y) = if p(y) then f(y) else h(y,φ(g(y)))
end [z]:=[φ(a)]
```

Najdite ekvivalentnu standardnu schemu pouzivajucu iba tri pracovne premenne.

24) Uvazujme triedu standardnych schem, ktore zastavia pre kazdu interpretaciu. Su v triede tychto schem problemy izomorfizmu a ekvivalencie rozhodnutelne?

39/40

II. cast

1) relacnym kalkulom vyjadrite ciastocnu spravnost programu P vzhladom na podmienky p a q; vyjadrite inferencne Hoareho pravidla v relacnom kalkule

{p}P{q}	*ciastocna spravnost*
{p(x^,g(x^,y^))}y^ := g(x^,y^){p(x^,y^)}	*axioma priradenia*
$\dfrac{\{p\}S_1\{q\}\quad \{q\}S_2\{r\}}{\{p\}S_1;S_2\{r\}}$	*kompozicia*
$\dfrac{\{p\&b\}S_1\{q\}\quad \{p\&\neg b\}S_2\{q\}}{\{p\}\,\textit{if } b \textit{ then } S_1 \textit{ else } S_2 \textit{ fi}\{q\}}$	*alternativa*
$\dfrac{\{p\&b\}S\{p\}}{\{p\}\,\textit{while } b \textit{ do } S \textit{ od}\{p\&\neg b\}}$	*iteracia*
$\dfrac{p \Rightarrow p_1,\ q_1 \Rightarrow q\quad \{p_1\}S\{q_1\}}{\{p\}S\{q\}}$	*nasledok*

2) urcte postacujucu podmienku ciastocnej spravnosti schemy (samozrejme strukturovanej) pomocou Hoareho metody, p(x), q(x,z), invariant I(x,y)

Nech P je strukturovany program, p(x^) vstupny a q(x^,z^) vystupny predikat. Ak H |— {p}P{q} potom je program P ciastocne spravny vzhladom na vstupnu podmienku p a vystupnu podmienku q.

3) Definovat najslabsiu vstupnu a najsilnejsiu vystupnu podmienku a v tom kontexte vysvetlit, preco sa vo Floydovej metode pri definicii Rab a rab pouziva spatna a nie dopredna substitucia (nove, ale co to znamena, to sa ma nepytajte)

Najslabsia vstupna podmienka k programu P a vystupnej podmienke q (oznacujeme wp(P,q)):
∀p ak plati {wp(P,q)}P{q}, {p}P{q}, potom p => wp(P,q).
Najsilnejsia vystupna podmienka k prog. P a vstupnej podmienke p (oznacujeme sp(P,p)):
∀q ak plati {p}P{sp(P,q)}, {p}P{q}, potom sp(P,p) => q.

easy (n.p.)

4) loop S1; when b exit; S2; pool -> upravit na ekvivalent v strukturovanych schemach + dokazat zdravost pravidla,

$\dfrac{\{p\&b\}S\{p \Rightarrow (\sim b \Rightarrow q)\}}{\{p\}\textbf{while } b \textbf{ do } S \textbf{ od}\{q\}}$ zistit, ci je pravidlo zdrave (polonove)

5) Sformulujte Floydovou metodou postac. podmienku ciastocnej spravnosti schemy, vzhladom na vstup. podmienku p(x), vystupnu q(x,z) a invarianty I_i(x,\bar{y}) v del. bodoch. Schema vyzerala priblizne nasledovne:

```
begin [y_1,y_2]:=[x,a]
 1: [y_1]:=[g(y_1,y_2)]
 2: if p_1(y_1) then goto 5
 3: [y_1,y_2]:=[f_1(y_1),f_2(y_2)]
 4: goto 1
 5: [y_2]:=[f_2(y_1)]
 6: if p_2(y_2) then goto end
 7: [y_1,y_2]:=[g_1(y_1),g_2(y_2)]
 8: goto 1
end [z]:=[g_1(y_1)]
```

Strucne popiste Hoareovsky inferencny system na dokazovanie ciastocnej spravnosti programu a definujte pojmy: platnost a dokaz induktivnej formuly, resp. korektnost a uplnost inf. sys.

platnost formul: |= α; Formula α plati v danej triede modelov resp. v danom konkretnom modeli.
dokaz formul: |— α; Formula α je dokazatelna ak existuje postupnost formul $\alpha_1,\dots,\alpha_n$ taka, ze $\alpha = \alpha_n$, pre kazde i (1≤i≤n) bud α_i je axioma alebo existuje pravidlo $\dfrac{\alpha_{j1},\dots,\alpha_{jk}}{\alpha_i}$ take, ze vsetky α_{jm} (1≤m≤k) su clenmi postupnosti $\alpha_1,\dots,\alpha_{i-1}$

zdravost systemu: |— α => |= α
uplnost systemu: |= α => |— α

6) Rozhodnite, ci su nasledovne inf. pravidla zdrave:

$\dfrac{\{p\}\ S\ \{p\}}{\{p\}\ \textbf{repeat}\ S\ \text{until}\ b\ \{p \,\&\, b\}}$ *Ano, ekvivalentne s {p}S; while ¬b do S od{p&b}*

$\dfrac{\{p\}\ S1\ \{p\}\quad \{p\}\ S2\ \{p\}}{\{p\}\ \textbf{loop}\ S1;\ \textbf{when}\ b\ \textbf{exit};\ S2;\ \textbf{pool}\ \{p \,\&\, b\}}$ *Nie, musi byt hore {p}S1{p} {p&¬b}S2{p}*

Svoje tvrdenie dokazte, alebo vyvratte kontraprikladom.

7) Uvedte, ako sa odlisuju Floydova a Hoarova metoda z hladiska induktivnych technik, charakterizujte tieto induktivne techniky

8) Na zaklade formalneho dokazu pomocou Floydovej metody odvodte postacujuce podmienky (totalnej spravnosti) schemy S vzhladom na vstupnu podmienku p(x) a vystupnu podmienku q(x,z) a invariant I(x,y1,y2,y3).

```
S: begin [y1,y2,y3]:=[a,a,c]
  I1
   1: [y2]:=[f(y2,y3)]
   2: if t(x,y2) then goto end
   3: [y1,y3]:=[g1(y1),g2(y3)]
   4: goto 1
  end [z]:=[y1]
```

9) Spravny minsort pomocou hoareho

III cast

1) poziadavky na algebraicku strukturu semantickych domen, ich konstrukcia (a dokaz ze splnaju definiciu)

2) zadefinujte semantiky O a M, dokazte M=O 87 - 82

3) uvazujme semantiku N, ktora je rovnaka ako M, ale:
S| while b do S od | σ = najm. horne ohranicenie funkcii \psi_i,
kde \psi_0 = λσ. ⊥
\psi_{i+1} = λσ. if W|b| σ then
N|S|(\psi_i(σ)) else σ fi
Plati N=M?

4) ci je f najmensim pevnym bodom: je pevný, ale či najmenší?
f(x,y) = if x=0 then y else 1
ϕ(x,y) <= if x=0 then y else
if y=0 then ϕ(x-1,1) else
ϕ(x-1,ϕ(x,y-1))

5) zadefinujte a dokazte semanticke kriterium korektnosti pravidiel, uvedte ktore to splnaju v jazykoch L_1 a L_2

6) Definujte Vstupno/Vystupnu operacnu a denotacnu semantiku jednoducheho jazyka - priradenie, kompozicia, vetvenie, iteracia +
loop S1; when b exit; S2 pool (polonove)

7) Sformulovat a dokazat vetu o najmensom pevnom bode funkcionalu (Kleene), zistit, ci f(x)=x+y+1 je najmensim pevnym bodom funkcionalu 89
ϕ(x,y)<= if xy=0 then x+y+1 else ϕ(ϕ(x-1,1),y-1) (polonove)

8) Cpr=Fp (stare, ako jeho gate) 95

9) Zadefinujte operacnu vstupno-vystupnu semantiku a denotacnu semantiku pre triedu programov so "standardnymi" priradenim, kompoziciou a vetvenim a s cyklom: **loop S_1; if b then exit; S_2; pool**

10) Zadefinujte semantiku rekurzivnych programov + vety o tom, ze ta definicia je korektna 97

11) Rozhodnite, ci je funkcia f |najmensim pevnym bodom| funkcionalu prisluchajuceho k definicii
ϕ(x,y) = if x=0 then y
else if y=0 then ϕ(x-1,1)
else ϕ(x-1,ϕ(x,y-1))
f(x,y) : if x=0 then y else ⊥

12) Zadefinujte semanticke kriterium |korektnosti pravidla| a najdite pravidlo, ktore ho nespina, ale je korektne v jazyku L_1

13) Majme triedu striktnych relacii nad \Sigma \cup ⊥, t.j. pre kazdu relaciu R z nej plati: [⊥,y] \in R \then y=⊥. Zadefinujte na tejto triede ciastocne usporiadanie \sqsubseteq (hranata podmnozina) zodpovedajuce tomu, ze relacia R aproximuje relaciu S. Specialny pripad ked R a S su funkcie musi zodpovedat aproximacii funkcii.

14) Definujte denotacnu a vstupno/vystupnu operacnu semantiku jednoducheho jazyka (priradenie, kompozicia, vetvenie, cyklus) a dokazte, ze pre vsetky programy P plati: M||P|| = O||P||

15) Dana je nasledovna definicia vyznamu "simultanne priradenie"
$M||x_1,x_2 := t_1,t_2||\sigma = \sigma'\{x_2/V||t_2||\sigma'\}$;
$\sigma' = \sigma\{x_1/V||t_1||\sigma\}$
Uvedte kontrapriklad zdovodnujuci, ze to nie je simultanne priradenie a navrhnite spravne riesenie. Zdovodnite.

16) Sformulujte a dokazte syntakticke kriteria korektnosti vypoctovych pravidiel. Uvedte pravidlo, ktore splna toto kriterium.

17) Definovat podmienky na algebraicku strukturu semantickych domen a uviest ich konstrukciu + dokaz. Vysvetlit induktivnu
motivaciu,ktora nas viedla k tymto poziadavkam.

18) Nech σ je mnozina stavov a $\sigma^\perp = \sigma \cup \perp$. Uvazujme triedu striktnych relacii nad $\sigma^\perp$ (t.j. takych, ze (⊥,y)∈R <=> y=⊥). Definujte na striktnych relaciach vlastnost R⊑S (hranate mensie rovne), ktora vyjadruje fakt, ze R aproximuje relaciu S (specialny pripad, ked R a S su funkcie musi zodpovedat aproximacii funkcii).

19) Nech C1 a C2 su cpo.
- Dokazte tvrdenie: ak C1 je diskretne cpo, potom C1 ->s C2 je podm. C1 ->m C2
- Plati toto tvrdenie, aj ked C1 nie je diskretne cpo? Dokazte, resp. uvedte kontrapriklad pre svoje tvrdenie.

20) Uvazujme hypoteticky iterativny prikaz **loop**(b,S1,S2) definovany semantickou rovnicou M [**loop**(b, S1, S2)] = najm.h.hr ϕ_i, kde
$\phi_0 = \lambda\sigma.\perp$
ϕ_{i+1} = λσ.if W[b]σ then ϕ_i(M[S1]σ) else M[S2]σ.
Na zaklade zakladnych prikazov (priradenie, kompozicia, vetvenie, cyklus) definujte programovy segment S taky, ze plati M [**loop**(b, S1, S2)] = M [S]. Dokazte.
Zrejme to asi bolo M||S|| = ||**while** b **do** S1 **od**;S2||

21) Sformulujte a dokazte tvrdenia, ktore zarucuju, ze semantiku rekurzivnej definicie mozno definovat pomocou najmensieho pevneho bodu.

22) Overte a zdovodnite, ci je funkcia f najmensim pevnym bodom funkcionalu zodpovedajuceho rekurzivnej definicii ϕ nad oborom prirodzenych cisel.
f(x, y): if x = 0 then y else 0
ϕ(x, y) <= if x = 0 then y else if y = 0 then ϕ(x-1,1) else ϕ(x-1,ϕ(x,y-1))

23). f(x):if x>100 then x-10 else gulova sedma(alebo cervene eso?) ci je najmensi pevny bod pre ϕ<=if x > 100 then x-10 else ϕ(ϕ(x+11))

Udajny pevny bod bol f(x)=if x>100 then x-10 else _. Staci vsak vypocitat f(100) a ϕ(100) a vyjdu rozne vysledky.

24). Nebezpecne korektne pravidlo.

Take pravidlo je napr. SPOS - sekvencny POS. Najprv vyberie vsetky vyskyty ϕ, ktore bude nahradzat a potom ich jeden po druhom (rozumnej postupne) nahradi. SPOS je korektny, lebo po konecne vela krokoch nahradi vsetky vyskyty ϕ presne ako POS (a ten je korektny). SPOS ale nie je bezpecny. Niektora z tych substitucii nemusi byt bezpecna (vid definicia bezpecnej substitucie), lebo moze substituovat "nepotrebne" ϕ.

25) definovat operacnu semantiku C: Stat -> (σ -> σ*)

26) Overte a zdovodnite, ci je funkcia f |najmensim pevnym bodom| funkcionalu zodpovedajuceho rekurzivnej definicii ϕ nad oborom N
f(x,y): if xy=0 then x+y+1 else ⊥
ϕ(x,y) <= if xy=0 then x+y+1 else ϕ(ϕ(x-1,1),y-1)

27) Definujte poziadavky na algebraicku strukturu semantickych domen a uvedte ich konstrukciu vratane dokazu, ze domeny maju pozadovane vlastnosti. Vysvetlite intuitivnu motivaciu, kt. nas priviedla k tymto poziadavkam.

28) Uvazujte semantiku charakterizovanu:
N[**while** b **do** S **od**]= U^∞_0 Psi_i
kde
Psi_0 = λσ.⊥
Psi_i+1 = λσ.if W[b]σ then N[S](Psi_i(σ)) else σ
Plati |M| = N ? Svoje tvrdenie zdovodnite

29) Overte a zdovodnite, ci je funkcia f |najmensim pevnym bodom| funkcionalu zodpovedajuceho rekurzivnej definicii ϕ nad oborom N
f(x,y): if x=0 then y else 1
ϕ(x,y) <= if x=0 then y else if y=0 then ϕ(x-1,1) else ϕ(x-1,ϕ(x,y-1))

30) |spojitost funkcionalu|,|prirodzene rozsirenie| a povedat o nejakych dvoch ci su spojite