Programozási módszertan elmélete tárgy első féléve (formázott kézirat ,1. változat)

EA2

1. Szignatura: $\Sigma = (S, OP)$

S: szortok halmaza;
OP: konstans es operacios szimbolumok halmaza;
$OP = K\_S \cup OP\_W,S$; s reszhalmaza S;
K_S: konstans szimbolumok halmaza;
OP_w,s operacios szimbolumok halmaza;
w argument szort, $w \in S^+$;
s eredmeny szort, $s \in S$;
K_s ; OP_w,s paronkent diszjunktak;

$K = U\_(s \in S)\ K\_s$; $OP = U\_(w \in S^+, s \in S)\ OP\_w,s$

$N \in K\_s$; $N: \to s$;

$N \in OP\_w,s$, $w=s\_1 \ldots s\_n$; akkor $N= s\_1 \ldots s\_n \to s$;

□

2. Pelda. Σ = ({S1, S2}, {N1:S1→S2, N2:S2 S1 → S2, N3: S1 S2 → S2, N4:S2 S1 S2 → S2});
A=({alphabet, bintree};
{leaf: alphabet → bintree, left: bintree alphabet → bintree, right: alphabet bintree → bintree, both: bintree alphabet bintree → bintree});

□

3. Adott Σ = (S, OP) szignaturahoz tartozo Σ algebra:
A = (S_A, OP_A), ahol S_A = (A_S)_($s \in S$) es N=(N_A)_($N \in OP$);
1. A_S, A bazishalmaza minden $s \in S$;
2. $N\_A \in A\_s$;
   minden $N \in K\_s$: $N \to s$ es $s \in S$ konstans szimbolumra;
   minden N: OP_{s1 ... sn → s} es s_1 ... s_n $\in$ S^+; $s \in S$ muveleti szimbolumra;

Megjegyzes: Ha Σ=(S1,...S_n, N1, ..., N_m); akkor A=(A_s1, ..., A_sn, N1_A, ..., Nm_A);
(megfeleltetesi sorrendben!)

□

5. Valtozo

Adott Σ = (S, OP), es X_s, $s \in S$, az s szorthoz tartozo valtozok halmaza.
X = $\cup$_{s$\in$ S} X_s, a Σ szignaturahoz tartozo valtozok halmaza.

Deklaracio: x, y $\in$ X_s;
Jelolesunk:**oprs**:x, y $\in$ S;

□

6. Pelda

Σ = (S, OP);
S=natbool;
natbool = nat $\cup$ bool;
n, m $\in$ X_{nat}   a, b, c $\in$ X_{bool}
Tétel: n, m $\in$ nat; a, b, c $\in$ bool;

□

7. Term szintaktikai definicioja:

Adott Σ = (S, OP); X a szignaturahoz tartozo valtozo.

T_{Σ(X)} = (T_{Σ(X),s}) s $\in$ S; definicioja:
Bazis termek:
X_s $\in$ T_{Σ(X),s};
n:→s $\in$ OP; akkor n $\in$ T_{Σ(X),s};
Osszetett termek:
n:s_1 ... s_k → s, k>=1, n $\in$ OP, t_1 $\in$ T_{Σ(X),s}, 1<=i<=k;
akkor n(t_1, ..., t_k) $\in$ T_{Σ(X), s};

(megjegyzes: szig-szig-szig-szignatura)

ezek a tomb definicioi (hoppa)

□

8. T_{Σ(X)} mas jelolesei: T_{OP}(X); T_Σ(X); T_{Σ(X)}; T_Σ; T_{OP};

Pelda:

nat0 =
sorts: nat   0 $\in$ T_{nat};
oprs:

$0: \to nat$
$succ^k(0) \in T_{nat}$;
$succ: nat \to nat$ $\quad k = 1, 2, ...$
eqns:
$n \in nat$;

(megjegyzes, a E-k nagy resze nagy szigma es valahogy igy kell irni:

```
----
\
 I
/
----  )
```

9. pelda: bintree.

$T_{alph} = \{k_1, ..., k_n\}$;
Minden $a \in T_{alph}$, $t_1, t_2 \in T_{bin}$; akkor
$leaf(a) \in T_{bin}$;
$left(t_1, a) \in T_{bin}$; $right(a, t_1) \in T_{bin}$; $both(t_1, a, t_2) \in T_{bin}$;

10. Strukturalis indukcio:

Adott $\Sigma = (S, OP)$; a szignaturahoz tartozo X valtozokkal.
Legyen p predikatum, amely a $t \in T_{op}(X)$ termekre van ertemezve, ha
1. $(\forall t \in K$ es $\forall t \in X)(p(t) =$ "true");
2. $(\forall N(t1, ..., tn) \in T_{op}(X))(p(N(t_1, ..., t_n) =$ "true");

akkor
$(\forall t \in T_{op}(X))(p(t) =$ "true").

11. A term kiertekelese

1. Adott $\Sigma = (S, OP)$; es a T_{op}; Legyen A egy Σ-algebra.
A kiertekeles
$eval: T_{op} \to A$;
definicioja:
$eval(N) = N_A$; ha $N_A \in K$;
$eval(N(t_1, ..., t_n)) = N_A(eval(t_1), ..., eval(t_n))$, ha $N(t_1, ..., t_n) \in T_{op}$.

12. Term (ertekadas) kiertekelese.

2. Adott $\Sigma = (S, OP)$; a szignaturahoz tartozo X valtozokkal, es
T_{op}; Legyen A egy Σ-algebra.
$ass: X \to A$, ahol $ass(x) \in A_s$, $x \in X_s$, $s \in S$;
$ass: T_{op}(x) \to A$; definicioja;
$ass(x) = ass(x)$, $x \in X$ valtozo;
$ass(N) = N_A$, $N \in K$ konstans szimbolum;
$ass(N(t_1, ..., t_n)) = N_A(ass(t_1), ..., ass(t_n))$;
$N(t_1, ..., t_n) \in T_{op} (X)$;

13. Eval-pelda

1.) Adva
nat1 = nat0 +
oprs: add: nat nat $\to$ nat ZOLD

13. Kiertekeles a N termeszetes szamok koreben:
$eval(add(succ(0), succ(0))) = eval(succ(0)) + eval(succ(0)) = (eval(0)+1)+(eval(0)+1) = (0+1) + (0+1) = 2$.
SARGA

Adva $X = \{n, m\}$ es $ass(n)=1$; $ass(m)=5$.
$ass(add(succ(n), m) = ass(succ(n)) + ass(m) = (ass(n)+1)+ass(m)) = (1+1)+5 = 7$.
KEK

14. Egyenletek

Adott $\Sigma = (S, OP)$; a szignaturahoz tartozo X valtozokkal.
- Az $e = (X, L, R)$ harmast, $L, R \in T_{OP, s}(X)$, $s \in S$ mellett
egyenletnek nevezzuk.

- Az $e = (X, L, R)$ egyenlet helyes az A Σ-algebraban, ha minden $ass: X \to A$ eseten $ass(L) = ass(R)$.

Specifikacio.
$SPEC = (S, OP, E)$;
$\Sigma = (S, OP)$; $E = \{e(X, L, R)\}$; $\forall x \in X, L=R$;

X valtozok halmaza, L, R, termek X-bol vett valtozokkal.

□

15. Type

<tipus neve> (<parameterek listaja>) **is a type specification =**
  parameters = <atvett aktualis tipusnev_1> **+** ... + <atvett aktualis tipusnev_k> **+**
    sorts: <formalis parameterek nevei>;
    oprs: <muveletek formai>;
    eqns: <muveletek jelentesenek leirasa>;
  export =
    type sort: <tipushalmaz neve>;
    oprs: <muveletek formai>;
    eqns: <muveletek jelentesenek leirasa>;
end <tipus neve>;

□

16. Σ-algebrak kozti homomorfizmus.

Legyenek A = (S_A, OP_A) es B = (S_B, OP_B) azonos
Σ = (S, OP) szignaturaju algebrak.

1. A h: A → B homomorfizmus egy fuggvenycsalad
   h = (h_s)_{s ∈ S}

   ahol
   h_s: S_A → S_B,

   ugy, hogy
   - minden N: → s ∈ OP es s ∈ S konstans szimbolumra teljesul: h_s(N_A) = N_B
   - valamint minden N: s_1 ... s_k → s_l ∈ OP es minden i=1, ..., k-ra es a_i ∈ A_{si}
     eseten teljesul a homomorfikus feltetel, azaz
     h_s(N_A(a_i...a_k)) = N_B(h_{si}(a_i) ... h_{sk}(a_{ik})).

□

17. A bijektiv homomorfizmust izomorfizmusnak nevezzuk.

Az A es B Σ-algebrakat izomorfikusnak nevezzuk, ha letezik az izomorfizmus A → B eseten es jelolese ekkor: A ≅B.

Megjegyzesek:

Homorfizmusok kompozicioja szinten homomorfizmus.
Ha h_s izomorfizmus, akkor h_s^{-1} is az.

Σ = (S, OP); OP = {k1:→ S, k2:→S, N1:S→S; N2: S S → S})

A = (bool, {T: → bool, F:→bool, ¬_:bool → bool, ∧ : bool bool → bool [infix]});
B = (bit, {1: → bit, 0: → bit, ch: bit → bit, *: bit bit → bit [infix]});

□

18. Zart

A→B; ugy, hogy
h_s: bool → bit; : h_{s1}(T) = 1; h_{s1}(F) = 0;
h_s(A(w)(a_1, ..., a_n)) = B(w)(h_{s1}(a_1), ..., h_{sn}(a_n))
h_s(¬B) = ch(h_{s1}(B)) = if B = T then ch(1) else ch(0) fi;
h_s(B1 ∧ B2) = h_{s1}(B1) * h_{s1}(B2) = if B1=T ∧ B2=T then h_{s1}(T)*h_{s1}(T) else h_{s1}(T) * h_{s1}(F) fi;

    1   0
h_{s1}(T) h_{s1}(F)

Egy adott Σ szignaturahoz tartozo absztrakt adattipus a Σ-algebrak egy olyan osztalya
amely az izomorfizmusra zart: C ⊂ Alg(Σ) es A ∈ C es A ≅ B ⇒ B ∈ C.

EA3

Problema: mit jelent
  formalis tipusspecifikacio → aktualis tipusspecifikacio ?
    specifikacio morfizmus.

Specialis esetei:
1) Atnevezes.
2) Benne foglaltatas, tartalmazas, bovites.
3) Abrazolas, reprezentacio.
4) Parameter atadas.
   Specialisan, formalis parameterek helyettesitese aktualis
   parameterekkel (parameter passing):
   spec(spec_1) → spec(spec_2).

□

Megjegyzesek.
1. Standard parameteratadas:
   spec(spec_1) → spec(spec_2)
   spec_1 formalis   spec_2 aktualis parameter
   ertekadassal torteno specifikacio,

2. Ismetelt parameteratadas:
$spec(spec_1(spec_A)) \rightarrow spec(spec_2(spec_B));$

Szignaturamorfizmus

Adott: $\Sigma = (S, OP)$, $\Sigma' = (S', OP')$, mellett a
$h_\Sigma:\Sigma \rightarrow \Sigma'$ lekepezest szignaturamorfizmusnak nevezzuk, ha
$h_\Sigma = (h_S:S \rightarrow S', h_{OP}: OP \rightarrow OP');$
ugy, hogy
$(\forall N: s_1 \ldots s_n \rightarrow s \in OP)(h_{OP}(N):h_S(s_1)\ldots h_S(s_n) \rightarrow h_S(s) \in OP').$

Kituntetett sortu szignaturamorfizmus:
$h_s(pt(\Sigma)) = pt(\Sigma').$

Pelda: kituntetett elemu szignatura morfizmusra a verem eseteben:
$pt(\Sigma) = stack$; $pt(\Sigma') = vector\ nat$;
$h_S(stack) = vector\ nat.$

sorts: <uj nev> = <regi nev>

A $h: \Sigma \rightarrow \Sigma'$ szignaturamorfizmus kiterjesztese valtozokra:

Legyenek X, X' rendre a Σ, Σ' valtozoi.
Tovabbiakban altalaban:
$h=h_{\Sigma}$; $h(s) = h_s$; $h(N) = h_{OP}(N)$;
$h_X:(\cup_{s \in S} X_S) \rightarrow (\cup_{s' \in S'} X'_{S'})$

ha $x \in X_S$, $s \in S$, akkor $h_X(x) \in X'_{h(s)=s'}$;

A $h: \Sigma \rightarrow \Sigma'$ szignaturamorfizmus kiterjesztese termekre:
Adottak: $T_{\Sigma(X)}$, $T_{\Sigma'(X')}$ rendere a Σ, Σ'
termeknek halmazai.
Minden $t \in T_{\Sigma(X)}$-hez tartozo $h_T(t) \in T_{\Sigma'(X')}$ definicoja:
- $(\forall x \in X)(h_T(x)=h_X(x));$
- $(\forall (N: \rightarrow s) \in OP)(h_T(N: \rightarrow s)=h_X(h(N): \rightarrow s);$
- $(\forall N:s_1 \ldots s_n \rightarrow s) \in OP)(h_T(N(t_1, \ldots, t_n)) = h(N)(h_T(t_1), \ldots, h_T(t_n));$

A $h:\Sigma \rightarrow \Sigma'$ szignaturamorfizmus kiterjesztese egyenletek formajaban
adott axiomakra.
Legyen $e=(X, L, R) \in E$, akkor e helyettesitendo
$h^{\wedge *}(e) = (X^{\wedge *}, h^{\wedge *}(L), h^{\wedge *}(R))$ egyenlettel $\in E'$, ahol

$\forall x \in X_{s \in S}$ valtozo helyettesitendo $x^{\wedge *} \in X^{\wedge *}_{h(s)=s' \in S\backslash}$
valtozoval, L es R kepzesnel pedig
$\forall N:s_1 \ldots s_n \rightarrow s \in OP$, eseten $N(t_1, \ldots, t_n) \in T_{OP}(X)$,
helyettesitendo $h(N)(h^{\wedge *}(t_1), \ldots, h^{\wedge *}(t_n))$ operacioval.

Roviden:
- minden x valtozo helyere $h(s) = s'$-nek megfelelo valtozo;
- L, R term a $h(N) = N'$-nek helyere megfelelo operaciokkal
  kepezett $L'=R'$ lep.

Specifikaciomorfizmus

Adva: $SPEC=(\Sigma, S, OP, E)$, $SPEC'=(\Sigma', S', OP', E')$,
$h_{SPEC}:SPEC \rightarrow SPEC'$;
$h_{SPEC}=(h_{\Sigma}, h_E)$; $h_{\Sigma}:\Sigma \rightarrow \Sigma'$; $h_E:E \rightarrow E'$;

$E'=h_E(E) = \{h^{\wedge *}(e) \mid (\forall e= (X, L, R) \in E)\}.$

Definíció:
Adva parameteres tipusspecifikacio:
$PSPEC=(SPEC, SPEC1)$; ahol
$SPEC=(S, OP, E)$, $SPEC1=SPEC \cup (S1, OP1, E1).$

Adva tovabba $h:SPEC \rightarrow SPEC'$ specifikacio morfizmus, ahol
$SPEC'=(S', OP', E').$

Patameteratado morfizmus diagramja:
p:tartalmazas

```
SPEC --------------→ SPEC1 = SPEC ∪ (S1, OP1, E1)
 |                |
 | h: SPEC→SPEC'      | h1
 |                |
 V                V
SPEC'--------------→ SPEC1'
    p': tartalmazas
```

A parameteratadas jelentese:
- Ha p es p' tartalmazas az osszes reszspecifikacio eseten;
- h1:
  (∀ s ∈ S ∪ S1)(h1(s) = if s∈ S1 then s else h(s) fi);

  (∀(N:s_1 ... s_n)∈ OP∪ OP', n>=0)(h1(N:s_1 ... s_n) =
  if (N:s_1 ... s_n)∈ OP then n:h1(s_1) ... h1(s_n)→h1(s)
  else h(N):h(s_1) ... h(s_n)→h(s) fi;
- SPEC1'=SPEC' ∪ (S', OP', E1'), ahol
  S1'=S1, OP1'=h1(OP1), E1'=h1^*(E1).

```
                         p
data ----------------------------→ string(data)
 |                                     |
 |                                     |
 | h                  =                | h1
 |                                     |
 V                                     V
nat  ----------------------------→ string(nat)
              p'
```

```
                 h_1
SPEC_A --------------------------→ SPEC_B
  |                                  |
  |                                  |
  |h_2            =                  | h_3
  |                                  |
  |                                  |
  V                                  V
SPEC_C --------------------------→ SPEC_D
                h_4
```
= jelentese: h_3 kor h_1 = h_2 kor h_4;

Adattipusosztaly specifikacioja:

PAR: formalis parameterek specifikacioja;
EXP=PAR∪(S1,OP1,E1): export felulet specifikacioja;
IMP=PAR∪(S2,OP2,E2): import felulet specifikacioja;
BOD=IMP+eb(EXP):megvalositas specifikacioja;

```
               e
PAR ----------------------→ EXP
 |                           |
 | i          =              | eb
 V                           V
IMP ----------------------→ BOD
              ib
```

Specifikacio: PAR, IMP;
Kituntetett sortu specifikacio:
EXP = (S_{EXP}, OP_{EXP}, E_{EXP}); pt(S_{EXP}) ∈ S_{EXP};
BOD = (S_{BOD}, OP_{BOD}, E_{BOD}); pt(S_{BOD}) ∈ S_{BOD};

Tartalmazas: e, i, ib; (Ha az absztrakt es a konkret parameterek azonosak!)
eb: EXP → BOD; kituntetett sortu morfizmus;
Jelolese a torzs reszben:
oprs: rep: pt(S_{BOD})→pt(S_{EXP})

Absztrakt adattipus specifikacioja:
```
PAR ---------------→ EXP
        e
```

Absztrakt adattipus az adattipusoknak egy olyan osztalya, amely zart az adattartomanyok, a muveletek, a tartomanyok peldanyainak es a muveleteknek az elnevezese alapjan. Igy az absztrakt adattipus fuggetlen az adatok abrazolasatol es az adott abrazolasok mellett a muvelek megvalositasatol.

```
<osztalynev> is a class specification parameters=
```

```
parameters=         exports=
    sorts:              class sort:<osztalynev>
    oprs:               oprs:
    eqns:               eqns:

imports=        body=
    sorts:              sorts:
    oprs:               oprs: rep:pt(S_{BOD})→pt(S_{EXP});
    eqns:               eqns:
                end <osztalynev>;
```

□

Osztalyspecifikacio, specialis esetek:

```
    SPEC -------→ SPECEXP -------→ SPEC0
      |                              |
      |                              |
      V                              V
    PSPEC  -----→ PSPECEXP ------→ CLASS
```

SPEC=(0, BOD, 0, BOD);
SPECEXP=(0, EXP, 0, BOD);
SPEC0=(0, EXP, IMP, BOD);
PSPEC=(PAR, BOD, 0, BOD);
PSPECEXP=(PAR, EXP, 0, BOD);
CLASS=(PAR, EXP, IMP, BOD);

□

Modul
interfesz szerkezet szolgaltatas

Modulokbol allo rendszer
modulo interakcio moduloperaciok (valtoztatasok)

Absztrakt modul:

1. import interfesz
2. export interfesz
3. parameter resz
4. torzs resza
5. reszek kozotti relaciok

EA4

koncepcio:
bacsi

```
          e
    par     exp     (specifikacio elemzese)
     |       |
     i    = eb
     |       |
     V       V
     imp    bod
        ib
```

Tulajdonsagok felirasa + tulajdonsagok bizonyitasa.

Definíció: termek szarmaztatasa.
Adva Σ = (S, OP) szignatura es a hozzatartozo E szemantikai egyenletek halmaza, rogzitett $X = X_e$ mellett, minden e =(L,R) ∈ E eseten. Az egyenlet ket helyettesitesi szabalyt definial:
L → R;
R → L;
Ha a $t_1 \rightarrow t_2$ szabaly alkalmazhato egy $t \in T_{OP}(X)$ termre, es t_1 a t-nek egy resztermje, akkor t_1 helyettesitese t_2-vel a t termben egy ujabb t' termet eredmenyez.
Jeloles: $t'=t(t_1/t_2)$.

Ekkor azt mondjuk, hogy t' term kozvetlen szarmaztatasa t termnek E axiomai altal a $t_1 \rightarrow t_2$ szabaly felhasznalasaval.

□

A kozvetlen szarmaztatasok egy $t_0 \rightarrow t_1 \rightarrow ... \rightarrow t_n$ sorozata eseten $t=t_0$ es $t'=t_n$ jeloles mellett az e'=(t,t') egyenlet E-bol szarmaztatott egyenletnek nevezzuk az adott Σ szignaturahoz tartozo rogzitett X mellett. A szarmaztatott egyenlet helyes, ha t kiertekelese megegyezik t' kiertekelesevel.

□

```
bool is a type specification =
    sorts: bool
    oprs:   T:→bool
        F:→bool
        ¬:bool → bool
        ^:bool bool → bool [infix]
    eqns:   b ∈ bool;
        ¬T=F
        ¬(¬b)=b
        b^T=b
        b^F=F
end bool;
```

Tétel: $\neg F=T$;

Bizonyitas: $b \to T \quad \neg T \to F$
$(\neg(\neg b)=b) \to (\neg(\neg T)=T) \to (\neg F=T)$

□

or: bool bool → bool;
b1, b2 ∈ bool;
b1 or b2 = ¬(¬b1 ^ ¬b2);

Tétel: b1 or T = T;
Bizonyitas: b2→T ¬T→F
(b1 or b2) = ¬(¬b1 ^ ¬b2)) → (b1 or T = ¬(¬b1 ^ ¬T))
→ ¬b1 ^ F → F (¬F → T) → (b1 or T = ¬(¬b1 ^ F)) → (b1 or T = ¬(F))
→ (b1 or T = T).

□

PAR → EXP

$\Sigma = (S, OP)$; Σ-algebra = (S_A, OP_A);
$SPEC_A = (S_A, OP_A, E_A)$;
$d_a=(A, F, E_A)$;
$d_a=(\{A_0, A_1, ..., A_n\}, \{f_0 \to A_0, ..., f_m:A_i ... A_i \to A_k\}, \{..., \alpha(a) \Rightarrow f_s(f_c(a))=h(a), ...\}$;
$a \in A:(a_i, ..., a_k) \in (A_i \times ... \times A_i)$;

Jelolesek: $F = F_c \cup F_s$; $f_c \in F_c$; $f_s \in F_s$;

□

Egyenloseg axioma:
$a_1 = a_2 \equiv ([a_1 = f_0 \wedge a_2 = f_0] \veebar [(\forall f_s \in F_s)(f_s(a_1)=f_s(a_2))])$;

A helyettesitesi szabaly:
$a_1 = a_2 \to ([a_1 = f_0 \wedge a_2 = f_0] \veebar [(\forall f_s \in F_s)(f_s(a_1)=f_s(a_2))])$;

Peldaul:
$n_1 = n_2 \to ([n_1 = zerus \wedge n_2 = zerus] \vee [\ prec(n_1) = prec(n_2)])$;

□

Pelda.
Axioma: prec(succ(n))=n;
Tétel: n≠zerus => succ(prec(n))=n;

Bizonyitas:
n≠zerus => succ(prec(n))=n → prec(succ(prec(n))) = prec(n) → prec(n)=prec(n) → T.

□

Strukturalis indukcio:
Adott $\Sigma = (S, OP)$; a szgnaturahoz tartozo X valtozokkal.
Legyen p predikatum, amely $t \in T_{OP}(X)$ termekre van ertelmezve.
Ha
1. $(\forall\ t \in K$ es $\forall\ t \in X)(p(t) \equiv T)$;
2. $(\forall\ N(t_1, ..., t_n) \in T_{OP}(X))(p(N(t_1, ..., t_n) \equiv T)$;

akkor
$(\forall\ t \in T_{OP}(X))(p(t) \equiv T)$.

□

Strukturalis indukcio atfogalmazasa:
Adott $\Sigma = (S, OP)$; a szignaturahoz tartozo X valtozokkal.
Legyen p(t): $H_1(t) = H_2(t)$ predikatum, amely a $t \in T_{OP}(X)$ termekre van ertelmezve.
Ha
1. Alapeset:
$(\forall\ t \in K$ es $\forall\ t \in X)(H_1(t)=H_2(t) \equiv T)$;
2. Indukcios lepes:
$(\forall\ N(t_1, ..., t_n) \in T_{OP}(X))(H_1(N(t_1, ..., t_n)) = H_2(N(t_1, ..., t_n)) \equiv T)$;

akkor
$(\forall\ t \in T_{OP}(X))(H_1(t)=H_2(t) \equiv T)$.

□

Adott $\Sigma=(S, OP)$; $t \in T_{OP}(X)$;
Legyen $t_1 = H_1(f_s(t))$; $t_2=H_2(t)$;
Tekintsuk a $f_s(f_c(t))=H_{sc}(t)$; axiomat.

Tétel: $H_1(f_s(t)) = H_2(t)$;
Bizonyitas:
Alapeset:
Bizonyitsuk be f_0 konstans szimbolumra, hogy $t=f_0$ eseten:
$H_1(f_s(f_0)) = H_2(f_0)$;
Strukturalis indukcios lepes:
Mutassuk ki, hogy minden $t = f_c(t') \in T_{OP}(X)$ konstrukcios

muveletre, hogy ha $H_1(f_s(t')) = H_2(t')=T$, akkor

$H_1(f_s(f_c(t'))) = H_2(f_c(t')) \equiv T$;

azaz

$H_1(H_{sc}(t)) = H_2(f_c(t')) \equiv T$;

A strukturalis indukcio ket helyettesitesi szabalya:
1. alapeset:
   $H_1(f_s(t)) = H_2(t) \rightarrow H_1(f_s(f_0)) = H_2(f_0)$;
2. indukcios lepes:
   $H_1(f_s(f_c(t'))) = H_2(f_c(t')) \rightarrow H_1(H_{sc}(t)) = H_2(f_c(t'))$;

□

```
                               e
kettos                  par --------------------→  exp        absztrakt specifikacio
specifikacio             I                        I
                   iI            =          I eb
                    I                       I
                    V                       V
                   imp --------------------→   bod       konkret specifikacio
                                ib
```

□

Reprezentacios fuggveny.

Adva egy adattipus absztrakt es konkret specifikacioja:

$d_a=(A,F,E_a)$; $d_c=(C, G, E_c)$;

$A=\{A_0, ..., A_n\}$; $C=\{C_0, ..., C_m\}$;

$F=\{f_0:\rightarrow A_0, ..., f_i:A_i...A_k\rightarrow A_l, ...\}$;

$G=\{g_0 \rightarrow C_0, ..., g_i:C_i...C_k\rightarrow C_l, ...\}$;

Az absztrakt es konkret objektumok egymashoz valo viszonya:

$\varphi: C\rightarrow A$

$\varphi=(\varphi_0, ..., \varphi_n)$, ahol $\varphi_0:C_0\rightarrow A_0; ...; \varphi_n:C_n \rightarrow A_n$;

□

Definíció: Adva d_a absztrakt es d_c konkret tipusspecifikaciok, amelyeknek szignaturajuk azonos. Adva tovabba $\varphi: C\rightarrow A$, morfizmus. A C objektumhalmazt az A egy **reprezentaciojanak** nevezzuk, ha

$(\forall a \in A)((\exists c \in C)(a=\varphi(c))$;

□

Tétel. Adva d_a absztrakt es d_c konkret tipusspecifikaciok azonos szignaturaval.

$\varphi: C\rightarrow A$ morfizmus.
$F_c \subset F$ a konstrukcios muveletek halmaza.
Felteves: $\forall f_c \in F_c$ konstrukcios muveletre fennall:

$a\in A \wedge f_c(a)\in A \wedge c\in C \wedge g_c(c)\in C \wedge a=\varphi(c)$.

Ha

$(\forall c \in C \wedge \forall f_c\in F_c)(f_c(\varphi(c))=\varphi(g_c(c)))$,

akkor C objektumhalmaz az A egy reprezentacioja.

Bizonyitas. Strukturalis indukcioval:
a.) alapeset: $a=f_0$. $f_0\in A_0$, $g_0\in C_0$, feltevesunk szerint $f_0=\varphi(g_0)$.
Tehat $a=f_0$ eseten letezik olyan $c\in C_0$, hogy $a=\varphi(c)$.
b.) indukcio: $a'=f_c(a)$, ahol feltesszuk, hogy $a=\varphi(c)$ es $c\in C_0$.

Tehat $a'=f_c(\varphi(c))$ es muvelettartasra vonatkozo feltevesunk alapjan:

$a'=\varphi(g_c(c))$, es $c'=g_c(c)$ valasztas mellett $a'=\varphi(c')$ es $c'\in C_0$.

□

A reprezentacios fuggveny implicit definicioja:

$f_0=\varphi(g_0)$;

$(\forall f_c\in F_c)(f_c(\varphi(c))=\varphi(g_c(C))$;

□

A reprezentacios fuggveny rekurziv (explicit) definicioja:
Tegyuk fel, hogy

$c=g_c(g_s(c))$.

Ennek alapjan a reprezentacios fuggveny rekurziv definicioja:

$\varphi(c)=$<u>if</u> $c = g_0$ <u>then</u> f_0 <u>else</u> $f_c(\varphi(g_s(c)))$.

A reprezentacios fuggveny definicioja nem egyertelmu!

EA5

Kedves kollegak, miert sarga???? Mivel tudom megelozni, hogy sarga legyen???

Miert nem mondjak azt, hogy kabelhiba? (miutan 1000 db-lel orditottam, hogy 'kabel, kabel, tanar ur, kabel'!

Interfesz specifikacio megvalositasa

```
PAR ----e   --→ EXP
|               |
i       =       eb
|               |
V               V
IMP ----ib  --→ BOD
```

Interfesz: (PAR, EXP, IMP, e, i);

(itten van ez az interfesz resz ;))

Interfesz lekepezesek:
  Megvalostias: interfesz → BOD_M
  Kiterjesztes: interfesz → BOD_M
  Finomitas: interfesz → interfesz'

(az M botu azert van ott, mert az egy modul lesz)

A megvalositas egy egyszeru formaja a parameteratadas.

```
formpar -----   p   ----→ SPEC(formpar)
 |               |
 b      =        h1
 |               |
 V               V
 aktpar ----p'  ----→ SPEC(aktpar)
```

Egzakt megvalositas:
Adott a modulspecifkacio:
  MOD=(PAR, EXP, IMP, BOD, e, eb, i, ib);
ahol a MOD modul interfesz specifikacioja:
  I(MOD) = (PAR, EXP, IMP, e, i).
Az INT interfesz specifikaciot a MOD modulspecifikacio egzakt megvalositasanak nevezzuk, ha I(MOD) = INT.

Megvalositas
Adott egy INT=(PAR, EXP, IMP, e, i); interfesz specifikacio.

A MOD'=(PAR', EXP', IMP', BOD', e', eb', i', ib');
modulspecifikaciot az INT interfesz specifikacio megvalositasanak nevezzuk, ha letezik olyan r= (r_P, r_E, r_I); specifikacio morfizmus harmas, amelyikre
  i' **o** r_P = r_I kor i; es e' **o** r_P = r_P kor e;

A megvalositas az alabbi diagram kommutaciojat fejezi ki:

```
PAR --- e   --→ EXP
 |  \               \
 i   r_P            r_E
 |     \               \
 V    PAR' ---  e' ---→  EXP'
IMP    |       |
 \     i'         e_b'
 r_I   |       |
   \   V       V
     IMP'  ---   ib' --→ BOD'
```

Ha r_P = r_E = r_I = identitas, akkor egzakt megvalositas.

Legyen SPEC'=(S', OP', E') a SPEC=(S, OP, E) specifikaciobol morfizmussal szarmaztatott specifikacio:
- Tartalmazas.
- Atnevezes. a Specifikaciot atnevezzuk ugy, hogy a sortok, az operaciok uj nevet kapnak, de ugy, hogy a szemantika valtozatlan marad
- Abrazolas, amely az atnevezes egy formaja

A sort atnevezesenek jelolese.

sorts: <uj sort neve> = <regi sort neve>

Az operacios szimbolum atnevezesenek jelolese.

oprs: <uj op neve> = <regi op neve>

Peldaul:
oprs: _ + 1 = succ
_ + _ = add
+ [infix] = add
put = _ [_] ::= _

□

Deklaracios resz atnevezese:
(a sortok atnevezesei alapjan automatikus)

eqns: a1, a2, ..., ak ∈ <regi sort neve>;

Atnevezes:
eqns: c1, c2, ..., ck ∈ <uj sort neve>;

□

Szemantikai egyenletek atnevezese:
(A sortok es operacios szimbolumok atnevezesei alapjan automatikus.)
regi axioma: e: L = R;
regi op neve: f_0, ..., f_n;
L: f_s(f_c(a));
R: f_i(...(f_j(a))...);

uj op nevek rendre: g_0, ..., g_n; akkor
uj axioma: e'=L'=R'; ahol
L': g_s(g_c(a));
R':g_i(...(g_j(a))...);

□

Tétel. Adva d_A absztrakt es d_c konkret tipusspecifikaciok azonos szignaturaval.

φ: C→A morfizmus.
F_C ⊂ F a konstrukcios muveletek halmaza.
Felteves: ∀ f_c ∈ F_C konstrukcios muveletre fennall:
a∈ A ∧ f_c(a)∈ A ∧ c∈ C ∧ g_c(c)∈ C ∧ a=φ(c).
Ha
(∀ c∈ C ∧ ∀ f_c ∈ F_C)(f_c(φ(c))=φ(g_c(c))),
akkor C objektumhalmaz az A egy reprezentaciojara.

Bizonyitas. Strukturalis indukcioval:
a.) Alapeset: a=f_0, f_0 ∈ A_0, g_0 ∈ C_0, feltevesunk szerint f_0 = φ(g_0),
Tehat a=f_0 eseten letezik olyan c∈ C_0, hogy a=φ(c).
b.) Indukcio: a'=f_c(a), ahol feltesszuk, hogy a=φ(c) es c∈ C_0.
Tehat a'=f_c(φ(c)) es muvelettartasra vonatkozo feltevesunk alapjan:
a'=φ(g_c(c)), es c'=g_c(c) valasztas mellett a'=φ(c') es c'∈ C_0.

□

A reprezentacios fuggveny implicit definicioja:
f_0 = φ(g_c);
(∀ f_c∈ F_c)(f_c(φ(c)) = φ(g_c(c));

□

A reprezentacios fuggveny rekurziv (explicit) definicioja:
Tegyuk fel, hogy c = g_c(g_s(c)).
Ennek alapjan a reprezentacios fuggveny rekurziv definicioja:
φ(c) = if c=g_0 then f_0 else f_c(φ(g_s(c))).
A reprezentacios fuggveny definicioja nem egyeretelmu!

□

Pelda: g_0: 0; g_s(n): n-1; g_c(n): n+1;

c=g_c(g_s(c)): n = (n-1)+1;
f_0: zerus, f_s(n):prec(n); f_c(n):succ(n);
φ(n) = if n=0 then zerus esle succ(φ(n-1)) fi

□

Reprezentacio jelolese:
body =
oprs: rep: vector nat data → stack
eqns: v∈ vector, n∈ nat, d∈ data;
create = (nil, zerus)
push(v, n, d) = (put(v, n+1, d), n+1)

body =
oprs: rep: C_0 C_1 ... C_i → A_0
eqns: c_0 ∈ C_0; c_1 ∈ C_1; ...; c_i ∈ C_i
f_0=g_0(c)
f_c(c_0, c_1, ..., c_i) = g_c(c)

□

A BOD spedcifikacio reprezentacios formaja:
(az ib, eb morfizmusok alapjan automatikus kiegeszitessel egyutt)
body = imports +
(kek) class sort: C_0
oprs: rep: $C\_0\ C\_1 \ldots C\_i \rightarrow A\_0$
(kek) $(\forall f\_i: A\hat{}+ \rightarrow A \in F)(g: C\hat{}+ \rightarrow C)$
eqns: $c\_0 \in C\_0$; $c\_1 \in C\_1 \ldots c\_2 \in C\_2$
$f\_0 = g\_0(c)$
$f\_c(c\_0, c\_1, \ldots, c\_i) = g\_c(c)$
(kek) $\forall(f\_s(f\_c(a)) = f\_i(\ldots(f\_j(a))\ldots) \in \text{exports} \wedge a=\text{rep}(c))$
(kek) $g\_s(g\_c(c)) = g\_i(\ldots(g\_j(c))\ldots)$;

Ami kekkel van irva, azt nem kell leirni, amikor ilyet csinalunk, mert azokat odateszi a morfizmus.

□

Reprezentacio elemzes:
- $\varphi\_1(c) = \varphi\_2(c)$?
- $attr\_c(c) = attr\_s(\varphi(c))$?
- $c\_1 = c\_2 \Rightarrow \varphi(c\_1) = \varphi(c\_2)$?
- $I\_c(c) \Rightarrow I\_a(\varphi(c))$?

(Ha $attr\_c(c) = attr\_a(\varphi(c))$ es $I\_c(c)$: $0 <= attr\_c(c) <= n$ es $I\_a(\varphi(c))$:$0<=attr\_a(\varphi(c))<=n$, akkor az $I\_c(c) \Rightarrow I\_a(\varphi(c))$ allitas trivialis).

(azt mondom, hogy en egy okos ember vagyok)

□

$\varphi\_1(c) = \varphi\_2() \equiv (\varphi\_1(c) = f\_0 \wedge \varphi\_2(c) = f\_0) \vee$
$(\forall f\_s \in F\_s)(f\_s(\varphi\_1(c)) = f\_s(\varphi\_2(c)))$;

Bizonyitas:
a.) alapeset. $c = g\_0$; $(\varphi\_1(g\_0) = f\_0 \wedge \varphi\_2(g\_0)=f\_0)$
b.) indukcios lepes. $c=g\_\{c1\}(c')$, ahol c're $\varphi\_1(c')=\varphi\_2(c')$.
$\varphi\_1(g\_\{c1\}(c')) = \varphi\_2(g\_\{c1\}(c')) = (\forall f\_s \in F\_s)(f\_s(\varphi\_1(g\_\{c1\}(c')))=f\_s(\varphi\_2(g\_\{c1\}(c')))) =$
$(\forall f\_s \in F\_s)(f\_s(f\_\{c1\}(\varphi\_1(c')))=f\_s(f\_\{c2\}(\varphi\_2(c'))))$?

□

Allitas.
$attr\_c(c) = attr\_s(\varphi(c))$.

Bizonyitas indukcioval.
a.) alapeset: $c=g\_0 \Rightarrow \varphi(g\_0)=f\_0$; $attr\_c(g\_0) = attr\_a(\varphi(g\_0))$?
b.) indukcios lepes: Felteves $c=g\_c(c')$ mellett $attr\_c(c') = attr\_a(\varphi(c'))$.
$attr\_c(c)=attr\_c(g\_c(c'))$

$attr\_a(\varphi(c))=attr\_a(\varphi(g\_c(c')) = attr\_a(f\_c(\varphi(c'))$

$attr\_c(g\_c(c')) = attr\_a(f\_c(\varphi(c'))$?

□

$c\_1=c\_2 \Rightarrow \varphi(c\_1) = \varphi(c\_2)$?
$c\_1=c\_2 \equiv (c\_1 = g\_0 \wedge c\_2=g\_0) \vee (\forall g\_0 \in G\_s)(g\_s(c\_1) = g\_s(c\_2)) \Rightarrow$
$\varphi\_1(c)=\varphi\_2(c) \equiv (\varphi\_1(c)=f\_0 \wedge \varphi\_2(c)=f\_0) \vee (\forall f\_s \in F\_s$

(ANYAD!!!)

EA6

Interfesz: (PAR, EXP, IMP, e, i);
e=i ≡ tartalmazas;

□

Megvalositas:

```
PAR --  e   -→  EXP
 |               |
 i      =        eb
 v               v
EMP --  ib  -→  BOD
```

ib: tartalmazas;
eb: tartalmazas abrazolassal

□

Kettos specifikacio.

Adott $d\_a=(A, F, E\_a)$; $d\_c = (C, G, E\_c)$;

$a_0=\{c \mid I_a(c)\}$; $C_0 = \{c \mid I_c(c)\}$;
abrazolas: $\varphi: C_0 \to A_0$;

$E_a=\{\ldots, \alpha(a) \Rightarrow f_s(f_c(a))=h(a); \ldots),$
$(\neg I_a(f_c(a)) \wedge I_a(a)) \Rightarrow f_c(a)=\text{"undifined"}\};$
$h(a)=f_i(\ldots(f_j(a))),$
$E_c=\{\ldots, \alpha_c(c) \Rightarrow g_s(g_c(c))=h_c(c);$
$(\neg I_c(g_c(c)) \wedge I_c(c)) \Rightarrow g_c(c)=\text{"undifined"}\};$ (algebrai leiras)
$h_c(c)=(g_i(\ldots(g_j(c))), \ldots),$
$E_c=\{\ldots, \{pre_i(c)\}\ c'=g_i(c,c')\ \{post_i(c,c')\}, \ldots,$ (elo-utofelteteles leiras)
$\{I_c(g_c(c)) \wedge I_c(c)\}\ c'=g_c(c,c')\{I_c(c') \wedge c'=f_c(c)\};$
$E_c=\{\ldots, Q_i(c,c'),\ldots\};$ (algoritmusos leiraas)

Hogyan donthetjuk el, hogy ez az utobbi 3 helyes-e az absztrakthoz kepest?

$\square$

Minden algebrai axioma elo- utofelteteles formara hozhato.

$\alpha(a) \Rightarrow f_s(f_c(a))=h(a);$
$\{\alpha(a) \wedge b=f_c(a)\}\ b'=f_s(b)\ \{b'=h(a)\}$

$(\neg I_a(f_c(a)) \wedge I_a(a)) \Rightarrow f_c(a) = \text{"undefined"};$
$\{I_a(a) \wedge I_a(f_c(a))\}\ a'=f_c(a)\ \{I_a(a') \wedge a'=f_c(a)\}$

$\square$

Definíció: (Az implementacio helyessege)

Adva
$d_a=(A, F, E_a)$ absztrakt specifikacio
$d_c=(C, G, E_c)$ konkret specifikacio, amelynek a szignaturaja azonos.
$\varphi: C \to A$ morfizmus.
HA
1) C az a egy reprezentacio az adott φ mellett;
2) $(\forall f_i \in F)(c \in C \wedge \varphi(c) \in A \wedge f_i(\varphi(c))$ ertelmezve van $\Rightarrow g_i(c)$ is ertelmezve van;
3) $(\forall f_i \in F)(c \in C \wedge c'=g_i(c) \wedge c' \in C \wedge \varphi(c) \in A \wedge \varphi(c') \in A$
$\Rightarrow f_i(\varphi(c)) = \varphi(g_i(c));$
akkor d_c a d_a szerint helyes.

$\square$

Szimulacio:

```
a<---------φ(c)-------------c
 \              / g_i(c)
  f_i(a)       /
   v           v
   a' <-----φ(c')---------c'
```

es be kell karikazni a bal es a jobb oldalt kulon, mivel ez eg absdztrakt meg egy konkret ter.

$\square$

Allitas.
Adva $d_a=(A, F, E_a)$, $d_c=(C, G, E_c)$, $\varphi: C \to A$;
d_c a d_a szerint helyes.
P_a(absztrakt program, $p_a(a)$: P_a programfuggvenye.
Allitsuk elo a P_c konkret programot a P_a absztrakt programbol ugy, hogy
$\forall a \in A$ helyere a megfelelo $c \in C$-t
$\forall f_i \in F$ helyere a megfelelo $g_i \in G$-t tesszuk.
Ha a konkret program programfuggvenye a $p_c(c)$ es a programok indulasakor
$a_0=\varphi(c_0)$
akkor
$p_a(\varphi(c_0)) = \varphi(p_c(c_0)).$

Bizonyitas.
Az adattipus programban szereplo muveleteinek szama szerinti teljes indukcioval.

a.) Alapeset. Felteves: indulaskor $a_0=\varphi(c_0)$.
b.) Indukcios lepes:
k a muveletek szama, $k>0$. A k-ik muvelet eredmenye:
$a_k=f(a_{k-1}), c_k=g(c_{k-1});$
Indukcios felteves:
$a_{k-1}=\varphi(c_{k-1}).$
A k-ik muvelet eredmenye:
$(f(a_{k-1}),g(c_{k-1})),$
$a_k=f(a_{k-1})=f(\varphi(c_{k-1}))=\varphi(g(c_{k-1}))=\varphi(c_k).$
Az utolso lepes eredmenye:
$(a', c'),$
$a'=\varphi(c'),$
$a'=p_a(a_0)$ es $c'=p_c(c_0),$

$a'=p_a(a_0) \wedge c'=p_c(c_0)$, ezert $a'=\varphi(c')$, azaz $p_a(a_0)=\varphi(p_c(c_0)),$
$p_a(\varphi(c_0))=\varphi(p_c(c_0))$

Allitas. (A kulso felilet specifikaciojaval adott konkret specifikacio absztrakt specifikacio szerinti helyessegenek egy elegseges feltetele.)

Adva
d_a=(A, F, E_a), d_c=(C, G, E_c),

E_a={{"true"} a=f_0 {post_f_0(a)}, ..., {pre_f_i(a)} a'=f_i(a) {post_f_i(a, a')}, ... },
E_c={{"true"} c=g_0 {post_g_0(c)}, ..., {pre_g_i(c)} c'=g_i(c) {post_g_i(c, c')}, ... }.

A_0 = {a| I_a(a)}, C_0 = {c| I_c(c)}, φ:C→A.

Ha bebizonyitjuk, hogy
1) I_c(c)⇒ I_a(φ(c);
2) post_g_0(c) ⇒ I_c(c);
3) post_g_0(c) ⇒ post_f_0(φ(c));
4) I_c(c) ∧ pre_f_i(φ(c)) ⇒ pre_g_i(c);
5) I_c(c) ∧ pre_f_i(φ(c)) ∧ post_g_i(c,c') ∧ I_c(c') > post_f_i(φ(c), φ(c'));

akkor a d_c konkret specifikacio a d_a absztrakt specifikacio szerint helyes.

Bizonyitas.
a) C az A egy reprezentacioja.
- a = f_0. 2. es 1. szerint: g_0∈ C ∧ φ(g_0) ∈ A. 3. szerint: φ(g_0) = f_0.
- 4. szerint:
ha f_c∈ F_c, f_c(φ(c)) ertelmezve van, akkor g_c(c) is ertelmezve van.
- 5. szerint:
φ(c')=f_c(φ(c)) ∧ c'=g_c(c),
f_c(φ(c)) = φ(g_c(c))

b)
- 3. szerint:
ha ∀ f_i∈ F, f_i(φ(c)) ertelmezve van, akkor g_i(c) is ertelmezve van.
- 5. es 1. szerint
c'=g_i(c) ∧ I_c(c') ∧ I_a(φ(c')) ∧ φ(c')=f_i(φ(c)),
(∀ f∈ F)(φ(g(c)) = f(φ(g(c))).

Megjegyzes: Ha p=f_i(a) es q=g_i(c) ahol p, q parameterek, akkor f_i(φ(c)) = φ(g_i(c)) helyebe p=q lep.

Reprezentacio elemzes:
φ_1(c) = φ_2(c) ?
length_c()=length_a(φ(c)) ?
c_1=c_2 ⇒ φ(c_1) = φ(c_2) ?
I_c(c) ⇒ I_a(φ(c)) ?

Implementacio elemzes:
post_g_0(c) ⇒ post_f_0(φ(c)) ?
I_c(c) ∧ pre_f_i(φ(c)) ⇒ pre_g_i(c) ?
I_c(c) ∧ pre_f_i(φ(c)) ∧ post_g_i(c,c') ∧ I_c(c') ⇒ post_f_i(φ(c), φ(c')) ?

Formalis parameterek aktualizalasaval torteno abrazolas.

Definíció:
Adott egy
INT = (PAR, EXP, IMP, e, i);
interfesz specifikacio.
Adott annak
MOD'=(PAR', EXP', IMP', BOD', e', eb', i', ib'); megvalositasa.
Ekkor a r=(r_P, r_E, r_I) specifikacio morfizmus harmasra:
i' **o** r_P = r_I **o** i; es e' **o** r_P = r_E **o** e;

```
PAR ------- e------→ EXP
 |  \                   \ r_E
 i   v r_P               v
 v    PAR' ----- e' -→ EXP'
 IMP   |                 |
  \    |                 |
   r_I i'               eb'
    v  v                 v
    IMP'----- ib' ---→ BOD'
```

Megjegyzes: Ha
e, e': tartalmazas;
PAR: a formalis parameterek specifikacioja;
r_P : PAR → PAR', ahol PAR' az aktualis parameterek specifikacioja.
r_E: EXP → EXP' a parameter atadasnak megfelelo tartalmazas.
i', ib': tartalmazas; eb': abrazolas+ tartalmazas;

i: PAR → IMP; ahol IMP = PAR ∪ IMP1;

r_I: IMP→IMP', ahol IMP' = PAR' ∪ IMP1';

akkor a MOD' az INT interfeszt aktualis parameterekkel es abrazolassal torteno megvalositasnak nevezzuk.

□

terminator ablak, es ott ultunk szilvivel a 3. sorban es nem birta ki, hogy ne rohogje el magat,
amikor o nevetett, persze mar nekem is kellett es akkor rajottek a tobbiek is, hogy mit mondott.
Aztan nacsa kerdezte, hogy mi baj van es hogy mondjuk el neki is...

□

```
PAR_a ----- e ----→  SPE_a(PAR_a)
 |   \ r_P         \ r_E
 i    \>PAR_C ----- e' ------→  EXP_a(PAR_c)
 |         |           |
 V         |           |
 IMP       i'          eb'
  \        |           |
   \ r_I   V           V
    ----→ IMP' ----ib' ---→  EXP_c(PAR_c)
```

□

```
  nat_a + data --------- e -----→  vector_a(nat_a, data)
      |       \                |
      i        \ r_P        \ r_E
      |     \            |
      |     V             V
      V    nat_c+data ------- e' -----→  vector_a(nat_c, data)
      IMP      |              |
       \       | i'          eb'
        \ r_I  |              |
     \    V             V
      \> IMP' ------ ib' ----→  vector_c(nat_c,data)
```

□

EA7

Interfesz realizaciok.

INT = (PAR, EXP, IMP, e, i); interfesz specifikacio.

r: INT → MOD;

1.) incialis realizacio;
2.) Vegleges realizacio.
Jeloles: Inicialis realizacio: IR(INT);
Vegleges realizacio: FR(INT);

Legyen
I(IR(INT)) = INT, I(FR(INT)) = INT,
akkor mindket modulspecifikacio egzakt realizacio.

□

Inicialis realizacio:
IR(INT) = (PAR, EXP, IMP, BOD, e, i, eb_1, ib_1);
Vegleges realizacio:
FR(INT) = (PAR, EXP, IMP, FINAL, e, i, eb_2, ib_2);

□

Szarmaztatas (Derivation): $t_0 \to t_1 \to ... \to t_n$, $e=(t_0,t_n)$, $t_1 \in T_{OP, X}$;

$e_i \in E: e_i=e(X, L_i, R_i)$; $e_0 \to e_1 \to ... \to e_n$; $e_i \equiv T$; $i=1,...,n$;

□

Jeloles:
d_a=(A, F, E_a): a PAR ∪ EXP export felulet specifikacioja.
d_c=(C, G, E_c): a BOD torzsresz specifikacioja, realizacio.
Adva φ(c) reprenzentacios fuggveny.

Definíció: Legyen $E_a=\{L_{ai}(a)=R_{ai}(a) | 1<=i<=k\}$, $E_c=\{L_{ci}(c)=R_{ci}(c) | 1<=i<=k\}$;
Ha $(\forall\ i, 1<=i<=n)(e_{ai}(a) \to ... \to e_{ai}(a); e_{ci}(c) \to ... \to e_{ck}(c))$;
es $(\forall\ i, 1<=i<=n)(e_{ai}(\varphi(c))=e_{ck}(c) \equiv T)$,
akkor d_c specifikaciot a d_a specifikacio szerinti korrekt realizacionak
nevezzuk az adott φ(c) reprezentacio mellett.

□

Tétel.
d_a=(A, F, E_a): az export felulet specifikacioja.
d_c=(C, G, E_c): a torzsresz specifikacioja.
Adva φ(c) reprezentacios fuggveny.
Ha
- $(\varphi(t_{c1} = \varphi(t_{c2})) \equiv (t_{c1} = t_{c2})$,

- barmely (t_a, t_c) par, ahol t_a ∈ T_{Σ a}, t_c∈ T_{Σ c}, t_a=φ(t_c);

akkor d_c a d_a szserinti korrekt realizacio a φ(c) reprezentacio mellett.

Bizonyitas.
L_{ai}(a) = R_{ai}(a);
l_{ai}(a) → L_{ai}(φ(c)) → φ(L_{ci}(c));
R_{ai}(a) → R_{ai}(φ(c)) → φ(R_{ci}(c));

Tehat
φ(L_{ci}(c)) = φ(R_{ci}(c)) ≡ L_{ci}(c) = R_{ci}(c);

□

Adott L_{ai}(φ(c)) = R_{ai}(φ(c));
Peldaul: f_s(f_c(φ(c))) = f_c(f_s(φ(c)))

Vegleges realizacio.
- {pre_g(c)}c'=g(c){post_g(c,c')};
- Q_g(c);

□

Adva az interfesz specifikacio:
INT_{stack} = (PAR_{stack}, EXP_{stack}, IMP_{stack}, e, i);

PAR_{stack} = (data, {undef: → data});

EXP_{stack} = ({stack, data}, {create: → stack, push: stack data → stack, pop: stack → stack, top: stack → data},
{s∈ stack, d∈ data; pop(create) = create, pop(push(s,d)) = s, top(create)=undef, top(push(s,d)) = d});

IMP_{stack} = (({vector, nat, data}, {nil: → vector, put: vector nat data → vector, access: vector nat → data},
{v∈ vector, j,k∈ nat, d∈ data; access(nil, j) = undef, access(put(v,j,d),k)= if j=k then d else access(v,k) fi},
i: PAR_{stack} → IMP_{stack};
e: PAR_{stack} → EXP_{stack};

□

Iniicialis realizacio:
r: INT → MOD'; r = (r_P, r_E, r_I);
r_P: identitas, azaz PAR_{IR(stack)}=PAR_{stack},
r_E: identitas, azaz EXP_{IR(stack)}=EXP_{stack},
r_I: identitas; azaz IMP_{IR(stack)}=IMP_{stack}.

□

φ: vector nat → stack;

φ(v_1, m_1) = φ(v_2, m_2) ≡ (φ(v_1, m_1)=create ∧ φ(v_2, m_2)=create) ∨
(pop(φ(v_1, m_1)) = pop(φ(v_2,m_2)) ∧ (ide sor elejetol az egesz ujra, csak top-pal)

□

v∈ vector, m∈ nat, d∈ data;

pop(create) = create;
pop(create) → pop(φ(v, 0)) → φ(v,0);
create → φ(v,0);
Tehat:
φ(pop_c(create_c)) = φ(create_c) ≡ pop_c(create_c) = create_c;

pop(push(s,d)) = s;
pop(push(s,d)) → pop(push(φ(v,m),d)) → pop(φ(put(v,succ(m),d),succ(m)) → φ(put(v,succ(m), d), prec(succ(m)));
s→ φ(v,m);
Tehat: φ(pop_c(push_c(v,m,d))) = φ(v,m) ≡ pop_c(push_c(v,m,d)) = (v,m);

□

Proceduralisan adott konkret specifikacio elo- es utofeltetelekkel adott absztrakt specifikacio szerinit helyessege.

Tétel.

Adottak a d_a es a d_c specifikaciok kozos szignaturaval:
d_a=(A,F,E_a)I
A={A_0, A_1, ..., A_n}; F={f_0:→A_0, f_1:A^+ → A, ..., f_m : A^+ → A};
{"true"} a = f_0 {post_f_0(a)} ∈ E_a,
{pre_f_i(a)}a' = f_i(a){post_f_i(a,a')}∈ E_a, f_i ∈ F;

d_c=(C,G,E_c);
C={C_0, C_1, ..., C_n}; G={g_0:→C_0, g_1: C^+→C, ..., g_m: C^+ → C};
(∀ i, i∈ {0,1,...,m}) (O_g_i∈ E_c, g_i∈ G);
ahol Q_g_i az f_i kiszamitasara szolgalo eljaras, program, azaz:
procedure g_0 egin Q_0 end;
precedure g_i begin Q_i end; i=1,..., n;

absztrakt invarians:
A_0={alI_a(a)},
konkret invarians:

$c_0=\{c|I_c(c)\}$,
A reprezentacios fuggveny:
$\varphi: C \rightarrow A$,

Ha a kovetkezo tetelek teljesulnek:
1. $(\forall c \in C)(I_c(c) \Rightarrow I_a(\varphi(c))$;
2. $\{\text{"true"}\}\ Q_0\ \{post\_f_0(\varphi(c) \wedge I_c(c)\}$;
3. $(\forall f_i \in F)$: $\{pre\_f_i(\varphi(c)) \wedge I_c(c)\}\ Q_i\ \{post\_f_i(\varphi(c), \varphi(c')) \wedge I_c(c')\}$;

ahol 2. es 3. teljes helyessegi tetelek, akkor a d_c konkret specifikacio
a d_a absztrakt specifikacio szerint helyes.

□

Bizonyitas.

- C az A egy reprezentacioja.

Minden $g_i(c)$ konstrukctios muvelet szimulalja $f_i(\varphi(c))$-t.

$(\{\text{"true"}\}\ Q_0\ \{post\_f_0 (\varphi(c)) \wedge I_c(c)\} \equiv \text{"true"}) \Rightarrow f_0 = \varphi(g_0) \wedge g_0 \in C)$.

$((\forall f_i\in F_c)$: $\{pre\_f_i(\varphi(c)) \wedge I_c(c)\}\ Q_i\ \{post\_f_i(\varphi(c), \varphi(c')) \wedge I_c(c')\}) \Rightarrow$
    1.) ha $f_i(\varphi(c))$ ertelmezve van, akkor $g_i(c)$ is.
    2.) $(c\in C \wedge c'=g_i(c) \wedge c'\in C \wedge a=\varphi(c)\wedge a\in A \wedge a'=f_i(a)\wedge a'\in A) \Rightarrow a'=\varphi(c')$.

- minden $g_i(c)$ nem konstrukcios muvelet is szimulalja $f_i(\varphi(c))$-t.
$((\forall f_i\in F)$: $\{pre\_f_i (\varphi(c))\wedge I_c(c)\}\ Q_i\{post\_f_i(\varphi(c),\varphi(c'))\wedge I_c(c')\}) \Rightarrow$
    3.) ha $f_i(\varphi(c))$ ertelmezve van akkor $g_i(c)$ is.
    4.) $(c\in C \wedge c'=g_i(c)\wedge c'\in C\wedge a=\varphi(c)\wedge a\in A\wedge a'=f_i(a)\wedge a'\in A) \Rightarrow a'=\varphi(c')$.

□

Determinisztikus program.

S::=skip | u<-t |S1;S2| if B then S_1 else S_2 fi| while B do S_1 od.

u<-t ertekadas; u valtozo, t kifejezes; u es t azonos tipusuak.
B kvantorfuggetlen logikai kifejezes;

EA8

{p} S{q}

□

Determinisztikus program:
S::=skip | u<-t | S_1;S_2 | if α then S_1 else S_2 fi | while α do S od.

u<- t ertekadas; u valtozo, t kifejezes; u es t azonos tipusuak.
α kvantor-fuggetlen logikai kifejezes;

□

Tipus:
    Alaptipusok: integer; bool; ...
    Osszetett tipusok:
        $T_1\ T_2 \ldots T_n \rightarrow T$; $n>=1$,
            ahol $T_1, T_2, \ldots, T_n, T$ alap tipusok.

□

Valtozo es konstans

- valtozo
    - egyszeru valtozo: (integer, bool, ...);
    - tomb valtozo: egy dimenzios, tobb dimenzios;
- konstans
    - alap tipusu: inteer, bool, ...;
    - osszetett tipusu: $T_1\ T_2 \ldots T_n \rightarrow T$;
        T bool: akkor relacio szimbolum,
        T nem bool: fuggveny szimbolum.

□

Tipusos kifejezesek

1. T alaptipusu konstans.
2. Egyszeru T tipusu valtozo.
3. Ha $s_1, \ldots, s_n$ rendre $T_1, \ldots, T_n$ tipusu kifejzesek es op: $T_1 \ldots T_n$
    $\rightarrow T$, akkor $op(s_1, \ldots, s_n)$ T tipusu kifejezes.
4. Ha $s_1, \ldots, s_n$ rendre $T_1, \ldots, T_n$ tipusu kifejezesek A egy
    $T_1 \ldots T_n \rightarrow T$ tomb, akkor $A[s_1, \ldots, s_n]$ T tipusu kifejezes.
5. Ha α Boolean kifejezes es s_1, s_2, T tipusu kifejzesek, akkor
    if α then s_1 else s_2 fi T tipusu kifejzes.

□

Az alap tipus-halazokon ertelmezett szokasos kifejezesek rekurziv definicioja:

kifejezes $e ::= c \mid x \mid (e_1 + e_2) \mid (e_1 - e_2) \mid (e_1 \times e_2)$;

bool kifejezes $b ::= (e_1 = e_2) \mid (e_1 < e_2) \mid \neg b \mid (e_1 \wedge e_2)$.

□

Szemantika:

<szintaktikai tartomany> → <szemantikai tartomany>;

□

Allapot.

Egy T tipusu konstans allapota annak konkret erteke.

Egy T tipusu v valtozo allapota $\Sigma(v)$.
A T tipusu lehetseges allapotainak halmazat jelolje: D_T.
A T tipusu v valtozo megfelelo allapota egy lekepezes D_T-re: $\Sigma(v) \in D_T$.

□

Kifejezes jelentese.

Jeloles. Adott D_T mellett a megfelelo allapotok halmazat jelolje Σ.

Definíció: Egy T tipusu s kifejezes jelenetese:

$\Sigma(s): \Sigma \to D_T$;

Definíció:

1. Ha az e kifejezes egy T tipusu d konstans: $\Sigma(e)=d$;
2. Ha az e kifejezes egy T tipusu v egyszeru valtozo: $\Sigma(e)=\Sigma(v)$;
3. Ha az e kifejezes egy T tipusu muvelet: $e=op(s_1, ..., s_n)$,
   amelyhez az $f(s_1, ..., s_n)$ lekepezes tartozik:
   $\Sigma(e)=f(\Sigma(s_1), ..., \Sigma(s_n))$;
4. Ha az e kifejezes egy T tipusu tomb: $e=A[s_1, ..., s_n]$,
   $\Sigma(e)=\Sigma(A)(\Sigma(s_1), ..., \Sigma(s_n))$;
5. Ha e if α then s_1 else s_2 fi formaju bool kifejezes:
   $\Sigma(\alpha)=$"true" → $\Sigma(e)=\Sigma(s_1)$;
   $\Sigma(\alpha)=$"false" → $\Sigma(e)=\Sigma(s_2)$.

□

Egy p allitas jelentese: $S(p): \Sigma \to \{$"true", "false"$\}$;

□

A program jelentese: M[S];

Denotacios szemantika; _Operacios szemantika_.

□

Az allapot-atmenet: $\langle S, \Sigma\rangle \to \langle S',\Sigma'\rangle$.

S program, Σ kiindulasi allapottal.
S' maradek program, Σ' eredmeny allapottal.
S'=E: programon beluli ures program.

□

Determinisztikus program jelentese.

$\langle$ skip, $\Sigma \rangle \to \langle E, \Sigma \rangle$;
$\langle u \leftarrow t, \Sigma \rangle \to \langle E, \Sigma[u \leftarrow t] \rangle$;

```
/------------------------------------------\
|  <S_1, Σ> → <S_2, Σ'>                    |
|  --------------------------------------  |
|  <S_1; S, Σ> → <S_2; S, Σ>               |
\------------------------------------------/
```

$\Sigma(\alpha) \Rightarrow \langle$ if α then S_1 else S_2 fi, $\Sigma \rangle \to \langle S_1, \Sigma\rangle$;
$\Sigma(\neg\alpha) \Rightarrow \langle$ if α then S_1 else S_2 fi, $\Sigma \rangle \to \langle S_2, \Sigma\rangle$;

$\Sigma(\alpha) \Rightarrow \langle$while α do S od, $\Sigma\rangle \to \langle S;$while α do S od$,\Sigma\rangle$;
$\Sigma(\neg\alpha> \Rightarrow \langle$while α do S od, $\Sigma\rangle \to \langle E, \Sigma\rangle$;

□

Az S program allapotainak halmaza: Σ.
Egy allapot: $\Sigma \in \Sigma$.

□

Az S_0 program vegrehajtasa:
$\tau: \langle S_0, \Sigma_0\rangle \to \langle S_1, \Sigma_1\rangle \to ... \to \langle S_{n-1}, \Sigma_{n-1}\rangle \to$
$\to \langle S_n, \Sigma_n\rangle$;

$\langle S_i, \Sigma_i\rangle \to \langle S_{i+1}, \Sigma_{i+1}\rangle$ atmenethez tartozik egy tranzakcio:
$(S_i, \alpha_i \to r_i, S_{i+1})$; ugy hogy $\alpha(\Sigma_1)=$"true" es
$\Sigma_{i+1}=r_i(\Sigma_i)$;

☐

Az S_0 program vegrehajtasa befejezodik Σ_n allapotban, ha
τ veges, es az utolso konfiguracio: $\langle E, \Sigma_n \rangle$;
$\langle S_0, \Sigma_0 \rangle \to^* \langle E, \Sigma_n \rangle$;

Jeloles: $val(\tau) = \Sigma_n$.

☐ (az undef mostanaban a feje tetejere allitott T-t jelenti es nem mast)

A τ vegrehajtas lehet vegtelen (divergens). Virtualis vegrehajtas:
$\langle S, \Sigma \rangle \to^* \langle E, undef \rangle$;
$val(\tau)=\perp$. $\perp \notin \Sigma$.

☐

comp(S)(Σ):
az S program osszes kiszamitasanak eredmenye, amely Σ kezdesi allapotahoz tartozik.

☐

Determinisztikus program eseten comp(S)(Σ) egyelemu.

☐

Az S program input output szemantikaja:
$M[S] : \Sigma \to \Sigma$.

☐

Az S program jelentese adott Σ eseten:
- $M[S](\Sigma) = \{\Sigma' \mid \Sigma' \in comp(S)(\Sigma)\}$,
- Ha az S vegrehajtasa sikertelen: $fail \in M[S](\Sigma)$.
- Ha az S vegrehajtasa divergens: $\perp \in M[S](\Sigma)$.

☐

Az S program parcialis helyessegi szemantikaja:

$M[S]: \Sigma \to P(\Sigma)$,
$M[S](\Sigma): \{\Sigma' \mid \langle S, \Sigma \rangle \to^* \langle E, \Sigma' \rangle\}$

Az S program teljes helyessegi szemantikaja:
$M_{tot}[S]:\Sigma \to (P(\Sigma \cup \{\perp\})$,
$M_{tot}[S](\Sigma) = M[S](\Sigma) \cup \{\perp\}$.

Az S program specifikacioja egy (φ, ψ) kettos, ahol φ a program elofeltetele es ψ az utofeltetele, azaz $\varphi(\Sigma)$ = "true" es $\forall\ \Sigma' \in M[s](\Sigma)$ eseten $\psi(\Sigma')$="true".

☐

Programhelyessegi kerdesek

☐

Parcialis helyesseg. Az S programot a (φ, ψ) specifikacio szerint parcialisan helyesnek mondjuk, ha minden $\Sigma \in \Sigma$ kezdo ertekhez tartozo allapot mellett, amelyre $\varphi(\Sigma)$ = "true", felteve, hogy a vegrehajtas befejezodik $\Sigma' \in \Sigma$ es $\psi(\Sigma')$="true" allapot mellett, akkor: $[(\varphi(\Sigma) = \text{"true"}) \wedge (\Sigma' \in M[S](\Sigma)) \Rightarrow \psi(\Sigma') = \text{"true"}$.

Jeloles: {φ}P{ψ}.

☐

Eredmenyesseg: $\varphi(\Sigma) = (fail \notin M[S](\Sigma))$;

Befejezodes: $\varphi(\Sigma) = (\perp \notin M[S](\Sigma))$;

☐

Teljes helyesseg. Az S programot a (φ, ψ) specifikacio szerint teljesen
helyesnek mondjuk, ha
$\forall\ \Sigma \in \Sigma$ kezdo ertekre, ha $\varphi(\Sigma)$ = "true",
a tranzakcio befejezodik es
$\Sigma' \in \Sigma$ eseten, amelyre $\psi(\Sigma')$="true":
- $\varphi(\Sigma) \Rightarrow (\{\perp, fail\} \cap M[S](\Sigma)=\{\})$;
- $[(\varphi(\Sigma)=\text{"true"}) \wedge (\Sigma' \in M[S](\Sigma))] \Rightarrow$
$\Rightarrow \psi(\Sigma')$ = "true".

Jeloles: {{φ}}P{{ψ}}.

☐

Induktiv allitasok (Floyd) modszere programok parcialis helyessegenek a bizonyitasara.

☐

Tranzakcio:
Szintaxis:
A tranzakcio egy (l, $\alpha \rightarrow r$, l') harmas.
l, l': cimke;
α: logikai kifejezes;
f: lekepezes;
Szemantika:
l cimketol k' cimkeig az f lekepezes valosul meg, ha α="true".

$$l \xrightarrow{\alpha \rightarrow f} l'$$

Tranzakcios diagram: (L, T, s, t) negyes, ahol
L az $l \in L$ cimek egy veges halmaza.
T a tranzakciok veges halmaza.
$s \in L$, egy kituntetett cim, az entry cim.
$t \in L$, egy kituntetett cim, az exit cim.

Σ allapotok halmaza;
$l, l' \in L$.
$\alpha: \Sigma \rightarrow$ bool.
$f: \Sigma \rightarrow \Sigma$, ahol egy allapot $\Sigma \in \Sigma$.
t cim, amelyre $\nexists$ ($l \in L$, $\alpha \rightarrow f \in T$)($t$, $\alpha \rightarrow f$, l).

Q-diagam.
Adva: PT = (L, T, s, t) tranzakcios diagram.
A Q-diagram a PT tranzakcios diagramnak egy olyan allitasokkal kiegeszitett formaja, amelyben egy Q fuggveny minden $l \in L$ cimkehez hozzarendel egy Q_l allitast.

Adva Q diagram a PT tranzakcios diagramhoz.
Egy $\pi = (l, \alpha \rightarrow r, l')$ tranzakcio verifikacios feltetele:
$V\_pi = Q\_l \wedge \alpha \Rightarrow Q\_\{l'\}$ **o** r.

A PT tranzakcios diagramhoz tartozo Q diagram osszes verifikacios feltetelenek a halmazat jelolje V(PT, Q).

A PT tranzakcios diagramhoz tartozo Q-diagramrol azt mondjuk, hogy az induktiv, ha ($\forall$ $V\_\pi \in$ V(PT, Q))(V_π="true").

A PT tranzakcios diagramhoz rendelt Q-diagramrol azt mondjuk, hogy az invarians, ha ($\forall$ $l\_i \in$ PT)($Q\_S(\Sigma\_0)$ = "true" $\Rightarrow$ $Q\_l\_i(\Sigma\_i)$="true").

Az induktiv Q-diagramot az adott (φ, ψ) specifikacio szerint konzisztensnek mondjuk, ha
$\varphi(\Sigma\_0) \Rightarrow Q\_s(\Sigma\_0)$ es $Q\_t(\Sigma\_n) \Rightarrow \psi(\Sigma\_s)$.

Floyd-fele induktiv allitasok modszere programok parcialis helyessegenek bizonyitasara.

1. Az adott P programhoz keszitsuk el a PT tranzakcios diagramot.
2. PT tranzakcios diagramhoz keszitsuk el a Q allitasokkal kiegeszitett Q-diagramot.
3. Bizonyitsuk be, hogy a Q-diagram induktiv es invarians.
4. Bizonyitsuk be, hogy a Q-diagram konzisztens a (φ, ψ) speifikacio szerint.

EA9

Kettos specifikacio.
Adott d_a=(A,F,E_a); d_c=(C, G, E_c);
A_0={aII_a(a)}; C_0={cII_c(c)};
abrazolas: φ : C_0 $\rightarrow$ A_0;

E_a={..., f_s(f_c(a))=h(a),....,I_a(f_c(a))="false" $\Rightarrow$ f_c(a)="undefined"};

E'_a= {...., {y=f_c(a)} z=f_s(y) {z=h(a)}, ..., {I_a(f_c(a))} z=f_c(a) {I_a(z) $\wedge$ z=f_c(a)}};

E'_a= {..., {y=f_c(a)$\wedge$ I_a(a)} z=f_s(y) {z=h(a)$\wedge$ I_a(a)},...,
{I_a(f_c(a))$\wedge$ I_a(a)} z=f_c(a) {I_a(z)$\wedge$ z=f_c(a) $\wedge$ I_a(a)}};
azaz
E'_a={..., {pre_f_i(a)} a'=f_i(a) {post_f_i(a,a')}, ...};

☐

a=φ(c);

☐

E_c = {..., g_s(g_c(c))=h_c(c),...,I_c(g_c(c)) = "false" ⇒ g_c(c) = "undefined"};
E_c = {..., pre_g_i(c)} c'=g_i(c) {post_g_i(c,c')},...};
E_c = {..., Q_g_i, ...};

☐

Ha bebizonyitjuk, hogy
1) I_c(c) ⇒ I_a(φ(c));
2) post_g_0(c) ⇒ I_c(c);
3) post_g_0(c) ⇒ post_f_0(φ(c));
4) I_c(c) \amd pre_f_i(φ(c)) ⇒ pre_g_i(c);
5) I_c(c) ∧ pre_f_i(φ(c)) ∧ post_g_i(c,c') ∧ I_c(c') ⇒ post_f_i(φ(c), φ(c'));
akkor a d_c konkret specifikacio a d_a absztrakt specifikacio szerint helyes.

☐

Ha a kovetkezo tetelek teljesulnek:
1. (∀ c ∈ C) (I_c(c) ⇒ I_a(φ(c));
2. {"true"} Q_0 {post_f_0(φ(c)) ∧ I_c(c)};
3. (∀ f_i∈ F): {pre_f_i(φ(c)) ∧ I_c(c)} Q_i {post_f_i(φ(c), φ(c')) \amd I_c(c')};
ahol 2. es 3. teljes helyessegi tetelek, akkor a d_c konkret specifikacio a d_a absztrakt specifikacio szerint helyes.

☐

Adott S determinisztikus program:
Tranzakcios diagram: (L, R, s, t);

s: u<-f;
({s,t},P(s, "true" → (u <- f), t)}, s, t);

```
      /-\"true" → (u<-f)   /-\
      |s|-----------------> |t|
      \-/                  \-/
```

(az s-et es a t-t be kell karikazni)

☐

S: S_1;S_2;
({s, r, t}, {(s, "true" → (u<- f_s_1), r), (r, "true" → (u<- f_s_2), t)}, s, t);

☐

```
   "true" → (u<-f_s_1)      "true" → (u<-f_s_2)
s ---------------------> r -----------------------> t
```

(s, r es t be van karikazva)

☐

```
           S_1T      S_2T
s           r           t
```

(s, r es t be van karikazva, r S_2T t egy kozos teglalapban van, aminek alanyulik az s S_1T teglalapja)

☐

S: if α then S_1 else S_2 fi;
({s, t}, {(s, α → (u<-f_s_1), t), (s, ¬α → (u<-f_s_1), t)}, s, t);

Q-diagram: Q=(Q_s, Q_t);

```
                Q_s
              /----s----\
              |         |
α → f_S_1     |         | ¬α → f_S_2
              |         |
              \--->t<---/
                 Q_t
```

(s es t be van keretezve)

```
     S_1T

s         t

     S_2T
```

(s es te be van keretezve; s, t es S_2T egy kozos konkav bigyoval be van keretezve, es ez ala nyulik S_1T, s es t bekeretezett konkav bigyoja)

☐

Adott {φ} S{ψ} parcialis helyessegi tetel.
Adott az S program ST = (L, T, s, t) tranzakcios diagramja, es Q diagramja, amelyrol bebozonyitottuk, hogy induktiv.
Ha bebizonyitjuk, hogy a Q-diagram a (φ, ψ) specifikacio szerint konzisztens, azaz

$\varphi \Rightarrow Q_s$; $Q_t \Rightarrow \psi$;
akkor az S program a (φ, ψ) specifikacio szerint parcialisan helyes.

□

A program befejezodesenek (konvergens tulajdonsaganak) bizonyitasa.
φ - konvergencia bizonyitasa.

□

Alapfogalom.
Jol rendezett halmaz.
Legyen W egy halomaz es $<$:WxW beinaris relacio. A $<$ relaciot rendezonek mondjuk, ha tetszoleges a, b, c $\in$ W -re:
- irreflexiv: $a < a$ ="false".
- asszimetrikus: $a<b$ ="true" $\Rightarrow$ $b<a$="false".
- tranzitiv: ($a<b$ = "true") $\wedge$ ($b<c$="true") $\Rightarrow$ $a<c$="true".

A parcialisan rendezett (W, $<$) halmazt jol rendezettnek nevezzuk, ha nem letezik vegtelen $\ldots < w_2 < w_1 < w_0$, sorozat, $w_i \in W$, eseten.

□

φ - konvergencia bizonyitasanak Floyd-fele modszere.

Allitas:
Adott a P program PT =(L, R, s, t) tranzakcios diagramja es φ.
1) Keszitsuk el PT alapjan a kovergencia bizonyitasahoz szukseges Q-diagramot.
2) Bizonyitsuk be, hogy Q-diagram induktiv es $\varphi \Rightarrow Q_s$.
3) Valasszunk meg egy (W, $<$) jol rendezett halmazt es a
$\rho=(\rho_l | l\in L\}$, fuggvenyhalmazt, ahol $p_l:\Sigma \rightarrow W$,
minden $l\in L$ eseten.
4) Bizonyitsuk be, hogy ρ_l definialva van: $Q_l(\Sigma) \Rightarrow \rho_l \in W$.
5) Mutassuk ki, hogy $(\forall(l, \alpha \rightarrow f, l')\in R)(Q_l(\Sigma)\wedge\alpha(\Sigma)\Rightarrow\rho\ l'(f(\Sigma))<\rho\ l(\Sigma))$.
Ha ezeket bebizonyitottuk, akkor a P program φ-konvergens.

□

Bizonyitas:
v: $<s, \Sigma_0> \rightarrow <l_1, \Sigma_1> \rightarrow \ldots$;
Q induktivitasabol kovetkezik, hogy Q invarians. Ha $\varphi \Rightarrow Q_s$, akkor
$\rho_s(\Sigma_0)$, $\rho_{l\_1}(\Sigma_1)$, ... definialva van es minden $<l, \Sigma> \rightarrow <l', \Sigma'>$
tranzakciora $\rho_{l'}(f(\Sigma))<\rho_l(\Sigma)$.
Igy W jol-rendezettsegebol kovetkezik, hogy a vegrehajtas veges.

□

Vegrehajtasi (runtime) hibamentesseg.
Tekintsuk az S programnak azokat a vegrehajtasait:
v: $<s, \Sigma_0> \rightarrow <l_1, \Sigma_1> \rightarrow \ldots$;
amelyekre $\varphi(\Sigma_0)$="true".
Ha nincs olyan vegrehajtas, amelynek soran valamely cimkenel a szoba joheto tranzakciok kozott letezik olyan, amely nincsen definialva, akkor azt mondjuk, hogy az S program a φ input specifikacio mellett mentes a vegrehajtasi hibatol.

□

Vegrehajtasi hibamentesseg bizonyitasa:
Adott a P program PT=(L, R, s, t) tranzakcios diagramja es φ. Keszitsuk el annak Q-diagramjat. Ha minden $l\in L$ eseten az osszes
$(l, \alpha\rightarrow f, l')$ tranzakcional $Q_l(\Sigma)\wedge(\alpha(\Sigma)\Rightarrow pre_f(\Sigma)$, akkor a P program mentes a vegrehajtasi hibatol.

□

P program tejes-helyessegenek bizonyitasa.

Adott a P program es annak (φ, ψ) specifikacioja.
- Konstrualjuk meg a P programhoz a vele szemantikailag ekvivalens PT diagramot.
- Keszitsuk el a Q diagramot.
- Bizonyitsuk be, hogy a P program parcialisan helyes az adott specifikacio szerint.
- Bizonyitsuk be, hogy a P program kovergens.
- Bizonyitsuk be, hogy a P program mentes a vegrehajtasi hibatol.

Akkor a P program a (φ, ψ) specifikacio szerint teljesen helyes.

EA10

Realizacio:
r: INT $\rightarrow$ MOD;
r= (r_P, r_E, r_I);

□

Kettos specifikacio.

E_c={..., g_s(g_c(c))=h_c(c), ..., I_c(g_c(c))="false" $\Rightarrow$ g_c(c)="undefined"};

E_c={..., Q_g_i, ...};

□

Program helyesseg.

$\{\varphi\}S\{\psi\}$

Program vegrehajtasa.
<S, Σ> →<S_1, Σ_1> → ... → <S_{n-1}, Σ_{n-1}> → <S_n, Σ_n>;
<s, Σ> →<l_1, Σ_1> → ... → <l_{n-1}, Σ_{n-1}> → <l_n, Σ_n>;

Tranzakcios diagram. PT.
Floyd modszer.

Determinisztikus programok helyessegenek bizonyitasa Hoare modszerrel.

Hoare fele harmas: {p}S{q}.

Definíció: A {p}S{q} formulat parcialis helyessegi ertelemben, helyesnek mondjuk, ha M[S]([{p}])⊂ {q}.
Jeloles: {p} S {q} = "true".

Definíció: A {{p}}S{{q}} formulat teljes helyessegi ertelemben, helyesnek mondjuk, ha M_{tot}[S]([{p}])⊂ {q}.
Jeloles: {{p}}S{{q}}="true".

Hoare fele bizonyitasi rendszer (BR):
axiomak + kovetkeztetesi szabalyok.
Dedukcio: axioma + kovetkeztetesi szabalyok ⇒ tetel.
BD: Determinisztikus programok bizonyitasi rendszere.

Determinisztikus program:
S::= skip | u<-t | S_1;S_2 | if α then S_1 else S_2 fi | while α do S od

Axiomak:

$\{P(x, y)\}\ \text{skip}\ \{P(x, y)\}$
$\{P(x,g(x,y))\}\ y \leftarrow g(x,y)\ \{P(x,y)\}$

Kovetkeztetesi szabalyok:

- Szekvencia:
$$\frac{\{P\}S\_1\{Q\_1\} \text{ es } \{Q\_1\}S\_2\{Q\}}{\{P\}S\_1;S\_2\{Q\}}$$
- Felteteles elagazas:
$$\frac{\{P \land \alpha\}S\_1\{Q\} \text{ es } \{P \land \neg \alpha\}S\_2\{Q\}}{\{P\} \text{ if } \alpha \text{ then } S\_1 \text{ else } S\_2 \text{ fi } \{Q\}}$$
- Iteracio:
$$\frac{\{P\land\alpha\}S\{P\} \text{ es } P\land\neg\alpha \Rightarrow Q}{\{P\} \text{ while } \alpha \text{ do } S \text{ od } \{Q\}}$$

A kovetkezmeny szabalya:
$$\frac{P \Rightarrow P\_1 \text{ es } \{P\_1\}S\{Q\_1\} \text{ es } Q\_1 \Rightarrow Q}{\{P\}\ S\ \{Q\}}$$

Ertekadas kovetkeztetesi szabalya:
$$\frac{p(x,f(x,y)) \Rightarrow q(x,y)}{\{p(x,y)\}y \leftarrow f(x,y)\{q(x,y)\}}$$

Iteracio kovetkeztetesi szabalyanak altalanos formaja:
$$\frac{P\Rightarrow I \text{ es } \{I\land \alpha\}S\{I\} \text{ es } I\land\neg\alpha \Rightarrow Q}{\{P\} \text{ while } \alpha \text{ do } S \text{ od } \{Q\}}$$

A felsorolt axiomak es kovetkeztetesi szabalyok alkotjak a
determinisztikus programok parcialis hwelyessegenek bizonyitasara
szolgalo bizonyitasi rendszert.
Jeloles: PD.

☐

A teljes helyesseg bizonyitasanak kovetkeztetesi szabalya.

$P(x,y) \Rightarrow I(x,y)$ es $I(x,y) \Rightarrow E(x,y) \in W<$ es

$\{\{I(x,y) \wedge \alpha(x,y) \wedge E=E(x,y)\}\}S\{\{I(x,y) \wedge E<E(x,y)\}\}$ es

$I(x,y) \wedge \neg\alpha(x,y) \Rightarrow Q(x,y)$

$\{\{P(x,y)\}\}$while $\alpha(x,y)$ do S od $\{\{Q(x,y)\}\}$

☐

Az iteracio kovetkeztetesi szabalya a ciklusszamlaloval:
$P(x,y) \Rightarrow I(x,y,0)$ es $I(x,y,i) \Rightarrow i<k(x)$ es $\{\{I(x,y,i) \wedge \alpha(x,y)\}\}S$
$\{\{I(x,y,i+1)\}\}$ es $I(x,y,i) \wedge \neg\alpha(x,y) \Rightarrow Q(x,y)$

$\{\{P(x,y)\}\}$ while $\alpha(x,y)$ do S od $\{\{Q(x,y)\}\}$

☐

A felsorolt axiomak es kovetkeztetesi szabalyok alkotjak a determinisztikus programok teljes helyessegenek bizonyitasara szolgalo bizonyitasi rendszert.
Jeloles: TD.

☐

Adott a bizonyitasi rendszer: BR.
Jeloles: {P}S{Q}/BR_{seq}="true" amelynek jelentese, hogy a {p}S{q} formula parcialis helyessegi ertelemben, levezetheto, bizonyithato a BR rendszerben.

☐

Adott a bizonyitasi rendszer: BR.
Jeloles: {{P}}S{{Q}}/BR_{seq}="true", amelynek jelentese, hogy a {{p}}S{{q}} formula teljes helyessegi ertelemben, levezetheto, bizonyithato a BR rendszerben.

☐

Definíció: Adott a BR bizonyitasi rendszer, es a programoknak egy C osztalya.

A BR bizonyitasi rendszert megbizhatonak mondjuk a C osztaly programjainak parcialis helyessegere vonatkozoan, ha minden $S \in C$ programra vonatkozo {P}S{Q} formulara {P}S{Q}/BR_{seq}="true" ⇒ {P}S{Q}="true".
A BR bizonyitasi rendszert megbizhatonak mondjuk a C osztaly programjainak teljes helyessegere vonatkozoan, ha minden $S \in C$ programra vonatkozo {{P}}S{{Q}} formulara {{P}}S{{Q}}/BR_{seq}="true"⇒{{P}}S{{Q}}="true".

☐

Definíció: Adott a kovetkezo formaju bizonyitasi szabaly:

$\varphi_1, ..., \varphi_k$

φ_{k+1}

A bizonyitasi szabalyt megbizhatonak nevezzuk parcialis (totalis) helyessegi ertelemben az adott C osztalyban, ha
φ_1="true"$\wedge ... \wedge\varphi_k$="true"$\Rightarrow\varphi_{k+1}$="true";
parcialis ill. totalis helyessegi ertelemben.

☐

Tétel. A PD bizonyitasi rendszer determinisztikus programok parcialis helyessegenek a bizonyitasara megbizhato.

Tétel. A TD bizonyitasi rendszer determinisztikus programok teljes helyessegenek a bizonyitasara megbizhato.

☐

Definíció: Adott a BR bizonyitasi rendszer, es a programoknak egy C osztalya.
A BR bizonyitasi rendszert teljesnek mondjuk a C osztaly programjainak parcialis helyessegere vonatkozoan, ha minden $S \in C$ programra vonatkozo helyessegi {P}S{Q} formulara
{P}S{Q}="true"⇒{P}S{Q}/BR_{seq}="true".
A BR bizonyitasi rendszert teljesnek mondjuk a C osztaly programjainak teljes helyessegere vonatkozoan, ha minden $S \in C$ programra vonatkozo helyessegi
{{P}}S{{Q}} formulara {{P}}S{{Q}}="true"⇒{{P}}S{{Q}}/BR_{seq}="true".

☐

Tétel. A PD bizonyitasi rendszer determinisztikus programok parcialis helyessegenek bizonyitasara teljes.

Tétel. A TD bizonyitasi rendszer determinisztikus programok teljes helyessegenek bizonyitasara teljes. (Az iteraciok szamara tett bizonyos megszoritasok eseten.)

☐

A nem teljesseg okai lehetnek:

1) A bizonyitasi rendszer nem teljes az allitasok kovetkezmenyeinek meghatarozasanal. (Godel Incompleteness Theorem)
2) Az allitasok leirasara hasznalt nyelv nem eleg teljes a helyessegi bizonyitas soran az allapotok es korlatozo fuggvenyek leirasara. (Megjavitom, de mindig lehet talalni olyan allitast, amit nem tudok bizonyitani.)
3) A bizonyitasi szabalyok az adott C osztalyra nezve nem teljesek.

□

Megjegyzes!
Definíció: Egy P determinsiztikus program es p,q predikatum eseten {p}P{q} helyessegi formulat igaznak mondjuk, ha {p} PT {q} igaz.

Definíció: Adott egy S determinisztikus program, amelynek programfuggvenye f_s(x,y).p(x,y),q(x,y) predikatum eseten a {p(x,y)}S{q(x,y)} helyessegi formulat igaznak mondjuk, ha P(x, f_s(x,y)) ⇒ q(x,y).

□

Pelda.

Legyen x,y∈ nat;
{x>=0∧ y>=0} div {quo × y + rem=x ∧ 0<=rem<y};
div:quo<-0;rem<-x;
while rem>=y do rem<-rem-y; quo<quo+1 od;

I:quo × y + rem=x ∧ rem>=0;

{x>=0 ∧ y>=0} quo=0; rem=x {I};

{I∧ rem>=y} rem<-rem-y;quo<-quo+1{Y};

(I∧¬(rem>=y)) ⇒(quo × y + rem=x ∧ 0<=rem<y)

div: quo<-0; rem<-x;
while rem>y do rem <- rem-y ; quo<-quo+1 od;

(× a direktszorzas keresztje)

□

Annotalt program:
div*:
quo<-0;rem<-x;
{I} while rem>=y do rem<-rem-y; quo<-quo+1 od;

□

Teljes helyesseg bizonyitasas.
E : rem;
I': I∧ y >0;

{x>=0 ∧ y>=0} quo=0; rem=x{I'};
{I'∧ rem>=y} rem<-rem-y; quo<-quo+1{I'};

{I'∧ rem>=y∧ rem=z}
rem<-rem-y;quo<-quo+1
{rem<z};

I'⇒ rem>=0;

EA11

{p}S{q}/BR_{sec}="true".
{p}S{q}="true".

Tétel. A PD bizonyitasi rendszer determinisztikus programok parcialis helyessegenek a bizonyitasara megbizhato.

Tétel. A TD bizonyitasi rendszer determinisztikus programok teljes helyessegenek a bizonyitasara megbizhato.

□

Az iteraciorol: while α do S od;
Ures iteracio: while "true" do skip od;

k=0 ⇒ (while α do S od)^k = while "true" do skip od;
k>=0 ⇒ (while α do S od)^{k+1}=
if α the S ; (while α do S od)^k else skip fi;

□

A szemantikarol: M[S](H); H = allapotok halmaza.
- Monoton: H_1 ⊂ H_1 ⇒ M[S](H_1) ⊂ M[S](H_2);
- M[S_1;S_2](H) = M[S_1](M[S_1](H));
- M[begin S_1; S_1 end; S_3](H) = M[S_1; begin S_2; S_3 end](H);
- M[if α then S_1 else S_2 fi](H) = M[S_1](H∈tersect {α})∪ M[S_2](H∈tersect {¬α});
- M[while α do S od] = ∪_{k=0}^{∈f}(while α do S od)^k;

Bizonyitas:

- Mutassuk meg, hogy minden axioma a PD ill. TD rendszerben igaz, azaz megbizhatoak.
- Mutassuk meg, hogy minden kovetkeztetesi szabaly a PD ill. TD rendszerben igaz, azaz megbizhatoak.
- A fentiakbol ezekutan teljes indukcioval kovetkeznek az allitasaink.

Mit jelent az, hogy egy axioma igaz?

Definíció:

Axioma: $\{\Sigma |\ p(\Sigma)\} \langle S,\Sigma\rangle \to \langle E, \tau\rangle \{\tau | q(\tau)\}$;

$\{p(\Sigma)\} \langle S, \Sigma\rangle \to \langle E, \tau\rangle \{q(\tau)\}$];

$\{p\} \langle s, \Sigma\rangle \to \langle E, \tau\rangle \{q\}$;

Ha $M[S](\{p\}) \subset \{q\}$; akkor az axioma igaz.

$S=skip$; $\langle skip, \Sigma\rangle \to \langle E, \Sigma\rangle$;

Axioma: $\{p\}$ skip $\{p\}$;

$M[skip](\{p\}) = \{p\} \Rightarrow \{p\}\ S\ \{p\}$ = "true";

$S=y\leftarrow f(x,y); \langle y\leftarrow f(x,y),\Sigma\rangle \to \langle E, \tau\rangle; \tau=\Sigma[y\leftarrow f(x,y)]$;

$M[y\leftarrow f(x,y)](\Sigma) = \{\tau\}$;

$M[y\leftarrow f(x,y)](\Sigma) = \{\Sigma[y\leftarrow f(x,y)]\}$;

Axioma: $\{p(x,f(x,y))\}y\leftarrow f(x,y)\{p(x,y)\}$;

(∀Σ∈ {p})(\sziga[y<-f(x,y)]∈{p})

$M[y\leftarrow f(x,y)](\{p\}) \subset \{p\} \Rightarrow \{p\}S\{p\}$ = "true";

Mit jelent az, hogy a

$$\frac{\varphi_i, \ldots, \varphi_k}{\varphi_{k+1}}$$

kovetkeztetesi szabaly igaz?

Definíció:

Ha $((\varphi_l = \text{"true"}) \wedge \ldots \wedge (\varphi_k=\text{"true"})) \Rightarrow (\varphi_{k+1} = \text{"true"})$,
akkor a kovetkeztetesi szabaly igaz, azaz megbizhato.

$S:S_1;S_2$;

Felteves:

- $M[S_1](\{p\}) \subset \{r\}$;
- $M[S_2](\{r\}) \subset \{q\}$;

$M[S_2](M[S_1](\{p\}) \subset M[S_2](\{r\}) \subset \{q\} \Rightarrow M[S_1;S_2](\{p\}) \subset \{q\} \Rightarrow$

$\Rightarrow \{p\}S_1, S_2\ \{q\} = \text{"true"} \equiv \{p\}S\{q\}=\text{"true"}$;

A

$$\frac{\{p\}S_1\{r\},\ \{r\}S_2\{q\}}{\{p\}S_1;S_2\{q\}}$$

kompozicio szabalya tehat parcialis helyessegi ertelemben megbizhato.

S: if α then S_1 else S_2 fi;

Felteves:

- $M[S_1](\{p\wedge \alpha\}) \subset\{q\}$;
- $M[S_2](\{p\wedge \neg\alpha\})\subset\{q\}$;

$M[\underline{if}\ \alpha\ S_1\ \underline{else}\ S_2\ \underline{fi}](\{p\}) = ((M[S_1](\{p\wedge\alpha\})\cup(M[S_2](\{p\wedge\neg\alpha\})\subset\{q\}) \Rightarrow$

$\{p\}\ \underline{if}\ \alpha\ \underline{then}\ S_1\ \underline{else}\ S_s\ \underline{fi}\ \{q\} = \text{"true"} \equiv \{p\}S\{q\} = \text{"true"}$;

A felteteles elagazas

$$\frac{\{p\wedge\alpha\}S_1\{q\},\ \{p\wedge\neg\alpha\}S_2\{q\}}{\{p\}\text{ if } \alpha \text{ then } S_1 \text{ else } S_2 \text{ fi } \{q\}}$$

kovetkeztetesi szabalya parcialis helyessegi ertelemben megbizhato.

S= while α do S_1 od;

Felteves:

$M[S_1](\{p\wedge\alpha\})\subset\{p\}$;

$(\forall\ k, k>=0)(M[(\text{while } \alpha \text{ do } S_1 \text{ od})^k](\{p\}) \subset \{p\wedge\neg\alpha\})$;

k=0;

Felteves k>= eseten igaz, bizonyitsuk

$M[(\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}})^{k+1}](\{p\}) \subset \{p\wedge\neg\alpha\}$;

$M[(\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}})^{k+1}](\{p\})=M[\underline{\text{if}}\ \alpha\ \underline{\text{then}}\ S_1;\ (\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}})^k\ \underline{\text{else}}\ \text{skip}\ \underline{\text{fi}}](\{p\})=$
$M[S_1;\ (\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_2\ \underline{\text{od}})^k](\{p\wedge \alpha\})\cup M[\text{skip}](\{p\wedge\neg\alpha\})=$
$M[(\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}})^k](M[S_1](\{p\wedge\alpha\}))\cup \{p\wedge\neg\alpha\}\subset$
$M[(\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}})^k](\{p\}) \cup \{p\wedge\neg\alpha\})\subset\{p\wedge\neg\alpha\}$.

$\cup_{k=0}^{\in f} M[(\text{while}\ \alpha\ \text{do}\ S_1\ \text{od})^k](\{p\}) \subset \{p\wedge \neg\alpha\}$.

$M[\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}}](\{p\}) = \cup_{k=0}^{\in f} M[(\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}})^k](\{p\}) \subset \{p\wedge\neg\alpha\}$.
$M[\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}}](\{p\}) \subset \{p\wedge \neg\alpha \} \Rightarrow \{p\}\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}}\{p\wedge \neg\alpha\} \equiv$
$\{p\}S\{p\wedge\neg\alpha\}=$"true".

Az iteracio

$\{p\wedge\alpha\}S_1\{p\}$

$\{p\}\ \underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}}\ \{p\wedge\neg\alpha\}$

kovetkeztetesi szabalya parcialis helyessegi ertelemben helyes.

□

Kovetkezmeny szabalya.
Felteves:
$p\Rightarrow p_1$, $M[S](\{p_1\})\subset\{q_1\}$;$q_1\Rightarrow q$;
$\{\Sigma|p(\Sigma)\}\subset\{\Sigma|p_1(\Sigma)\}$; azaz $\{p\}\subset\{p_1\}$;
$\{\Sigma|q_1(\Sigma)\}\subset\{\Sigma|q(\Sigma)\}$; azaz $\{q_1\}\subset\{q\}$;

$M[S](\{p\})\subset M[S](\{p_1\})\subset \{q_1\}\subset\{q\}$;

□

Teljes helyesseg.
$S=\underline{\text{while}}\ \alpha\ \underline{\text{do}}\ S_1\ \underline{\text{od}}$;
Felteves.
- $M_{tot}[S](\{p\wedge \alpha\})\subset\{p\}$;
- $M_{tot}[S](\{p\wedge \alpha \wedge t=z\})\subset \{t<z\}$;
- $p \Rightarrow t>=0$;
- z integer valtozo es nem fordul elo p, α, t, S formulakban;

Allitas. $\perp \notin M_{tot}[S](\{p\})$.

□

Adott:
- $S(x,y)$: while $\alpha(x,y)$ do $A(x,y)$ od,
  iteracio, a $f_s(x,y)$=if $\neg\alpha(x,y)$ then y else $f_s(x,f_s(x,y))$ fi programfuggvenyel,
  $\alpha(x,y)$ kvantor - fuggetlen logikai kifejezes.
- $\{P(x,y)\}S(x,y)\{R(x,y)\}$;
- $I(x,y,i)$: ciklus invarians,
- $k(x)$: a ciklusszamlalo felso korlatja,
- $f_A(x,y)$: a ciklusmag programfuggvenye.

□

Ha bebizonyitjuk, hogy
- $P(x,y) \Rightarrow I(x,y,0)$,
- $I(x,y,i) \Rightarrow i<=k(x)$,
- $I(x,y,i)\wedge\alpha(x,y) \Rightarrow I(x, f(x,y), i+1)$,
- $I(x,y,i) \wedge \neg\alpha(x,y) \Rightarrow R(x,y)$,

akkor minden olyan y_0-ra, amelyre $P(x,y_0)$="true",
letezik $f_s(x,x_0)$ es $R(x, f_s(x,y_0))$ = "true".

□

Bizonyitas. A bizonyitas k(x) szerinti teljes indukcioval.
Alapeset. $k(x) =0$.
$k(x) = 0 \Rightarrow \alpha(x,y_0)$ = "false".

Tegyuk fel ugyanis: $\alpha(x,y_0)$ = "true",
akkor $I(x, f_A(x, y_0), 1)$ = "true", es $1<=k(x)$, ami ellentmondas.
Igy $I(x,y_0, 0) \wedge \neg\alpha(x,y_0) \Rightarrow R(x,y_0)$, amde, most $f_s(x,y_0)=y_0$.

□

Indukcio. $k(x)>0$ es felteves $k'(x)<=k(x)-1$-re az allitas igaz.
- $\alpha(x,y_0)$="false". Ekkor ugyanugy, mint fent belathato, hogy igaz az allitas
- $\alpha(x,y_0)$="true". Ekkor $I(x, f_A(x, y_0), 1)$ = "true".
  Legyen tehat
  $y_1 = f_A(x,y_0)$,
  $I_1(x,y_1,i)=I(x,f_A(x,y_0), i+1)$.

Ekkor az uj ciklusszamlalo korlatja:
$I_1(x,y_1,i) \Rightarrow i<=k(x)-1$.

y_1-re tehat letezik $f_s(x,y_1)$, es erre $R(x, f_s(x,y_1))$ = "true", azaz
$R(x, f_s(x, f_A(x,y_0))$ = "true"
$= \quad f_s(x,y_0))$.

Ezzel az iteracio kovetkeztetesi szabalyanak helyesseget bebizonyitottuk.

EA12

Adott BR bizonyitasi rendszer es BR_{sec} a bizonyitasi rendszerben levezetheto formulak halmaza.

- A BR bizonyitasi rendszer megbizhato, ha $(\forall\ \varphi, \varphi \in BR\_\{sec\}) \Rightarrow \varphi =$ "true".
- A BR bizonyitasi rendszer teljes, ha $(\forall \varphi, \varphi \in BR \wedge \varphi=$"true"$) \Rightarrow \varphi \in BR\_\{sec\}$.

A Hoare modszer teljessegi tetele.

Adva S(x,y) strukturalt program,
S(x,y) tetszoleges reszprogramja: s(x,y).

Felteves:
$\{Q(x) \wedge y=I\_S(x)\}\ s(x,y)\ \{Q(x) \wedge y=O\_S(x)\} =$ "true", ahol $f\_s(x, I\_S(x))=O\_S(x)$.

Allitas. Minden ilyen tulajdonsagu s(x,y) resz - programra vonatkozo fenti tetel, A Hoare-modszer segitsegevel levezetheto.

Megjegyzes. Nyilvan:
$\{Q(x)\wedge y=x\}\ S(x,y)\ \{Q(x)\wedge y=O\_s(x)\}=$"true",
azaz
$P(x){:}Q(x)\wedge y=x;\ R(x,y){:}Q(x) \wedge y=O\_s(x);$
$\{P(x)\}\ S(x,y)\ \{R(x,y)\} =$ "true".

Bizonyitas. (parcialis helyesseg) Az alapstrukturak egymasba skatulyazasanak szama szerinti teljes indukcioval.

Alapeset. $s(x,y) = y \leftarrow g(x,y)$
A tetel: $\{Q(x) \wedge y=I\_s(x)\}\ y \leftarrow g(x,y)\ \{Q(x) \wedge y=O\_s(x)\} =$ "true",

$$\frac{Q(x) \wedge y=I\_s(x) \Rightarrow Q(x) \wedge g(x,y)=O\_s(x)}{\{Q(x) \wedge y=I\_s(x)\}\ y \leftarrow g(x,y)\ \{ Q(x) \wedge y=O\_s(x)\}}$$

Indukcio. Tegyuk fel, hogy "k" melysegu egymasva skatulyazas eseten a tetel igaz es
bizonyitsuk "k+1"-re is.
Harom eset:
- a k+1. struktura egy szekvencia;
- a k+1. struktura egy felteteles elagazas;
- a k+1. struktura egy iteracio.

a.) A k+1. struktura egy szekvencia: $s(x,y) = s\_1(x,y)\ ;\ s\_2(x,y);$
A program fuggvenyek: $f\_s\_1(x,y),\ f\_s\_2(x,y)$.
Indukcios feltevesunk:
$\{Q(x) \wedge y=I\_s\_1(x)\}\ s\_1(x,y)\ \{Q(x) \wedge y=O\_s\_1(x)\}$,
$\{Q(x) \wedge y=I\_s\_2(x)\}\ s\_2(x,y)\ \{Q(x) \wedge y=O\_s\_2(x)\}$,
tetelek helyessege Hoare modszerrel bebizonyithatoak.
azaz
$O\_s\_1(x)=f\_s\_1(x,I\_s\_1(x))$, es $O\_s\_2(x) = f\_s\_2(x,I\_s\_2(x))$.

A szekvencia szemantikaja alapjan:
$I\_s(x) = I\_s\_1(x)$, es $O\_s\_1(x) = I\_s\_2(x)$ es $O\_s\_2(x) = O\_s(x)$;
ezert
$f\_s(x, I\_s(x)) = f\_s\_2(x,f\_s\_1(x, I\_s\_1(x)))$.

$$\frac{\{Q(x)\wedge y=I\_s\_1(x)\}\ S\_1(x,y)\{Q(x)\wedge y=O\_s\_1(x)\},\quad \{Q(x)\wedge y=I\_s\_2(x)\}\ S\_2(x,y)\{Q(x)\wedge y=O\_s\_2(x)\},}{\{Q(x)\wedge y=I\_s(x)\}S\_1(x,y);S\_2(x,y)\{Q(x)\wedge y=O\_s(x)\}}$$

b.) a k+1. struktura egy felteteles elagazas:
$s(x,y) =$ if $\alpha(x,y)$ then $S\_1(x,y)$ else $S\_2(x,y)$ fi,
Indukcios feltevesunk szerint:
$\{Q(x) \wedge y=I\_s\_1(x)\}S\_1(x,y)\{Q(x)\wedge y=O\_s\_1(x)\}$,
$\{Q(x) \wedge y=I\_s\_2(x)\}S\_2(x,y)\{Q(x)\wedge y=O\_s\_2(x)\}$,
tetelek helyessege Hoare modszerrel bebizonyithatok.
$(Q(x)\wedge y=I\_s(x,y) \wedge\alpha(x,y)) \Rightarrow (Q(x)\wedge y=I\_s\_1(x,y))$,
$(Q(x)\wedge y=I\_s(x,y) \wedge\neg\alpha(x,y)) \Rightarrow (Q(x)\wedge y=I\_s\_2(x,y))$,
masreszt
$(Q(x)\wedge y=I\_s(x,y)\wedge \alpha(x,y))\Rightarrow O\_s(x)=O\_s\_1(x)$,
$(Q(x)\wedge y=I\_s(x,y)\wedge \neg\alpha(x,y))\Rightarrow O\_s(x)=O\_s\_2(x)$,

$Q(x) \wedge y=I\_s(x,y) \wedge \alpha(x,y)) \Rightarrow (Q(x) \wedge y=I\_s\_1(x,y))$,

$(Q(x) \wedge y=I\_s(x,y) \wedge \neg\alpha(x,y)) \Rightarrow (Q(x) \wedge y=I\_s\_2(x,y))$,

$\{Q(x) \wedge y=I\_s\_1(x))\}\ S\_1(x,y) \{Q(x) \wedge y=O\_s\_1(x)\}$,

$\{Q(x) \wedge y=I\_s\_2(x)\} S\_2(x,y)\ \{Q(x) \wedge y=O\_s\_2(x)\}$,

$(Q(x) \wedge y=O\_s\_1(x)) \Rightarrow y=O\_s(x)$,

$(Q(x) \wedge y=O\_s\_2(x)) \Rightarrow y=O\_s(x)$,

$\{Q(x) \wedge y=I\_s(x)\}$

if $\alpha\ (x,y))$ then $S\_1(x,y)$ else $S\_2(x,y)$ fi

$\{Q(x) \wedge y=O\_s(x)\}$.

c.) k+1. struktura egy iteracio:

s(x,y) = while $\alpha(x,y)$ do A(x,y) od.

$\{Q(x) \wedge y=I\_A(x)\}A(x,y)\{Q(x) \wedge y=O\_A(x)\}$

mar bizonyithato a Hoare modszerrel.

Legyen A(x,y) programfuggvenye: f_A(x,y).
A tetel, amely bizonyithato: $\{Q(x) \wedge y=I\_A(x)\}\ y \text{<-} f\_A(x,y)\ \{Q(x) \wedge y=O\_A(x)\}$.

Jeloles:

$k=0 \Rightarrow h(x,y,k) = y$.

$k>0 \Rightarrow h(x,y,k) = f\_A(x, f\_A(x, ...(f\_A(x,y))))$;

1 2 ... k

Az iteracio szemantikajat leiro invarians:
$I(x,y,k){:}Q(x) \wedge y=h(x,I\_A(x),k) \wedge f\_s(x,y) = f\_s(x, I\_s(x))$.

A bizonyitando tetelek:

$(Q(x) \wedge y=I\_s(x)) \Rightarrow I(x,y,0)$;

$\{I(x,y,k) \wedge \alpha(x,y)\}\ A(x,y)\ \{I(x,y,k+1)\ \}$;

$(I(x,y,k) \wedge \neg\alpha(x,y)) \Rightarrow (Q(x) \wedge y=O\_s(x))$.

1. tetel $(Q(x) \wedge y=I\_s(x)) \Rightarrow I(x,y,0)$,

$(Q(x) \wedge y=I\_s(x)) \Rightarrow (Q(x) \wedge y=I\_A(x) \wedge f\_s(x,y) = f\_s(x, I\_s(x)))$, ami trivialis.

2. tetel. $\{I(x,y,k) \wedge \alpha(x,y)\}\ A(x,y)\ \{\ I(x,y,k+1)\}$,

az ertekadas kovetkeztetesi szabalya alapjan:

$(I(x,y,k) \wedge \alpha(x,y)) \Rightarrow I(x,f\_A(x,y), k+1)$,

$(Q(x) \wedge y=h(x,I\_A(x), k) \wedge f\_s(x,y) = f\_s(x, I\_s(x)) \wedge \alpha(x,y)) \Rightarrow$

$(Q(x) \wedge f\_A(x,y)=h(x, I\_A(x), k+1) \wedge f\_s(x, f\_A(x,y)) = f\_s(x,I\_s(X))$.

3. tetel. $(I(x,y,k) \wedge \neg\alpha(x,y)) \Rightarrow (Q(x) \wedge y=O\_s(X))$, azaz

$(Q(x) \wedge y=h(x,I\_A(x), k) \wedge f\_s(x,y) = f\_s(x,I\_S(x)) \wedge \neg\alpha(x,y)) \Rightarrow (Q(x) \wedge y=O\_S(x))$.

Mivel $\neg\alpha(x,y)$ eseten $f\_s(x,y)=y$, masreszt a definicio alapjan $f\_s(x,I\_s(x)) = O\_s(x)$.

$Q(x) \wedge y=I\_s(x) \Rightarrow I(x,y,0)$;

$\{I(x,y,k) \wedge \alpha(x,y)\}\ A(x,y)\ \{\ I(x,y,k+1)\ \}$;

$I(x,y,k) \wedge \neg\alpha(x,y) \Rightarrow Q(x) \wedge y=O\_s(x)$.

$\{Q(x) \wedge y=I\_s(x)\}$ while $\alpha(x,y)$ do A(x,y) od $\{Q(x) \wedge y=O\_s(x)\ \}$.

Adva d_s=(A, F, E_a); absztrakt specifikacio
d_c=(C,G,E_c); konkret specifikacio.

$A=\{A\_0, A\_1, ..., A\_n\}$; $F=\{f\_0, f\_1, ..., f\_m\}$;

$C=\{C\_0, C\_1, ..., C\_n\}$; $G=\{g\_0, g\_1, ..., g\_m\}$;

$E\_a = \{..., e\_a\_i(..., f\_j(a), ...; a), ...\}$; $E\_c=\{...., i\_c\_i(..., g\_j(c), ...;c), ...\}$;

A reprezentacios fuggveny:

$\varphi: C \rightarrow a=A$,

$f\_0=\varphi(g\_0)$,

$(\forall\ f\_c \in F\_c)(\forall\ c \in C)((f\_c(\varphi(c))=\varphi(g\_c(c))$.

Altalanos formaban az azonos jelentes:
$e_{a_i}(\ldots, f_j(\varphi(c)), \ldots;\varphi(c)) = e_{c_i}(\ldots, g_j(c), \ldots;c)$.

A helyesseg definiciojat szimulacio alapjan definialtuk:
$(\forall\ f_s \in F_s)(\forall\ c \in C)(f_s(\varphi(c)) = \varphi(g_s(c)))$.

Az azonos jelentes:
- "Algebrai - algebrai" eset:
$f_s(f_c(\varphi(c)) = f_c(f_s(\varphi(c))) \equiv g_s(g_c(c)) = g_c(g_s(c))$.

- "Kulso felulet - kulso felulet" eset:
$(pref_i(\varphi(c)) \wedge postf_i(\varphi(c),\varphi(c'))) \equiv (preg_i(c) \wedge postg_i(c,c'))$.