Formális helyességbizonyítás II.

Majzik István
Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék
http://www.mit.bme.hu/~majzik/

Helyességbizonyítás strukturált programokra

- Cél a „komponálhatóság”:
  - Ha egy P program P_1 és P_2 szintaktikai egységekből áll, akkor P tulajdonságai P_1 és P_2 tulajdonságai alapján bizonyíthatók
  - Strukturális indukció elve
- Strukturált programok: PLW nyelv

P::= x:=e | skip | P_1; P_2 | if B then P_1 else P_2 fi | while B do P od

- Példa:

P_{div}: r:=x; q:=0; while r≥y do r:=r-y; q:=q+1 od

PLW szemantikája

- Konfiguráció: $C=(P, \sigma)$ ahol
  - P a szintaktikus folytatás (E az üres folytatás jelölése)
  - σ a látható állapot (állapotváltozók)
- Átmenet reláció: $C \rightarrow C'$
  - $(x:=e, \sigma) \rightarrow (E, \sigma[e/x])$
  - $(skip, \sigma) \rightarrow (E, \sigma)$
  - $(P_1; P_2, \sigma) \rightarrow (P_1'; P_2, \sigma')$ ha $(P_1, \sigma) \rightarrow (P_1', \sigma')$
  - $(\text{if } B \text{ then } P_1 \text{ else } P_2 \text{ fi}, \sigma) \rightarrow (P_1, \sigma)$ ha $\sigma[B]=true$
    $\rightarrow (P_2, \sigma)$ ha $\sigma[B]=false$
  - $(\text{while } B \text{ do } P \text{ od}, \sigma) \rightarrow (P; \text{while } B \text{ do } P \text{ od}, \sigma)$ ha $\sigma[B]=true$
    $\rightarrow (E, \sigma)$ ha $\sigma[B]=false$

Itt az $E;P \equiv P$ azonosság alkalmazható a végén

H dedukciós rendszer részleges helyesség bizonyításához (1)

- Axiómák:
  - ASS: $\{p[e/x]\}\ x:=e\ \{p\}$
  - SKIP: $\{p\}\ \text{skip}\ \{p\}$

Utófeltételként p teljesül, ha előfeltételként p[e/x] teljesül

- Szabályok a szintaktikai struktúrákhoz:
  - SEQ: $$\frac{\{p\}\ P_1\ \{r\} \quad \text{és} \quad \{r\}\ P_2\ \{q\}}{\{p\}\ P_1; P_2\ \{q\}}$$
  - COND: $$\frac{\{p \wedge B\}\ P_1\ \{q\} \quad \text{és} \quad \{p \wedge \neg B\}\ P_2\ \{q\}}{\{p\}\ \text{if B then } P_1 \text{ else } P_2 \text{ fi}\ \{q\}}$$
  - REP: $$\frac{\{p \wedge B\}\ P\ \{p\}}{\{p\}\ \text{while B do P od}\ \{p \wedge \neg B\}}$$

Ha (feltétel) / akkor (köv.)

p ciklus invariáns

H dedukciós rendszer részleges helyesség bizonyításához (2)

- Általános szabályok:
  - CONS: $$\frac{p \Rightarrow p_1 \text{ és } \{p_1\}\ P_1\ \{q_1\} \text{ és } q_1 \Rightarrow q}{\{p\}\ P\ \{q\}}$$
  - AND: $$\frac{\{p\}\ P\ \{q_1\} \text{ és } \{p\}\ P\ \{q_2\}}{\{p\}\ P\ \{q_1 \wedge q_2\}}$$
  - OR: $$\frac{\{p_1\}\ P\ \{q\} \text{ és } \{p_2\}\ P\ \{q\}}{\{p_1 \vee p_2\}\ P\ \{q\}}$$

Előfeltétel bővítés és utófeltétel szűkítés

Utófeltétel részekre bontása

Előfeltétel szétválasztása esetekre

- Domén axiómái és szabályai:
  - A tételbizonyítónak ismernie kell
  - {true}skip{p} alakú állítások is

Részleges helyességbizonyítás példa ☺

{x≥0∧y≥0} r:=x; q:=0; while r≥y do r:=r-y; q:=q+1 od {x=q·y+r ∧ 0≤r<y}

1. $\{x = 0 \cdot y + x \wedge x \geq 0\}\ q := 0\ \{x = q \cdot y + x \wedge x \geq 0\}$ (ASS)
2. $\{x = q \cdot y + x \wedge x \geq 0\}\ r := x\ \{x = q \cdot y + r \wedge r \geq 0\}$ (ASS)
3. $\{x = 0 \cdot y + x \wedge x \geq 0\}\ q := 0; r := x\ \{x = q \cdot y + r \wedge r \geq 0\}$ (1)(2)(SEQ)
4. $x \geq 0 \wedge y \geq 0 \Rightarrow x = 0 \cdot y + x \wedge x \geq 0$ (ARITHMETIC)
5. $\{x \geq 0 \wedge y \geq 0\}\ q := 0; r := x\ \{x = q \cdot y + r \wedge r \geq 0\}$ (3)(4)(CONS)
6. $\{x = (q+1) \cdot y + r - y \wedge r - y \geq 0\}\ r := r - y\ \{x = (q+1) \cdot y + r \wedge r \geq 0\}$ (ASS)
7. $\{x = (q+1) \cdot y + r \wedge r \geq 0\}\ q := q + 1\ \{x = q \cdot y + r \wedge r \geq 0\}$ (ASS)
8. $\{x = (q+1) \cdot y + r - y \wedge r - y \geq 0\}\ r := r - y; q := q + 1\ \{x = q \cdot y + r \wedge r \geq 0\}$ (6)(7)(SEQ)
9. $x := q \cdot y + r \wedge r \geq 0 \wedge r \geq y \Rightarrow x = (q+1) \cdot y + r - y \wedge r - y \geq 0$ (ARITHMETIC)
10. $\{x = q \cdot y + r \wedge r \geq 0 \wedge r \geq y\}\ r := r - y; q := q + 1\ \{x = q \cdot y + r \wedge r \geq 0\}$ (8)(9)(CONS)
11. $\{x = q \cdot y + r \wedge r \geq 0\}$ while $r \geq y$ do $r := r - y; q := q + 1$ od $\{x = q \cdot y + r \wedge 0 \leq r < y\}$ (10)(REP)
12. $\{x \geq 0 \wedge y \geq 0\}\ q := 0; r := x;$ while $r \geq y$ do $r := r - y; q := q + 1$ od $\{x = q \cdot y + r \wedge 0 \leq r < y\}$ (5)(11)(SEQ)

Dedukciós rendszer részleges helyesség bizonyításához (3)

- Egy C állítás bizonyítása: $Tr_I \vdash_H C$ ahol
  - I a domén, Tr_I a domén axiómái és szabályai
  - H a dedukciós rendszer
- Példa az állításra:
  - $\{x \geq 0 \wedge y \geq 0\}\ P_{div}\ \{x = q \cdot y + r \wedge 0 \leq r < y\}$
- Gyakorlati problémák:
  - Domén szabályait a tételbizonyítónak ismernie kell
  - Szabályok alkalmazásának stratégiája (keresés)
- Jellemzők:
  - Az előbb felvázolt H helyessége bizonyítható
    - $Tr_I \vdash_H \{p\}P\{q\}$ következménye $\models_I \{p\}P\{q\}$
  - H teljessége:
    - Ha a domén axióma- és szabályrendszere kellően bonyolult (elegendően erős): Gödel első nemteljességi tétele érvényes, azaz lehet olyan igaz állítás, ami nem bizonyítható
  - Specifikációs nyelv kifejezőképessége elegendő-e?

Gödel tétele

Tétel – Gödel első nemteljességi tétele

- Minden ellentmondásmentes, a természetes számok elméletét tartalmazó, formális-axiomatikus elméletben megfogalmazható olyan állítás, mely se nem bizonyítható, se nem cáfolható.
- Másképp: Minden elegendően erős, ellentmondásmentes elméletben van olyan állítás, mely eldönthetetlen, miközben igaz.

Megjegyzések:

- Formális-axiomatikus elmélet alatt bármilyen formalizált (például elsőrendű nyelvre épített) axiomatikus-deduktív elméletet érthetünk, melynek axiómarendszere rekurzívan felsorolható.
- Ellentmondásos egy axiomatikus-deduktív elmélet, ha van benne olyan mondat, mely bizonyítható is és cáfolható is.
- Azon, hogy tartalmazza a természetes számok elméletét azt értjük, hogy szerepeljenek a formális nyelvben olyan kifejezések, melyek megfeleltethetők a természetes számoknak, az összeadásnak, a szorzásnak úgy, hogy a Peano-aritmetika axiómái megfogalmazhatók és egyben levezethetők is legyenek az elméletben. Ezt a feltételt még úgy is meg szokták fogalmazni, hogy az elmélet elegendően erős.

Specifikációs nyelv kifejezőképessége

- Definíció:
  - SL specifikációs nyelv kifejező egy PL programnyelv és I domén esetén, ha
    $\forall p \in SL$, $\forall P \in PL$ esetén $post_I(p,P)$ kifejezhető SL-ben

  - Egy D dedukciós rendszer relatív teljes a részleges helyesség bizonyításához,
    ha $\forall SL$, $\forall PL$, $\forall I$ esetén, ahol SL kifejező PL-re és I-re, fennáll: $\models_I \{p\}P\{q\}$ következménye $Tr_I \vdash_D \{p\}P\{q\}$

H* dedukciós rendszer helyesség bizonyításhoz

- Cél: PLW programok helyességének bizonyítása
  - while B do P od ciklusok befejeződése kérdéses
- Ötlet (itt is): Paraméterezett állítások
  - Jól megalapozott halmaz (pl. n természetes szám)
  - pi(x,n) ciklus invariáns választása kell
- Módosuló REP szabály ebben az esetben:
  - REP*:

$$\frac{pi(x,n) \Rightarrow B \text{ és } \langle pi(x,n)\rangle P \langle pi(x,n-1)\rangle \text{ és } pi(x,0) \Rightarrow \neg B}{\langle \exists n: pi(x,n)\rangle \text{ while } B \text{ do } P \text{ od } \langle pi(x,0)\rangle}$$

  - Többi szabály:
    {…} helyett egyszerűen <…> kell

Aritmetikai kiterjesztés

- A módosult REP* szabály miatt:
  - SL-nek támogatnia kell a jól megalapozott halmaz (itt: természetes számok) használatát
- Tipikus eset: Peano aritmetika támogatása
  - Természetes számok és +, *, < műveletek
  - Tételbizonyító erre felkészítve
- Definíciók:
  - SL aritmetikai kiterjesztése SL^+, ha minimális bővítésként a Peano aritmetikát tartalmazza
  - I domén aritmetikai kiterjesztése I^+, ha minimális bővítésként a Peano aritmetika doménjét tartalmazza

Aritmetikai helyesség és teljesség

- Aritmetikai helyesség $p,q \in SL^+$ mellett:
  - Tr_{I+} $\mathbf{I}\text{-}_{\mathbf{D}}$ <p>P<q> következménye $\mathbf{I}=_{\mathbf{I+}}$ <p>P<q>
- Aritmetikai teljesség $p,q \in SL^+$ mellett:
  - $\mathbf{I}=_{\mathbf{I+}}$ <p>P<q> következménye Tr_{I+} $\mathbf{I}\text{-}_{\mathbf{D}}$ <p>P<q>
- Itt: H* aritmetikai helyessége bizonyítható

Összefoglalás

- Alacsony szintű pszeudo-nyelvek (blokk diagram):
  - Részleges helyesség ciklusmentes programokra:
    - Visszalépéses számítási indukció
  - Részleges helyesség ciklust tartalmazó programokra:
    - Induktív állítások
  - Helyesség ciklust tartalmazó programokra:
    - Paraméterezett induktív állítások
- Strukturált programnyelvek (while programok):
  - Részleges helyesség:
    - Dedukciós rendszer (strukturális indukció)
  - Helyesség:
    - Dedukciós rendszer, kifejezőképesség
    - Aritmetikai kiterjesztés, aritmetikai helyesség és teljesség

Program helyességbizonyítás a gyakorlatban

Néhány példa:

- Spec# Programming System: C# kiterjesztés
  - Előfeltételek, utófeltételek megadása (metódusokhoz)
  - Objektum invariánsok (pl. adattartományok)
  - Boogie: Az utófeltételek automatikus ellenőrzése
  - Ha sikertelen az ellenőrzés: Futásidejű verifikáció még bevethető
- JML: Java Modelling Language
  - Előfeltételek, utófeltételek, invariánsok megadása
  - ESC/Java2: JML részhalmazhoz automatikus ellenőrzés
- SPARK: Ada nyelvi részhalmaz
  - Interaktív tételbizonyítóval támogatott ellenőrzés
- B módszer: Speciális modellezési nyelv és megközelítés
  - Tételbizonyítás nagy részben automatikus