TÖL301G Formleg mál og reiknanleiki

Heimadæmi 9

Hjörvar Sigurðsson

Verkefnið var rætt við Arnar Sigurðsson

1. Hetjan deyr ef setningin er sönn, og hún deyr ef setningin er ósönn. Eini möguleikinn fyrir hetjuna til að lifa er því að segja setningu sem risinn getur aldrei ákveðið hvort sé sönn eða ósönn.

   Hetjan getur sagt eftirfarandi:

   „Þessi setning er ósönn."

   Þessi setning er óákvarðanleg, en það þýðir að risinn getur aldrei ákveðið hvort hún sé sönn eða ósönn.

   Í grein 4.2 í bók er fjallað um einmitt slík óákvarðanleg vandamál.

2.
   a. Til þess að þýðandinn geti ákvarðað hvort forritið upphafsstillir allar breytur sínar áður en gildi þeirra eru lesin, þyrfti forritið alltaf að geta lesið allt forritið í heild sinni. Það er vegna þess að þangað til að síðasta lína forritsins hefur verið lesin, þá getur þýðandinn ekki verið viss um að ný breyta muni ekki vera skilgreind seinna / neðar í kóðanum, eða að breyta sem hefur verið skilgreind en ekki upphafsstillt verði ekki notuð áður en hún er upphafsstillt. Tökum dæmi:

      i. Hérna er hægt að lesa allann kóðann og ákvarða í lokin að breytan x var upphafsstillt áður en gildi hennar var lesið:

```
Forrit Fi() {
        breyta x;
        x = 5;
        Prenta(x);
}
```

      ii. Hérna er ekki hægt að ákvarða hvort að breytan x var upphafsstillt áður en gildi hennar var lesið, þar sem að forritið nær ekki að klára að lesa allann kóðann:

```
Forrit Fii() {
        breyta x;
        while(...) {} // Óendanleg lykkja.
        Prenta(x);    // Þýðandi kemst aldrei hingað.
}
```

Við vitum að $HALT_{TM}$, eða vandamálið hvort að tiltekin Turing-vél hætti alltaf keyrslu eða ekki, er óákvarðanlegt sbr. setningu 5.1 í bók.

Þar með getum við sagt eftirfarandi:

i. Gefum okkur að það sé hægt að ákvarða hvort að allar breytur séu upphafsstilltar áður en gildi þeyrra er lesið.
ii. Ef að það (i.) er hægt, þá er hægt að ákvarða $HALT_{TM}$, eða hvort að forritið stöðvi keyrslu eða lendi í óendanlegri lykkju.
iii. Við vitum að $HALT_{TM}$ er ekki ákvarðanlegt sbr. setningu 5.1 í bók, og því leiðir forsenda okkar (i.) til mótsagnar.
iv. Við höfnum því forsendu okkar og segjum að það vandamál, hvort að allar breytur hafi verið upphafsstilltar áður en gildi þeirra var lesið, er óákvarðanlegt.

b. Þýðendur geta notað ýmsar aðferðir til að athuga hvort að breytur hafa verið upphafsstilltar. Til dæmis gæti þýðandi varað við mögulegum óupphafsstilltum breytum í hvert sinn sem að breyta er skilgreind (e. declared) án þess að vera upphafsstillt í sömu línu.
Þýðandinn gæti líka brotið kóðann upp í bita, og athugað hvort að reynt sé að lesa gildi breytu í tilteknum bita áður en að hún er upphafsstillt í bita sem kom á undan – í slíkum tilfellum væri þetta framkvæmt áður en farið er í lykkjur og slíkt sem gæti keyrt endalaust.
Til dæmis gæti þýðandi skoðað eftirfarandi kóða, og gert athugasemd við það að x er ekki augljóslega upphafsstillt án þess að kanna dýpra í lykkjur og slíkt, og varað því við að x sé mögulega óupphafsstillt:

```
Forrit 2b() {
int x;
while () {}      // Lykkja sem mögulega keyrir endalaust. Mögulega er x
                 // upphafsstillt í lykkjunni.
print(x);
}
```

3.

i. Við skulum kalla vandamálið, eða hvort að hægt sé að ákvarða hvort að tiltekið ástand í Turing-vél sé gagnlaust, $GAGNLAUST_{TM}$.
ii. Við gerum ráð fyrir að $GAGNLAUST_{TM}$ sé ákvarðanlegt.
iii. Við gefum okkur Turing-vél R sem að ákvarðar $GAGNLAUST_{TM}$.
iv. Við setjum <M, q_A>, þar sem M er Turing-vél og q_A er samþykktarástand M, sem inntak í R.
M.ö.o., við notum R til þess að ákvarða hvort samþykktarástand M sé gagnslaust eða ekki.
v. Ef að R samþykkir inntakið <M, q_A>, þá þýðir það að samþykktarástand M sé gagnslaust, sem svo þýðir að mál M er tómt þar sem að M samþykkir ekkert inntak.
Ef að R hafnar inntakinu, þá þýðir það að samþykktarástand M sé ekki gagnslaust, sem svo þýðir að mál M er ekki tómt þar sem að M samþykkir einhvern streng.

vi. Næst ímyndum við okkur Turing-vél, E, sem að notar R til þess að ákvarða fyrir hvert samþykktarástand í E hvort ástandið sé gagnlaust eða ekki.
Ef að öll samþykktarástönd í E eru gagnlaus, þá vitum við að mál E er tómt.
vii. Hér með hefur forsenda okkar, eða að hægt sé að ákvarða hvort að tiltekið ástand í Turing-vél sé gagnlaust, leitt til þess að við getum ákvarðað vandamálið E_{TM}, eða hvort að mál tiltekinnar Turing-vélar sé tómt (þ.e.a.s. Turing-vélin samþykkir engann streng).
viii. Við vitum, samkvæmt setningu 5.2, að E_{TM} er ekki ákvarðanlegt. Því hefur forsenda okkar leitt til mótsagnar, og því höfnum við forsendunni og segjum að $GAGNLAUST_{TM}$, eða hvort að tiltekið ástand í Turing-vél sé gagnlaust, er óákvarðanlegt.

4. Það sést strax að til þess að ákvarða að M samþykki **ekki** að minnsta kosti einn samhverfan streng, þá þurfi fyrst að ákvarða að M stöðvi keyrslu.
Dæmi:
Útbúum Turing-vél T sem að útfærir eftirfarandi forrit:

```
Program p(String s) {
While (...) {} // Mögulega óendanlaus lykkja.
If s == „abcabc“ {return 1}
else return -1;
}
```

Til þess að ákvarða að T samþykki að minnsta kosti einn samhverfan streng (t.d. „abcabc“) þá þarf fyrst að ákvarða að T stöðvi alltaf og lendi því aldrei í mögulegri óendanlausri lykkju. Það að ákvarða að Turing-vél / forrit stöðvi keyrslu er $HALT_{TM}$ vadamálið, en við vitum að það er óákvarðanlegt vandamál.

Sönnum að L sé óákvarðanlegt á eftirfarandi vegu:
i. Gefum okkur að L sé ákvarðanlegt.
ii. Gefum okkur Turing-vél T (sjá skilgreiningu hér að ofan).
iii. Til þess að T samþykki að minnsta kosti einn samhverfan streng, þá ákvörðum við fyrst að vélin stöðvi ávallt keyrslu á endanum.
iv. Þar með hefur forsenda okkar, eða að L sé ákvarðanlegt, leitt okkur til þeirrar niðurstöðu að $HALT_{TM}$ sé ákvarðanlegt.
v. Við vitum að $HALT_{TM}$ er óákvarðanlegt, en því höfnum við forsendu okkar og segjum L óákvarðanlegt.

5. A_{01TM} er útgáfa af almenna vandamálinu A_{TM} sem við vitum að er óákvarðanlegt sbr. setningu 4.11 í bók.

Við sýnum að A_{01TM} sé óákvarðanlegt með því að gefa okkur að A_{01TM} sé ákvarðanlegt og leiða til mótsagnar með því að sýna að forsendan leiðir til þeirrar niðurstöðu að A_{TM} sé ákvarðanlegt.

i. Gefum okkur að A_{01TM} sé ákvarðanlegt.
ii. Gefum okkur Turing-vél H sem ákvarðar A_{01TM}.

H tekur sem inntak <T> | T er Turing-vél.
H samþykkir ef $01 \in L(T)$.
H hafnar ef $01 \notin L(T)$.

iii. Við búum til Turing-vél S:
   - S tekur sem inntak <M, w> | M er Turing-vél og w er strengur.
   - S býr til Turing-vél A:
     - A tekur sem inntak s | s er strengur.
     - A samþykkir ef M samþykkir w.
     - A hafnar ef M hafnar w.
   - S samþykkir ef H samþykkir A.
   - S hafnar ef H hafnar A.

iv. Við setjum A sem inntak í H.
H getur gert annað af tvennu:
   a. H samþykkir ef $01 \in L(A)$.
      $01 \in L(A)$ ef $w \in L(M)$ (sbr. skilgreiningu á A).
   b. H hafnar ef $01 \notin L(A)$.
      $01 \notin L(A)$ ef $w \notin L(M)$ (sbr. skilgreiningu á A).

v. Þar með höfum við:
S samþykkir ef H<A> samþykkir; H<A> samþykkir ef M<w> samþykkir; svo S samþykkir ef M<w> samþykkir.
M.ö.o. S samþykkir ef $w \in L(M)$.
S hafnar ef H<A> hafnar; H<A> hafnar ef M<w> hafnar; svo S hafnar ef M<w> hafnar.
M.ö.o. S hafnar ef $w \notin L(M)$.

vi. Þar með ákvarðar S A_{TM}, en við vitum að það er ekki hægt, svo að forsenda okkar hefur leitt til mótsagnar. Við höfnum því forsendunni og segjum A_{01TM} óákvarðanlegt.