Verkefnatími í viku 9

Þið eruð eindregið hvött til að leysa dæmin saman í 2 - 3ja manna hópum. Þannig eru minni lýkur á að sitja lengi fastur í einu dæmi, og það er miklu meira stuð þar að auki. Það þurfa þó allir að skila sinni eigin lausn til kennara fyrir lok tímans. Gætið að merkja lausnir með nafni og HÍ-tölvupóstfangi. Reiknið eins mörg dæmi og þið komist yfir.

Efni vikunnar: Óákvarðanleg verkefni (4.1, 5.1).

1. Þegar búið er að sýna að eitt verkefni sé óákvarðanlegt er hægt að sýna með yfirfærslum (reductions) að önnur verkefni séu óákvarðanleg.

   Í fyrirlestri sýndum við að málið

   $$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ er Turing-vél sem stöðvar á } w\}$$

   sé óákvarðanlegt (undecidable) með yfirfærslu *frá* samþykktarverkefninu (acceptance problem)

   $$A_{TM} = \{\langle M, w\rangle \mid M \text{ er Turing-vél sem samþykkir } w\},$$

   þ.e. hvernig hægt væri að ákvarða A_{TM} (sem *er* óákvarðanlegt sbr. sönnun á setningu 4.11) *ef* $HALT_{TM}$ væri ákvarðanlegt.

   a) Sýnið að $HALT_{TM}$ sé Turing-þekkjanlegt (Turing recognizable) með því að lýsa Turing-vél sem þekkir málið.

   b) Hvers vegna er ekki hægt að nota vélina úr a) til að ákvarða (decide) $HALT_{TM}$?

   c) Notið hornalínusönnun til að sýna að $HALT_{TM}$ sé óákvarðanlegt með því að gera ráð fyrir að til sé Turing-vél M_H sem ákvarðar málið og skilgreinið vél D þannig,

      $D =$"Með inntak $\langle M\rangle$:

      1) Keyra M_H með inntak $\langle M, M\rangle$.

      2) Ef M_H samþykkir þá fara í óendanlega lykkju, annars samþykkja."

      Hvað gerir $D(\langle D\rangle)$? Það er, hvernig bregst D við þegar hún fær lýsingu á sjálfri sér sem inntak? Stöðvar vélin? Stöðvar hún ekki? *Ábending*: Skoðið skilgreiningar á D og M_H.

2. Í þessu dæmi skoðum við stöðvunarverkefnið í samhengi við Python forrit frekar en Turing-vélar. Almenna stöðvunarverkefnið er

   Stöðvar Python forrit P á inntaki x?

   Athugum verkefni sem í fyrstu kann að virðast aðeins einfaldara,

   Stöðvar Python forrit F einhverntímann? (F hefur ekkert inntak).

   Sýnið að það verkefni sé óákvarðanlegt með því að gera ráð fyrir að til sé forrit T sem leysir seinna verkefnið og notið það til að leysa það fyrra. *Ábending*: Þið getið harðkóðað (e. hard code) hluti/forrit/inntök í forritum/föllum sem þið setjið fram.

3. Sýnið að það verkefni að ákvarða hvort tiltekið Python fall skili einhverntíman heiltölunni 301 sé óákvarðanlegt. Sýnið það m.þ.a. gera ráð fyrir að slíkt fall sé til, þ.e. fall á forminu

```
def skilar301(P):
  # Inntak: P : Python fall
  # Úttak: True ef P skilar gildinu 301 en False annars
  # ... (svaka-flókinn-kóði) ...
  return val
```

og sýnið að þá mætti ákvarða stöðvunarverkefnið

Stöðvar Python forrit P á inntaki x?

með því að lýsa falli $f(y)$ sem er þannig að f skilar 301 þá og því aðeins að P stöðvar á inntaki x. *Ábending*: Þið getið harð-kóðað (hard code) hluti í fallinu f.

4. Sýnið að málið

$$NOTEMPTY = \{\langle M \rangle \mid M \text{ er Turing-vél og } L(M) \neq \emptyset\}$$

sé óákvarðanlegt með því að sýna hvernig ákvarða mætti $HALT_{TM}$ ef $NOTEMPTY$ væri ákvarðanlegt. Gerið ráð fyrir að til sé Turing-vél N sem ákvarðar $NOTEMPTY$ og lýsið Turing-vél T sem ákvarðar $HALT_{TM}$ með því að nota N sem undirforrit.

Skoðið vélar í setningum 5.1 og 5.2 í bók. Þið getið notið eftirfarandi uppkast að vélinni T:

- T smíðar Turing-vél M_w og kannar hvort $M_w \in NOTEMPTY$. Lágvísirinn (subscript) w á M táknar að w sé harðkóðað í vélina.
- M_w er útbúin þannig að $\langle M, w \rangle \in HALT$ þ.þ.a.a. $\langle M_w \rangle \in NOTEMPTY$.