Den universella Turing-maskinen

Robert Eklund
Stockholms Universitet
HT 1993

Bakgrund

Eftersom en Turing-maskin (tm) är definierad som en finit mängd av kvadrupler med ett tilldelat initialtillstånd, är det möjligt att numrera all möjliga tm. Vi kan koda alla tillstånd och alfabetiska symboler som sekvenser av 0:or och 1:or. Varje tm kan ges en bestämd plats i ett schema, således:

tm		nummer
tm_i	—	0
tm_1	—	1
tm_2	—	2
.	—	.
.	—	.
.	—	.
.	—	.
tm_n	—	n
= alla tm	—	= N

Detta ger ett ett-till-ett-förhållande med de naturliga talen.

ERGO: Antalet tm $= \aleph_0$

Antalet strängar i A* $= \aleph_0$

Antalet delmängder i A* $= 2^{\aleph_0}$

Detta visar att det finns ett överuppräkneligt antal språk som inte är Turing-acceptabla.

Dessutom kan utdata kodas som en sekvens av 0:or och 1:or. Således:

E(M) Kodning av alla möjliga tm.

E(X) Kodning av alla möjliga insträngar. (OBS! Denna mängd inkluderar givetvis alla möjliga tm.)

TM(U) Den universella tm. Den tar som indata E(M)E(X) och mimar M:s beteende på X.

Om M stannar på X så stannar även TM(U) på E(M)E(X), och utdata blir M:s resultat på X. Om M inte stannar på X, så stannar inte heller TM(U) på E(M)E(X).

ERGO: Språket L = { E(M)EX) | M accepterar X } är Turing-acceptabelt.

$\Sigma = \{ 0,1 \}$

Stopp-problemet

Kan vi nu avgöra om en godtycklig tm och en godtycklig insträng kommer att stanna? Man formulerar problemet i termer av Turing-acceptabla och Turing-avgörbara språk.

$$L = \{ E(M)EX) \mid M \text{ accepterar } X \}$$

… är som tidigare nämnts Turing-acceptabelt. Om M stannar på X så stannar TM(U) på E(M)E(X). Men om nu M inte stannar på X? Kan det avgöras? Vi vill med andra ord veta huruvida

$$L = \{ E(M)EX) \mid M \text{ accepterar } X \}$$

… är Turing-avgörbart.

Detta är inte fallet!

Ponera: Anta att L är Turing-avgörbart av en tm M_L.

Om nu L är avgörbart för alla E(M)E(X), kommer det även att vara avgörbart när X är lika med E(M) självt, dvs när insträngen utgörs av maskinen själv (vilket inte utgör något problem, eftersom både E(M) och E(X) är kodade som 0:or och 1:or).

Om nu L_1 är Turing-avgörbart, måste även dess komplement, L_1', vara Turing-avgörbart. Med andra ord, om språket som accepterar sig självt som insträng är Turing-avgörbart, måste även språket som *inte* accepterar sig självt som insträng att vara Turing-avgörbart. Då finns en tm M* som accepterar L_1' (L_1:s komplement, alltså.)

ERGO: $L_1' = \{ x\{ 0,1 \}^* \mid$ x är inte en tm-kodning, eller så är x en kodning av en tm som inte accepterar sig själv som insträng. $\}$

Q: Finns nu kodningen av M* (dvs den tm som accepterar språken som inte accepterar sig själva) själv i L_1'?

A1: $E(M^*) \in L_1'$

I så fall är E(M*) en av strängarna som accepteras av M* (baserat på antagandet att M* accepterar L_1'). Men eftersom $E(M^*) \in L_1'$ är det kodningen av en tm som inte accepterar sin egen kodning!

$\Rightarrow$ Motsägelse!

A2: $E(M^*) \notin L_1'$

I så fall är E(M*) inte en sträng som accepteras av M* (baserat på antagandet att M* accepterar L_1'). Då är M* en tm som inte accepterar sin egen kodning, och är således en medlem i L_1'.

$\Rightarrow$ Motsägelse!

ERGO: M* finns inte, dvs L är inte Turing-avgörbart.