

Lengoaia bereiztezinak badaude

Turing-en makinak azken batean funtzioak direla esan dugu (adibidez, Turing-en makina bat Haskell-eko funtzio bat bezala ikus dezakegu). Eta badakigu funtzio denak 0 eta 1 sinboloak bakarrik erabiliz kode daitezkeela. Beraz funtzioak $A = \{0, 1\}$ alfabetoaren gaineko hitzak bezala ikus ditzakegu. Esate baterako 000111000011110101 funtzio bat izan daiteke.

$A = \{0, 1\}$ izanda, A^* multzoko hitzen bidez kodetu ditzakegu beraz funtzio denak.

Bestetik, badakigu datu denak ere 0 eta 1 sinboloak erabiliz kode daitezkeela. Beraz, existitzen diren lengoaia denak 2^{A^*} multzoan daude, $A = \{0, 1\}$ izanda.

Turing-en makina bakoitzak 2^{A^*} multzoko lengoaia bat definitu dezake, hau da Turing-en makina batek ezin ditzake bi lengoaia desberdin definitu. Ondorioz, 2^{A^*} multzoko lengoaia bakoitzeko Turing-en makina bat beharko genuke.

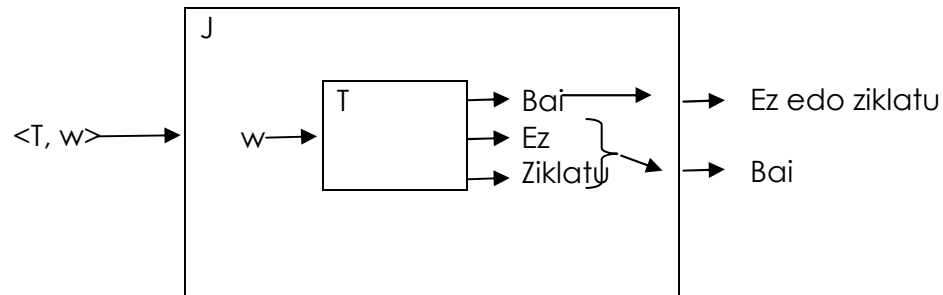
Baina badakigu A^* zenbagarria dela eta 2^{A^*} multzoa ez dela zenbagarria, hau da, 2^{A^*} multzoan A^* multzoan baino elementu gehiago daude. Beraz lengoaia-kopurua makina-kopurua baino handiagoa da eta ondorioz, lengoaia batzuentzat ez dago makinarik. Makinarik ez duten lengoaia horiek lengoaia bereiztezinak dira.

L_{bai} lengoaiaren osagarria $\overline{L_{bai}}$ ez da bereizgarria (bereiztezina da)

$\overline{L_{bai}} = \{ \langle T, w \rangle \mid T \text{ makinak } w \text{ hitzarentzat "Ez" erantzuten du edo ziklatu egiten du} \}$

$\overline{L_{bai}}$ bereizgarria dela frogatzeko, $\overline{L_{bai}}$ lengoaiako hitzentzat, hau da, T makinak w hitzarentzat "Ez" erantzuten du edo ziklatu egiten du baldintza betetzen duten $\langle T, w \rangle$ erako hitzentzat "Bai" erantzuten duen makina bat eraiki beharko genuke (baldintza hori betetzen ez duten hitzentzat "Ez" erantzun lezake edo ziklatu egin lezake).

Kontraesanaren teknika erabiliz $\overline{L_{bai}}$ ez dela bereizgarria frogatuko dugu. Horretarako bereizgarria dela suposatuko dugu eta T makinak w hitzarentzat "Ez" erantzuten du edo ziklatu egiten du baldintza betetzen duten $\langle T, w \rangle$ erako hitzentzat "Bai" erantzuten duen J makina bat existitzen dela suposatuko dugu.



Beraz, J -ri $\langle T, w \rangle$ erako hitz bat ematen diogunean, hau da, T makina edo funtzio baten deskribapena edo definizioa eta T makina horri eman beharreko w datua ematen dizkiogunean, J makinak (edo funtzioak) T makina edo funtzioa exekutatzen du (T makina edo funtzioari deitzen dio) datu bezala w emanez. T makinak "Ez" erantzuten badu w hitzarentzat edo ziklatu egiten badu w hitzarekin, orduan J makinak "Bai" erantzungo du $\langle T, w \rangle$ hitzarentzat (edo datuarentzat). T makinak "Bai" erantzuten badu, berdin zaigu J makinak zer egiten duen $\langle T, w \rangle$ hitzarentzat ("Ez" erantzun edo ziklatu, J makinak L_{bai} lengoaiaren osagarria bereizteko balio behar duelako eta ez dauka L_{bai} lengoaiaren osagarria erabakitzeke balio beharrik).

Bestetik badakigu L_{bai} bereizgarria dela, L_{bai} lengoaiako hitzentzat, hau da, T makinak w hitzarentzat "Bai" erantzuten du baldintza betetzen duten $\langle T, w \rangle$ erako hitzentzat "Bai" erantzuten duen makina bat eraiki dugu lehen.

