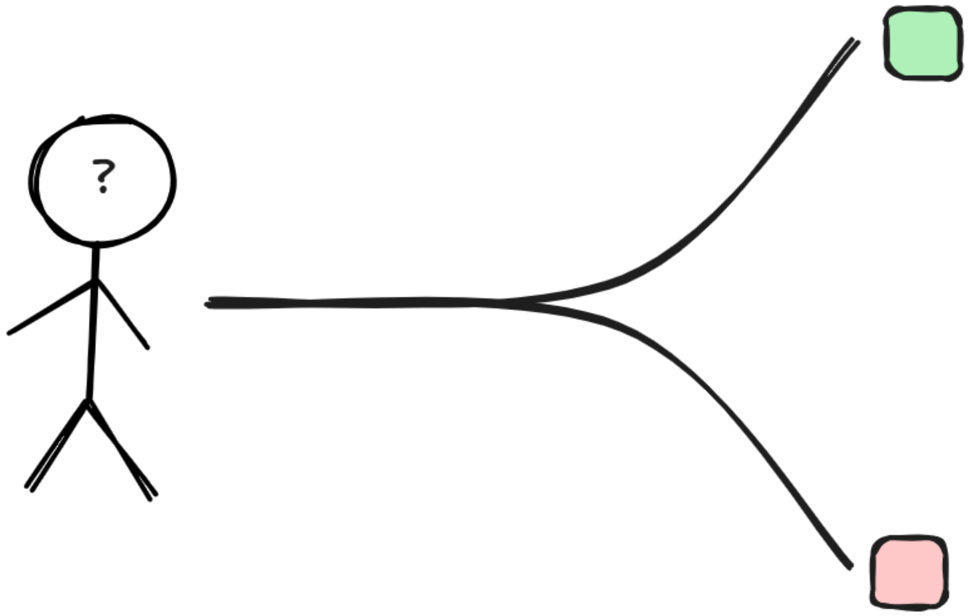
(Ne)odločljivost

Uroš Čibej



Pregled

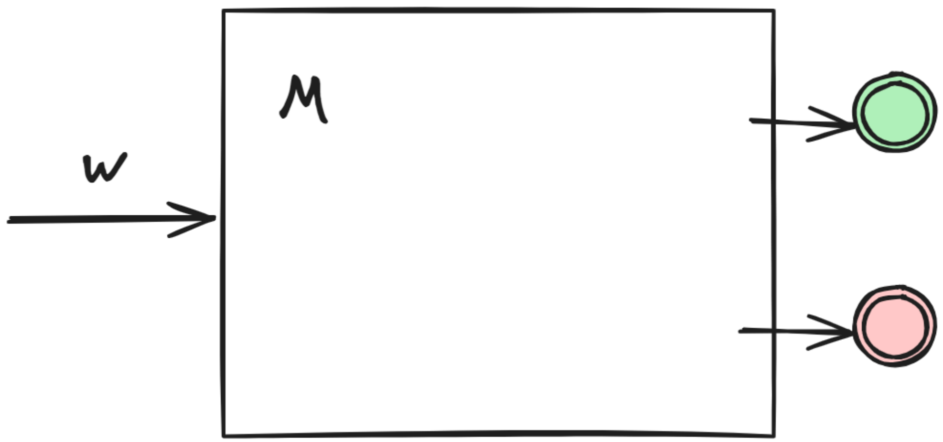
- Church Turingova teza
- Odločljivi problemi
  - O regularnih modelih
  - O kontekstno neodvisnih modelih
- Neodločljivost univerzalnega jezika

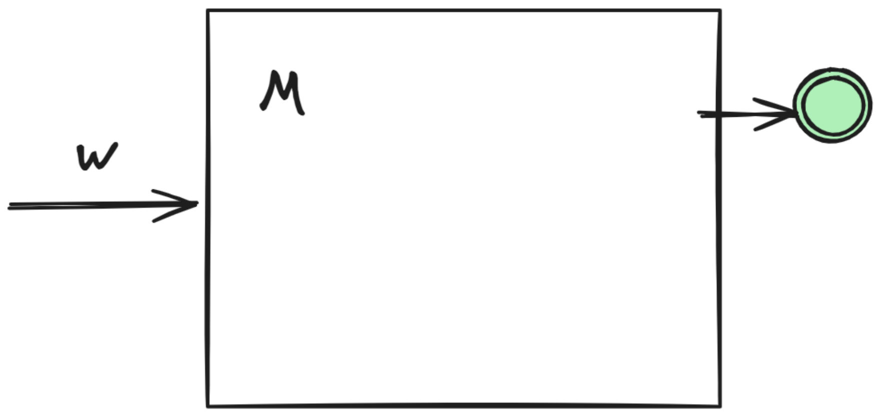
Literatura

- Sipser poglavje 4.

Ponovimo

- Turingov stroj in različni ekvivalentni modeli računanja
- definicija odločljivosti, polodločljivosti


M
w


M
w

Church-Turingova teza

TS $\longleftrightarrow$ Algoritem

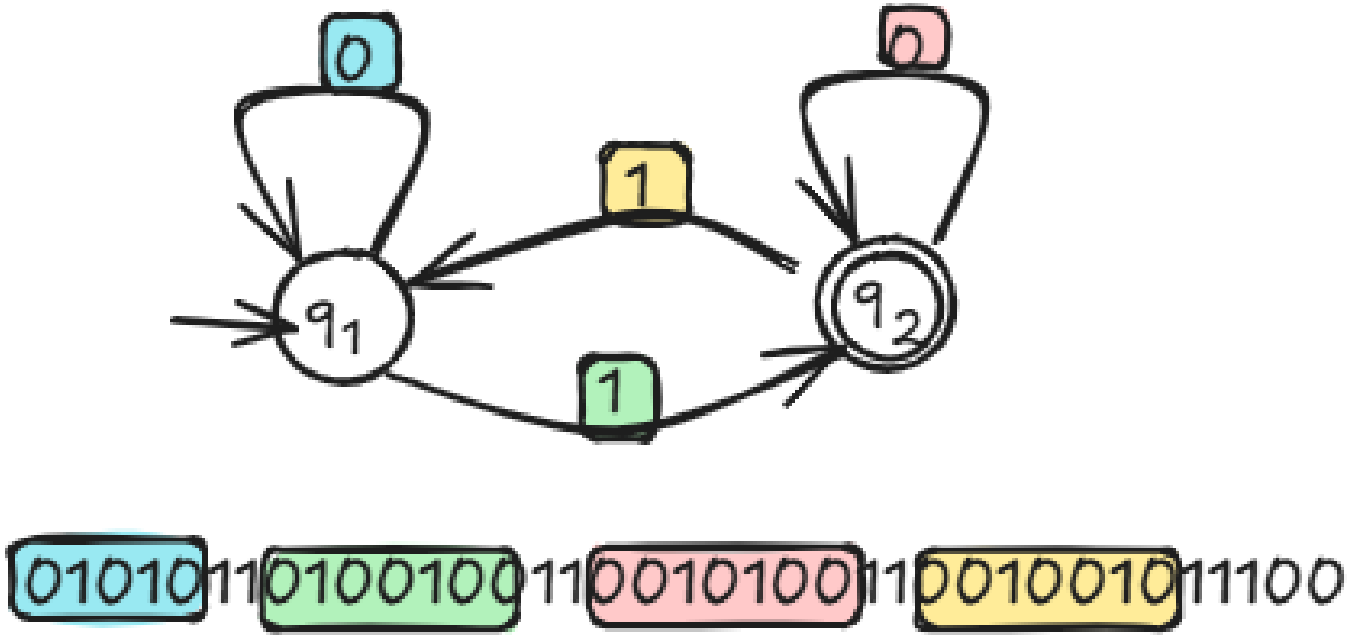
Zakaj zaupamo Church-Turingovi tezi?

- V teh modelih znamo izraziti vse kar intuitivno razumemo kot algoritem
- **OGROMNO** zelo različnih modelov računanja je ekvivalentnih TS
- teoretični rezultati, dobljeni iz teh modelov dobro odražajo praktične situacije

Odločanje o lastnostih končnih avtomatov

- Stroj postane vhod drugemu stroju
- Za katere lastnosti lahko zapišemo algoritem?

Kodiranje



Dve vrsti vprašanj

1. Lastnost samega avtomata (sintaksa)
2. Lastnost jezika tega avtomata (pomen, semantika)

Problem pripadnosti

$$A_{DKA} = \{\langle M, w\rangle \mid M \text{ je DKA, ki sprejme besedo } w\}$$

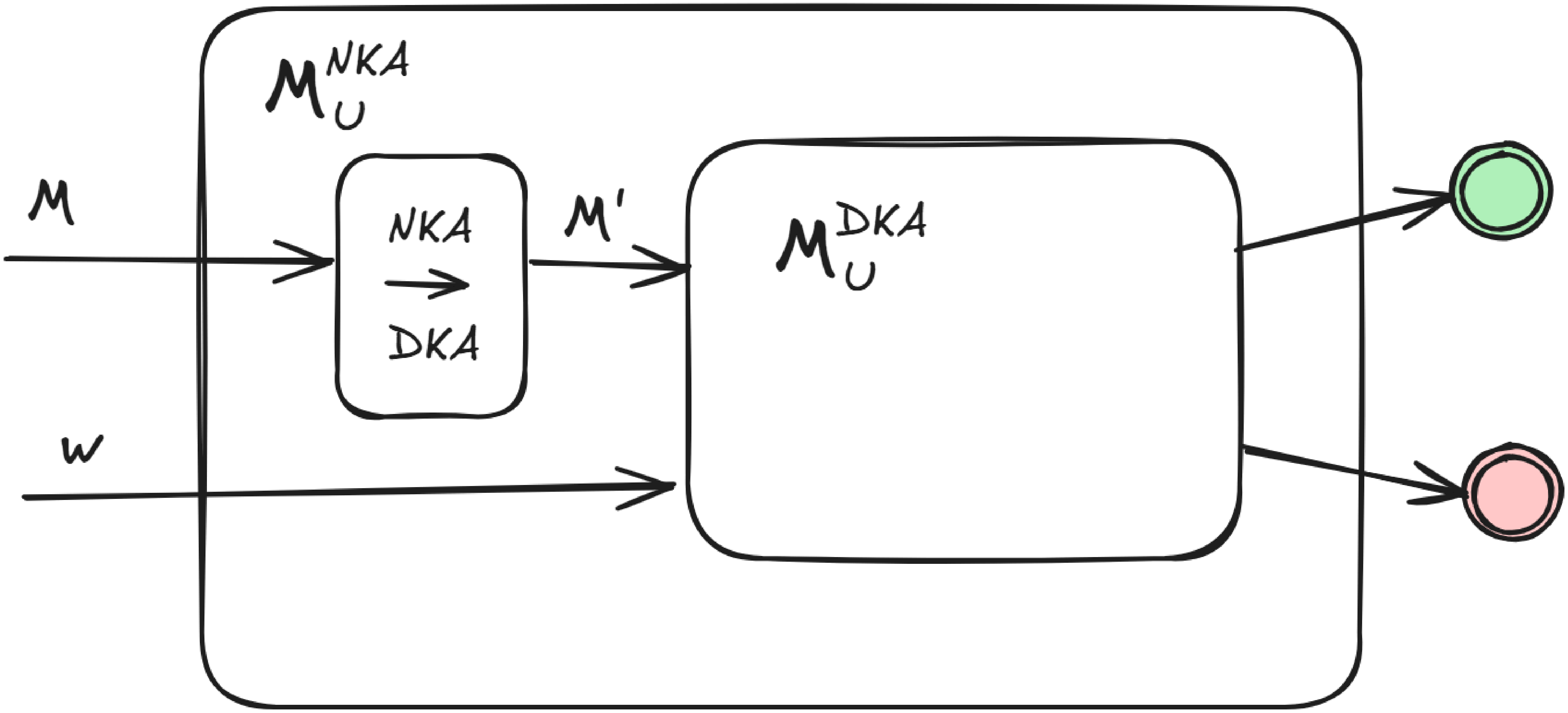
A_{DKA} je odločljiv jezik

3-tračni Turingov stroj, ki simulira poljuben DKA M na vhodu w

1. na drugi trak prepišemo vhod w - premikamo se samo desno
2. dokler na drugem traku ne naletimo na B
   - najdi δ prehod na prvem traku
   - prepiši novo stanje na tretji trak

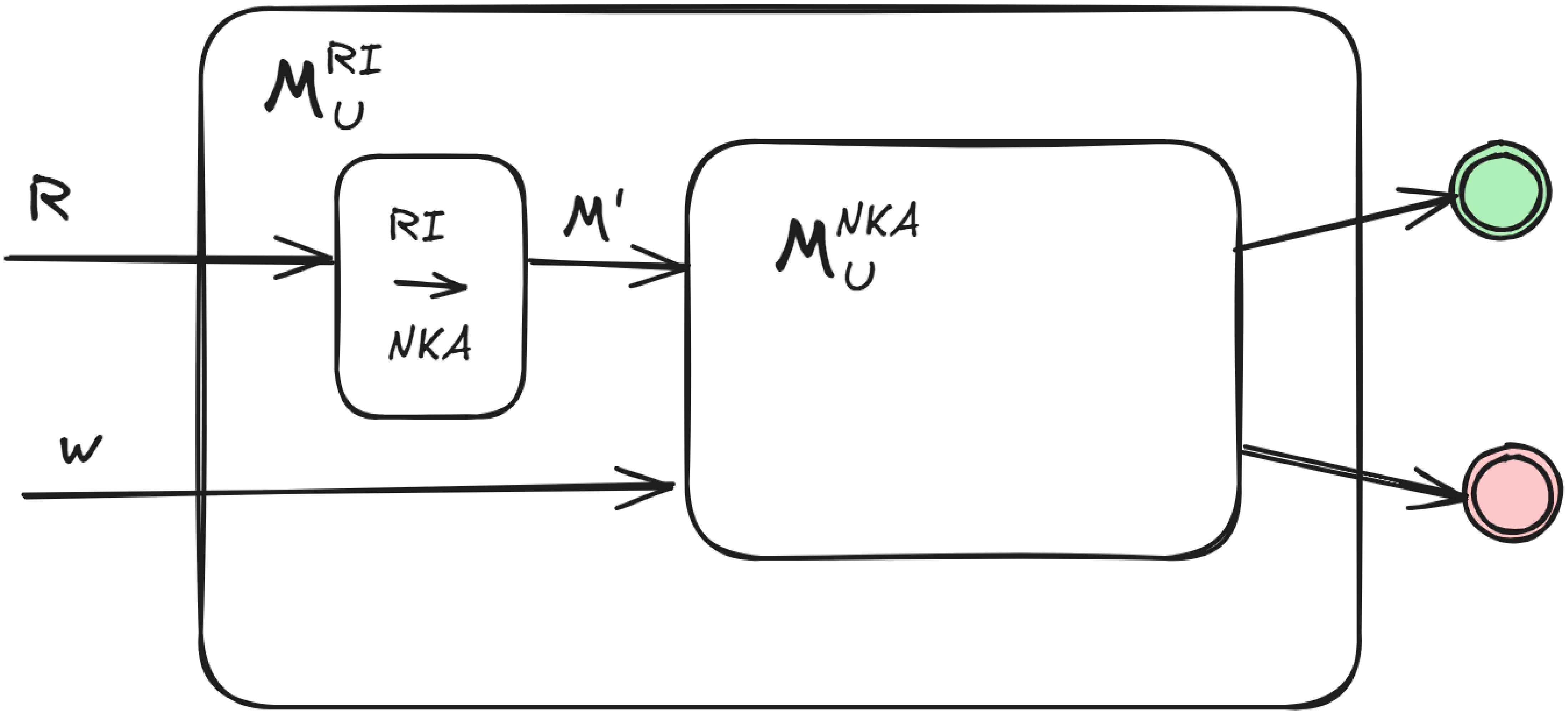
A_{NKA} je odločljiv jezik

$$A_{NKA} = \{\langle M, w\rangle \mid M \text{ je NKA, ki sprejme besedo } w\}$$


M_U^{NKA}
M
NKA → DKA
M'
M_U^{DKA}
w

A_{RI} je odločljiv jezik

$$A_{RI} = \{\langle R, w\rangle \mid R \text{ je reg. izraz, ki opisuje tudi } w\}$$


M_U^{RI}
R
RI
→
NKA
M'
M_U^{NKA}
w

Preverjanje praznosti

$$E_{DKA} = \{\langle M\rangle \mid M \text{ je DKA in } L(M) = \emptyset\}$$

E_{DKA} je regularen jezik

$$L(M) = \emptyset \iff \exists \text{pot v avtomatu } q_0 \to q, q \in F$$

Algoritem za ugotavljanje praznosti

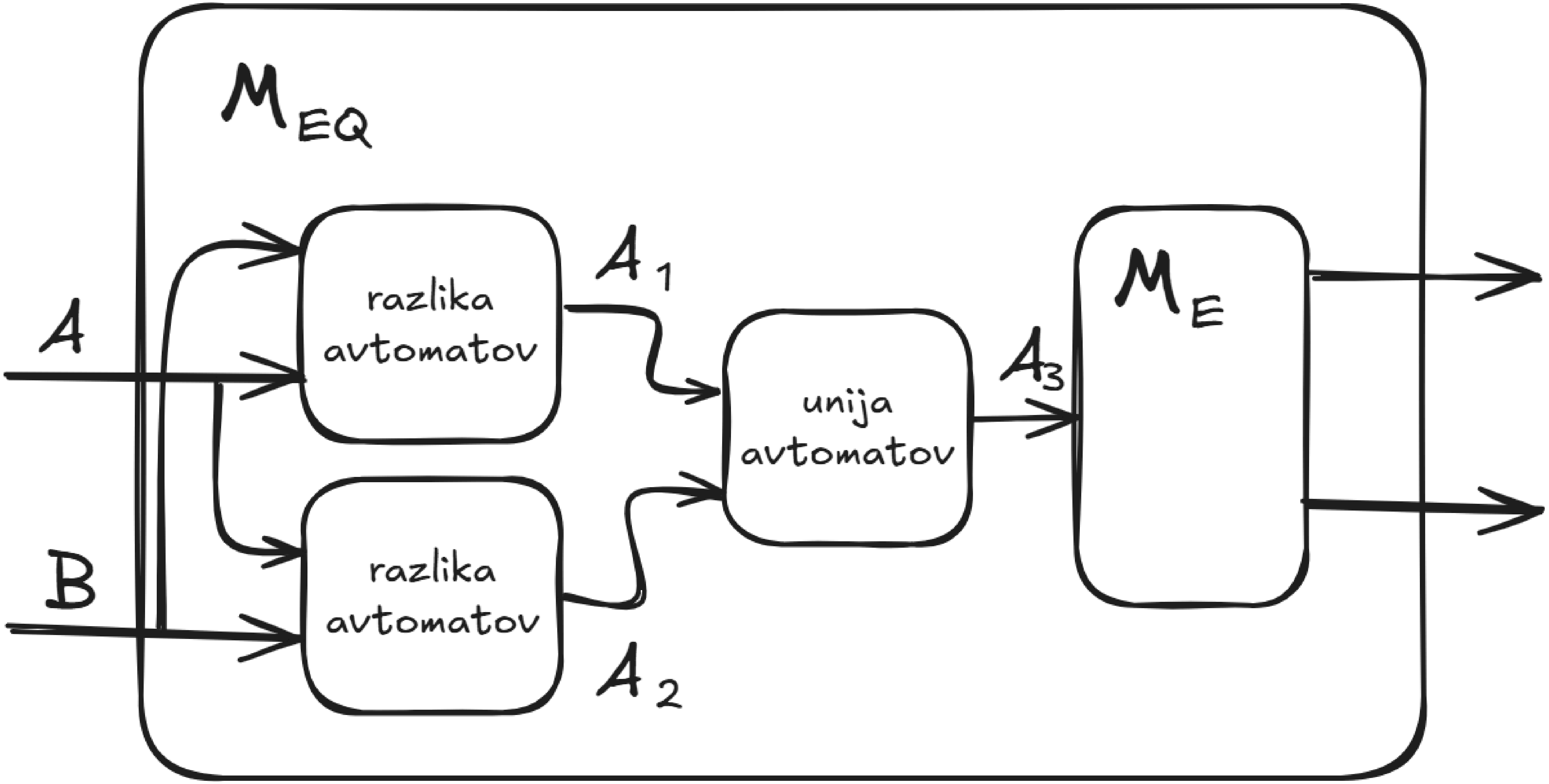
1. označimo q_0
2. dokler se množica označenih stanj spreminja
   - označi vsa neoznačena stanja $r \in Q$, za katere obstaja označeno stanje q in $\delta(q, a) = r$
3. če smo označili neko končno stanje, jezik **ni prazen**, sicer **je prazen**

Preverjanje enakosti

$$EQ_{DKA} = \{\langle A, B \rangle \mid L(A) = L(B), A, B \text{ sta DKA } \}$$

EQ_{DKA} je odločljiv jezik

$$L(A) \setminus L(B) \cup L(B) \setminus L(A) = \emptyset$$



M_{EQ}
A
B
razlika avtomatov
A_1
razlika avtomatov
A_2
unija avtomatov
A_3
M_E

Sorodni problemi za gramatike (SA)

$$A_{KNG} = \{\langle G, w\rangle \mid G \text{ je KNG in } w \in L(G)\}$$
$$E_{KNG} = \{\langle G\rangle \mid G \text{ je KNG in } L(G) = \emptyset\}$$
$$EQ_{KNG} = \{\langle G_1, G_2\rangle \mid G_1, G_2 \text{ sta KNG in } L(G_1) = L_{G_2}\}$$

Problemi nad Turingovimi stroji

1. Lastnost samega stroja
   - preverjanje pravilne sintakse programa
   - transformacija v nek drug jezik (prevajanje)
2. Lastnost jezika tega stroja (pomen, semantika)
   - interpretacija programa
   - ali dva programa počneta isto
   - dokazovanje pravilnosti delovanja

Kodiranje Turingovih strojev

$$\delta(q_i, S_j) = (q_k, S_l, P_m)$$

zakodiramo kot

$$0^i 1 0^j 1 0^k 1 0^l 1 0^m$$

Problem pripadnosti

$$A_{TS} = \{\langle M, w\rangle \mid M \text{ je TS in } w \in L(M)\}$$

- Temu jeziku pravimo tudi univerzalni jezik
- Stroj, ki poskuša to implementirati je splošnonamenski računalnik
- Takemu stroju bomo rekli tudi univerzalni Turingov stroj M_u

Izrek: A_{TS} ni odločljiv jezik

Števna neskončnost

- Σ^* je števno neskončna množica
- TS je števno neskončna množica

Množica neskončnih dvojiških nizov

- $\mathcal{B}$ je množica neskončnih dvojiških nizov
- 2^{Σ^*} je množica vseh jezikov

Izrek. Množici $\mathcal{B}$ in 2^{Σ^*} sta enako močni.

Množica $\mathcal{B}$ ni števno neskončna

Naj bo f poljubna funkcija $\mathrm{N} \to \mathcal{B}$.

Pokažimo, da funkcija f ni surjektivna

$$\overline{d} = d_1 d_2 d_3 d_4$$

kjer je $d_i = \overline{f(i)[i]}$

Primer (slika na desni)

$$\overline{d} = 0100100\ldots$$

N	B							
1	1	0	1	1	0	1	1	0
2	0	0	1	0	1	1	0	1
3	1	1	1	1	0	0	1	0
4	0	0	1	1	0	0	0	0
5	0	1	0	1	0	1	1	0
6	1	0	0	1	1	1	1	1
7	0	1	0	1	1	0	1	0

Množica $\mathcal{B}$ ni števno neskončna

$\overline{d}$ ni slika nobenega elementa N, torej nobena funkcija $f : \mathrm{N} \to \mathcal{B}$ ni bijekcija.

Posledica

Jezikov (problemov) je več kot je algoritmov.

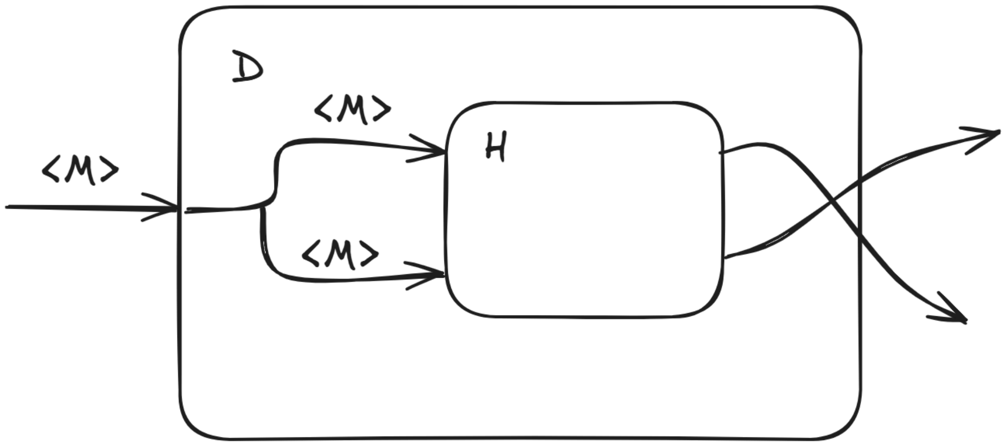
Zelo po domače:

$$|TS| = |\mathrm{N}| < |\mathrm{B}| = |2^{\Sigma^*}|$$

Univerzalni jezik A_{TS} ni odločljiv

1. Naj bo H stroj, za katerega $L(H) = A_{TS}$
2. S pomočjo H skonstruirajmo stroj D

$$D(\langle M \rangle) = \begin{cases} True & M \text{ ne sprejme } \langle M \rangle \\ False & \text{sicer} \end{cases}$$



Univerzalni jezik A_{TS} ni odločljiv

Kakšen rezultat ima $D(\langle D \rangle)$?

$$D(\langle D \rangle) = \begin{cases} True & D \text{ ne sprejme } \langle D \rangle \\ False & \text{sicer} \end{cases}$$

Prišli smo do protislovja $\rightarrow\!\leftarrow$

Stroj D ne obstaja $\implies$ stroj H ne obstaja.

Povezava z diagonalizacijo

	$\langle M_1 \rangle$	$\langle M_1 \rangle$	$\langle M_1 \rangle$	$\langle D \rangle$
M_1	1	0	1	0
M_2	1	0	1	0
M_3	1	0	1	0
D				?