Univerzalita a robustnosť

IB110

Jednoduchý výpočtový model

Hľadáme čo najjednoduchší počítač (výpočtový model), ktorý je schopný realizovať všetky algoritmické výpočty.

Prečo?

- aký najjednoduchší model je schopný realizovať všetky výpočty?
- obecnosť výsledkov o praktickej neriešiteľnosti a nerozhodnuteľnosti
- presná formulácia a formálne dôkazy tvrdení týkajúcich sa praktickej neriešiteľnosti a nerozhodnuteľnosti

Jednoduchý výpočtový model - dáta

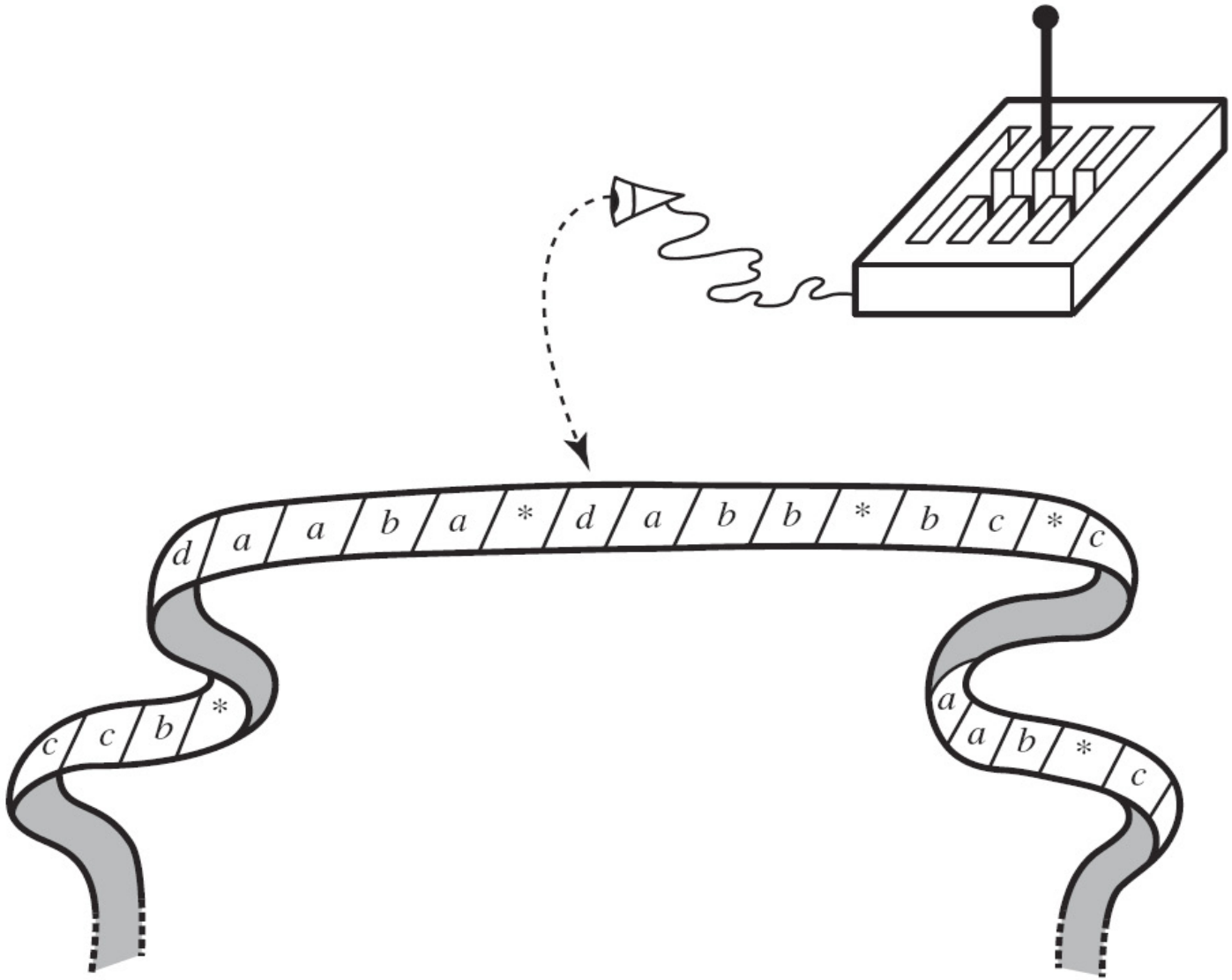
- dáta = reťazce symbolov
- počet rôznych symbolov potrebných na zakódovanie reťazcov je konečný (*podobne ako na zakódovanie všetkých čísel nám stačí v desiatkovej resp. binárnej sústave 10 resp. 2 číslice*)
- dáta môžeme zapisovať na jednorozmernú pásku, ktorá obsahuje políčka; na každom políčku je zapísaný jeden symbol, ktorý je prvkom vstupnej abecedy

Linearizácia dátových štruktúr

- linearizácia zoznamu
- linearizácia dvojrozmerného poľa, matice
- linearizácia stromu

Dynamické dátové štruktúry a neohraničenosť jednosmernej pásky

Jednoduchý výpočtový model - riadiaca jednotka



Jednoduchý výpočtový model - riadiaca jednotka

- výpočet je realizovaný riadiacou jednotkou
- riadiaca jednotka je vždy v jednom z konečne veľa rôznych stavov (*stav zodpovedá inštrukcii algoritmu*)
- riadiaca jednotka vždy snímá práve jedno políčko jednorozmernej pásky
  (*hodnota, s ktorou manipuluje inštrukcia algoritmu*)
- atomické akcie
  - prečítanie symbolu z políčka pásky
  - zápis symbolu na políčko pásky
  - posun o jedno políčko na páske
  - zmena stavu riadiacej jednotky
- závislosť zmeny na aktuálnych hodnotách

Jednoduchý výpočtový model - základné operácie

jeden krok výpočtu

- prečítanie symbolu
- podľa aktuálneho stavu riadiacej jednotky a prečítaného symbolu sa vykoná
  - zmena stavu
  - zápis symbolu
  - posun o jedno políčko vpravo alebo vľavo

Turingov stroj

Alan Turing, 1936

Turingov stroj

Turingov stroj (TS) pozostáva z

- (konečnej) množiny stavov
- (konečnej) abecedy symbolov
- nekonečnej pásky rozdelenej na políčka
- čítacej a zapisovacej hlavy, ktorá sa pohybuje po páske a sníma vždy 1 políčko pásky
- prechodového diagramu

Turingov stroj - prechodový diagram

Prechodový diagram

- orientovaný graf
- vrcholy grafu sú stavy TS
- hrana z vrcholu s do vrcholu t reprezentuje prechod a je označená dvojicou tvaru $\langle a/b, L\rangle$ alebo $\langle a/b, R\rangle$;
  - a je symbol, ktorý hlava TS z pásky číta (tzv. spínač)
  - b je symbol, ktorý na pásku zapisuje
  - L resp. R určuje smer pohyb hlavy doľava resp. doprava
- požadujeme, aby diagram bol jednoznačný (deterministický Turingov stroj), tj. zo stavu nesmú vychádzať dve hrany s rovnakým spínačom
- jeden zo stavov je označený ako štartovný (počiatočný) stav (označený šípkou)
- niektoré zo stavov sú označene ako koncové stavy (označené výrazným ohraničením)

Turingov stroj - výpočet

Krok výpočtu

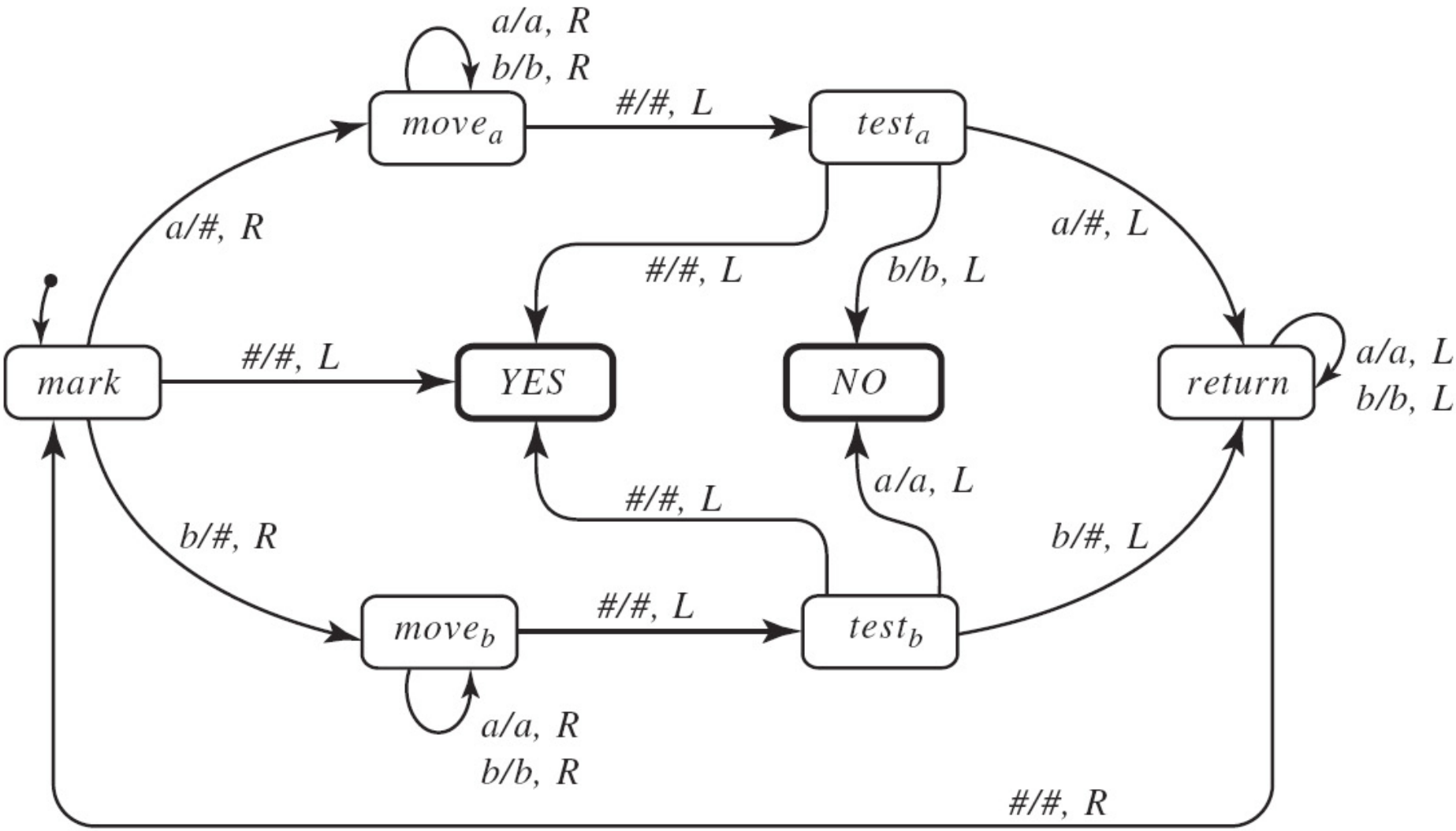
prechod z s do t označený $\langle a/b, L\rangle$ v prechodovom diagrame

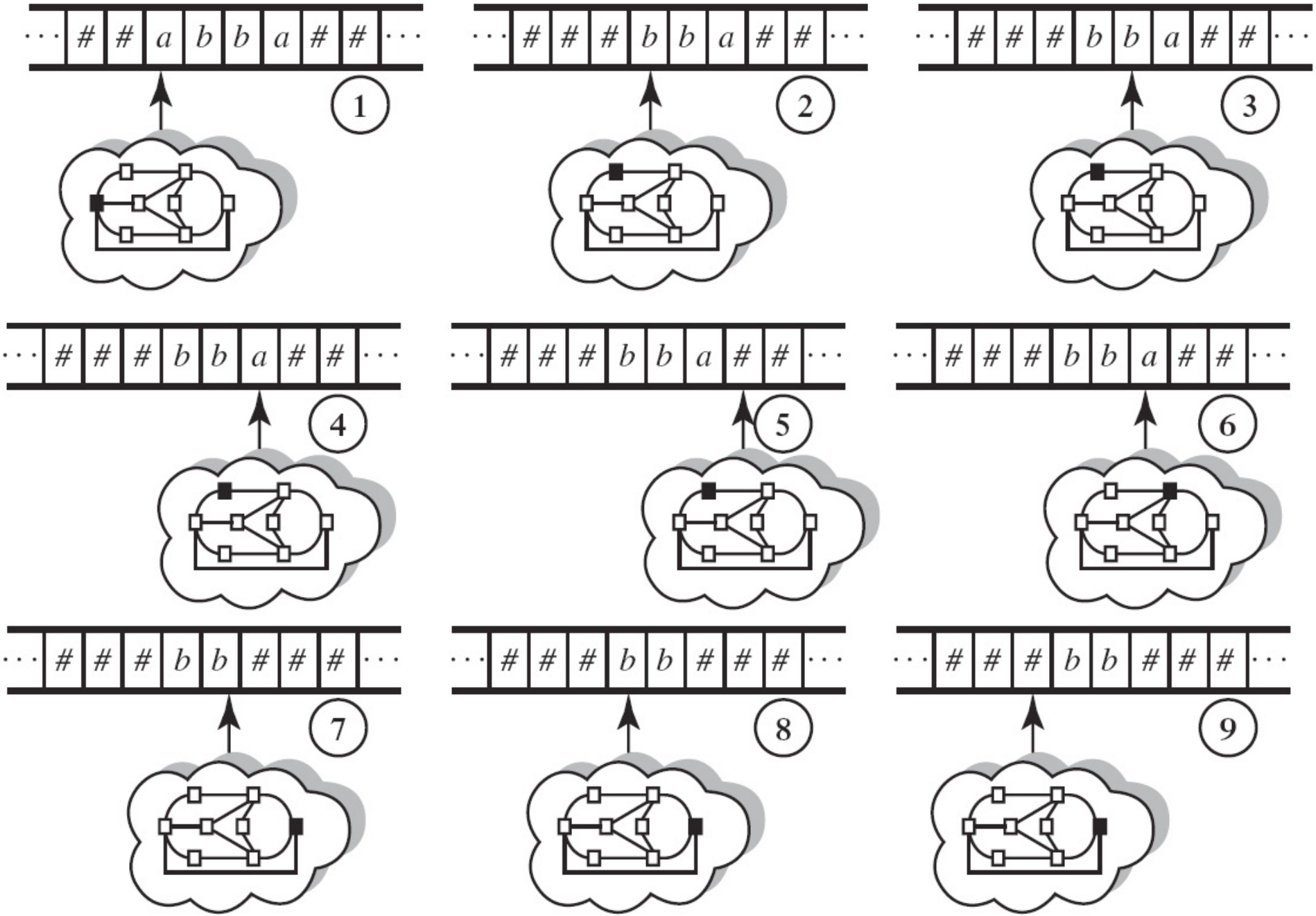
ak riadiaca jednotka TS je v stave s a hlava číta symbol a, tak hlava prepíše symbol a symbolom b, posunie sa o 1 políčko doľava a stav riadiacej jednotky sa zmení na t
(*analogicky pre* $\langle a/b, R\rangle$ *a pohyb vpravo*)

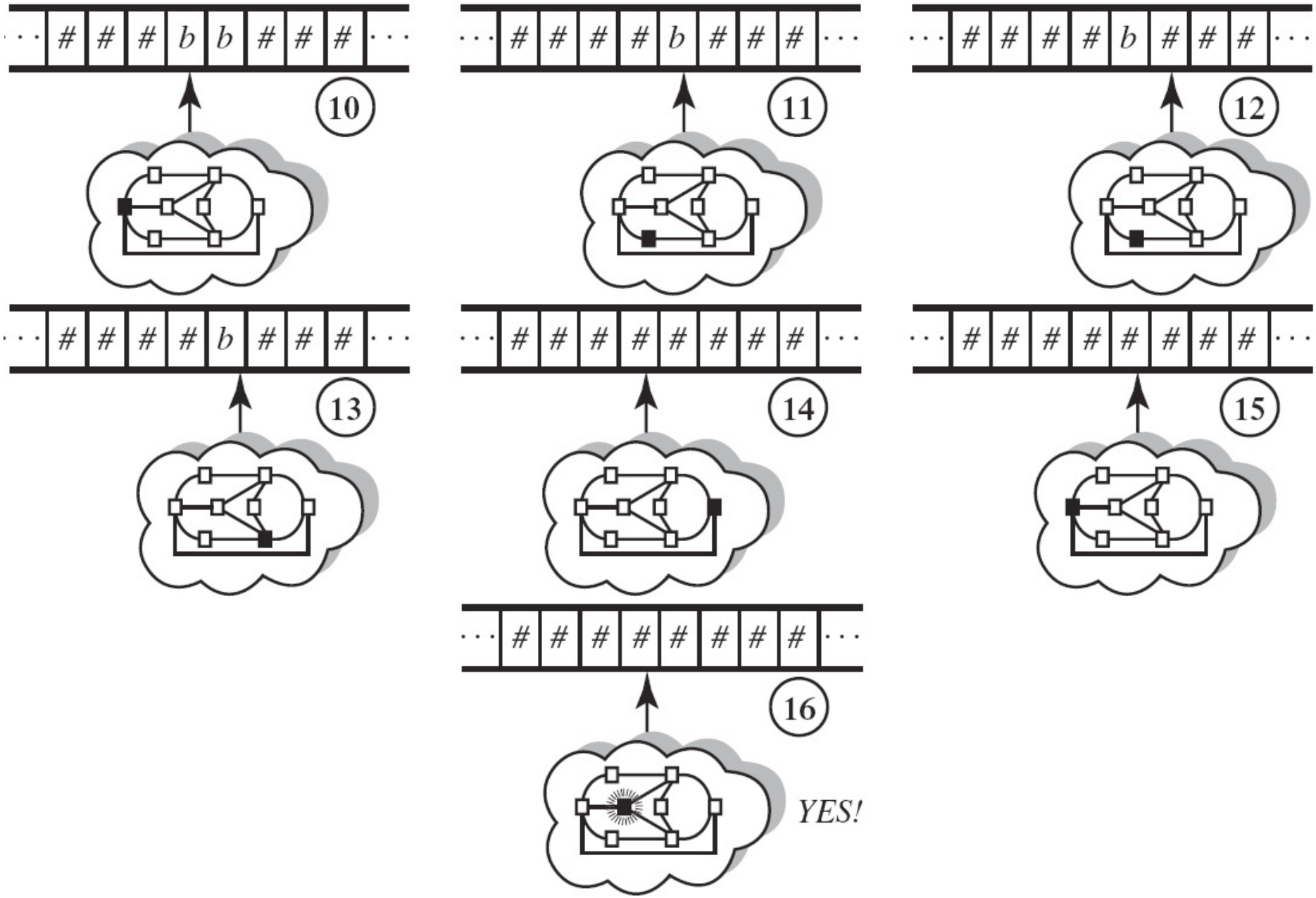
Výpočet

Výpočet začína v počiatočnom stave na najľavejšom neprázdnom políčku pásky.
Výpočet prebieha krok po kroku tak, ako predpisuje prechodový diagram.
Výpočet sa zastaví keď dosiahne niektorý z koncových stavov.

Turingov stroj pre palindrómy




··· # # a b b a # # ···
1
··· # # # b b a # # ···
2
··· # # # b b a # # ···
3
··· # # # b b a # # ···
4
··· # # # b b a # # ···
5
··· # # # b b a # # ···
6
··· # # # b b # # # ···
7
··· # # # b b # # # ···
8
··· # # # b b # # # ···
9


10
11
12
13
14
15
16
YES!

Simulátor Turingových strojov

http://www.fi.muni.cz/ xbarnat/tafj/turing

Turingov stroj ako algoritmus

- Turingov stroj môžeme chápať ako počítač s jedným, fixovaným programom
- softwarom je prechodový diagram; hardwarom je riadiaca jednotka a páska
- jednotlivé TS sa líšia iba svojim softwarom, preto často hovoríme o *programovaní* Turingovho stroja

Turingov stroj ako algoritmus

- Turingov stroj môžeme naprogramovať tak, aby riešil rozhodovací problém
- pre rozhodovací problém P, ktorého množina vstupných inštancií je kódovaná ako množina linearizovaných reťazcov, konštruujeme Turingov stroj M s počiatočným stavom s a dvoma končovými stavmi YES a NO

pre každý vstup X, ak M začne výpočet v stave s na najľavejšom symbole reťazca X, tak M skončí výpočet v stave YES a NO v závislosti na tom, či výstupom P pre X je "Áno" alebo "Nie"

Turingov stroj ako algoritmus

Turingove stroje môžu byť naprogramované aj pre riešenie iných než rozhodovacích problémov.

V takomto prípade predpokladáme, že keď sa TS zastaví (prejde do koncového stavu), tak výstupom je reťazec zapísaný na páske medzi dvoma špeciálnymi znakmi (napr. !)

Ak sa výpočet zastaví a na páske je iný počet symbolov ! ako 2, chápeme výpočet ako neukončený (tj. ako výpočet, ktorý cyklí donekonečna).

Churchova Turingova hypotéza

Aké problémy sú riešiteľné pomocou vhodne naprogramovaného TS?

Churchova Turingova hypotéza

Každý algoritmický problém, pre ktorý existuje program v nejakom programovacom jazyku vyššej úrovne a je riešiteľný na nejakom hardwaru, je riešiteľný aj na Turingovom stroji.

Prečo hypotéza?

CT hypotéza formuluje vzťah medzi dvoma konceptami:

- matematicky presný pojem riešiteľnosti na Turingovom stroji a
- neformálny koncept algoritmickej riešiteľnosti, ktorý je postavený na pojmoch "programovací jazyk vyššej úrovne", "program v programovacom jazyku"

Argumenty pre Churchovovu Turingovu hypotézu

CT hypotézu formulovali v 30-tych rokoch nezávisle Alonso Church a Alan Turing.
Od tej doby bolo navrhnutých množstvo "univerzálnych" modelov (*absolútnych, schopných riešiť všetky mechanicky riešiteľné problémy*)

- Turingove stroje (*Alan Turing*)
- lambda kalkulus (*Alonso Church*)
- produkčné systémy (*Emil Post*)
- rekurzívne funkcie (*Stephen Kleene*)
- kvantové počítače
- ...

Fakt

O všetkých navrhnutých formalizmoch je dokázané, že sú ekvivalentné v tom zmysle, že určujú zhodnú triedu algoritmicky riešiteľných problémov.

Dôsledky Churchovej Turingovej hypotézy

- extrémne výkonné superpočítače nie sú silnejšie než malé počítače s jednoduchým programovacím jazykom; za predpokladu neohraničeného času a veľkosti pamäte dokážu obidva riešiť tie isté algoritmické problémy
- pojem algoritmicky riešiteľného (rozhodnuteľného) problému je robustný, tj. je nezávislý na konkrétnej voľbe výpočtového modelu resp. programovacieho jazyka
- CT hypotéza podporuje správnosť definície nerozhodnuteľných problémov