Model RAM (příklad konkrétního stroje = programu)

```
 1   READ
 2   JZERO 13
 3   STORE 2
 4   LOAD 3
 5   ADD =1
 6   STORE 3
 7   STORE 1
 8   LOAD 2
 9   STORE *8
10   ADD 4
11   STORE 4
12   JUMP 1

13   LOAD 4
14   DIV 3
15   STORE 5
16   LOAD =1

17   STORE 1
18   SUB 3
19   JGTZ 26
20   LOAD *8
21   SUB 5
22   WRITE
23   LOAD 1
24   ADD =1
25   JUMP 17

26   HALT
```

Simulace mezi RAMy a Turingovými stroji

Je vcelku jasné, jak lze jakýkoli Turingův stroj s jednostranně nekonečnou páskou přímočaře simulovat RAMem.

Naopak je to technicky komplikovanější, ale myšlenkově to pro programátory zase není tak náročné – simulaci RAMu vícepáskovým Turingovým strojem ilustruje jedna z animací.

(později: jednotková vs. logaritmická míra složitosti ...)

Nerozhodnutelné problémy

Název:Halting Problem
Vstup: Turingův stroj M, vstupní slovo w
Otázka: Zastaví se (výpočet stroje) M, když začne pracovat na (vstupním) slově w ?

Název:Diagonal Halting Problem
Vstup: Turingův stroj M
Otázka: Zastaví se (výpočet stroje) M, když začne pracovat na (vstupním) slově, které je kódem M ?

Převeditelnost mezi (ANO/NE) problémy

(ANO/NE) problém P_1 je *algoritmicky převeditelný* (stručněji *převeditelný*) na problém P_2, což označujeme $P_1 \rightsquigarrow P_2$,
jestliže existuje algoritmus, který k instanci I_1 problému P_1 sestrojí instanci I_2 problému P_2 tak, že odpověď na otázku pro I_1 v P_1 je stejná jako odpověď na otázku pro I_2 v P_2.

např. DHP je převeditelný na HP (DHP$\rightsquigarrow$HP) ...

Tvrzení:
Když P_1 je nerozhodnutelný a $P_1 \rightsquigarrow P_2$, tak P_2 je nerozhodnutelný.

Částečná rozhodnutelnost, Postova věta

Připomněli jsme *částečnou rozhodnutelnost* problémů. Přitom byla podstatná následující definice, kterou zde formulujeme v řeči „tabulek“ (problém P určuje tabulku T_P, stroj M definuje tabulku T_M):

> *Turingův stroj M částečně rozhoduje problém P (typu* ANO/NE*), jestliže u každého vstupu, pro nějž je v T_P* ANO*, je v tabulce T_M výstup* ANO *a pro každý vstup, pro nějž je v T_P* NE*, je v T_M výstup* NE *nebo znak $\perp$ (nedefinováno).*

(Pro vstupy, kterým problém P přiřazuje odpověď NE, nemusí stroj M svůj výpočet skončit.) O problému řekneme, že je *částečně rozhodnutelný*, jestliže existuje algoritmus (Turingův stroj), který jej částečně rozhoduje.

Speciálně jsme si uvědomili Postovu větu:
Problém P je rozhodnutelný právě tehdy, když P i jeho doplňkový problém $\overline{P}$ jsou částečně rozhodnutelné.
Např. problém HP je částečně rozhodnutelný, ale ne rozhodnutelný, tedy $\overline{\text{HP}}$ (problém “nezastavení” TS) není (ani) částečně rozhodnutelný.

Univerzální Turingův stroj

Algoritmus, kterým jsme prokazovali částečnou rozhodnutelnost problému zastavení (HP) byl tento: na zadaný stroj (program) M a vstup w spusť „interpret“, který provádí činnost (výpočet) M na vstupu w.

Takový interpret je ovšem také program, tedy algoritmus, a šlo by jej proto realizovat (naprogramovat) ve formě konkrétního Turingova stroje U (viz Věta 7.3.).