## 1.2 Počítač s libovolným přístupem – RAM

**1.2.1** Algoritmus jsme nedefinovali, ale existuje několik různých formálních modelů, které algoritmus popisují. Ukážeme si dva z nich — první, počítač s libovolným přístupem, je blíže „klasickému“ počítací, druhý, Turingův stroj, je natolik jednoduchý, že se bude hodit pro různé teoretické úvahy.

**1.2.2 Počítač s libovolným přístupem,** též nazývaný *RAM* se skládá z programové jednotky, aritmetické jednotky, paměti a vstupní a výstupní jednotky.

**1.2.3 Programová jednotka** obsahuje programový register a vlastní program (programový registr ukazuje na instrukci, která má být provedena).

**1.2.4 Aritmetická jednotka** provádí aritmetické operace sčítání, odčítání, násobení a celočíselné dělení.

**1.2.5 Paměť** je rozdělena na paměťové buňky, každá buňka může obsahovat celé číslo. Předpokádáme neomezený počet paměťových buněk a neomezenou velikost čísel uložených v paměťových buňkách. Pořadové číslo paměťové buňky je *adresa* této buňky.

Buňka s adresou 0 je *pracovní registr*, s adresou 1 je *indexový registr*.

**1.2.6 Vstupní jednotka** je tvořena vstupní páskou a hlavou. Vstupní páska je rozdělena na pole (v každém poli může být celé číslo). Hlava snímá v každém okamžiku jedno pole. Po přečtení pole se hlava posune o jedno pole doprava.

**1.2.7 Výstupní jednotka** je tvořena výstupní páskou a hlavou. Obdobně jako v případě vstupní jednotky je páska rozdělena na pole. Výstupní hlava zapíše číslo do pole výstupní pásky a posune se o jedno pole doprava.

**1.2.8 Konfigurace** počítače s libovolným přístupem je přiřazení, které každému poli vstupní i výstupní pásky, každé paměťové buňce a programovému registru přiřazuje celé číslo. *Počáteční konfigurace* je konfigurace, pro kterou existuje přirozené číslo $n$ s následujícími vlastnostmi:

- kromě prvních $n$ vstupních polí obsahují všechna pole, paměťové buňky číslo 0,
- programový registr obsahuje číslo 1
- prvních $n$ polí obsahuje vstup počítače.

**1.2.9 Výpočet** počítače s libovolným přístupem je posloupnost konfigurací, taková, že začíná počáteční konfigurací a každá následující konfigurace je určena programem počítače.

**1.2.10 Program** počítače s libovolným přístupem používá následující příkazy:

- příkazy přesunu: LOAD operand, STORE operand,

- aritmetické příkazy: ADD operand, SUBTRACT operand, MULTIPLY operand, DIVIDE operand,
- vstupní a výstupní příkazy: READ, WRITE,
- příkazy skoku: JUMP návěští, JZERO návěští, JGE návěští,
- příkazy zastavení: STOP, ACCEPT, REJECT.

**1.2.11 Operand** je buď číslo $j$, zapisujeme $= j$, nebo obsah $j$-té paměťové buňky, zapisujeme $j$, nebo obsah paměťové buňky s adresou $i+j$, kde $i$ je obsah indexového registru, zapisujeme $*j$.

**1.2.12 Návěští** je přirozené číslo, které udává pořadové číslo instrukce, která bude prováděna, dojde-li ke skoku.

**1.2.13 Časová složitost.** Řekneme, že program $P$ pro RAM pracuje s časovou složitostí $\mathcal{O}(f(n))$, jestliže pro každý vstup délky $n$ je počet kroků počítače $T(n)$ roven $\mathcal{O}(f(n))$.

**1.2.14 Paměťová složitost.** Řekneme, že program $P$ pro RAM pracuje s pamětí velikosti $m$, jestliže během výpočtu nebyl proveden žádný příkaz, který by měl adresu operandu větší než $m$ a byl proveden příkaz s adresou $m$.

**1.2.15 Poznámka.** Jestliže se na nějakém vstupu program pro RAM nezastaví, není definována ani časová ani paměťová složitost.

**1.2.16 Věta.** Ke každému Turingovu stroji $M$ existuje program $P$ pro RAM takový, že oba mají stejné chování. Navíc, jestliže $M$ potřeboval $n$ kroků, $P$ má časovou složitost $\mathcal{O}(n^2)$.

**1.2.17 Věta.** Pro každý program $P$ pro RAM existuje Turingův stroj $M$ s pěti páskami takový, že $P$ i $M$ mají stejné chování.

**1.2.18 Věta.** Jestliže program $P$ pro RAM splňuje následující podmínky:

- program obsahuje pouze instrukce, které zvětšují délku binárně zapsaného čísla maximálně o jednu;
- program obsahuje pouze instrukce, které Turingův stroj s více páskami provede na slovech délky $k$ v $\mathcal{O}(k^2)$ krocích,

pak Turingův stroj z věty 1.2.17 simuluje $n$ kroků programu $P$ pomocí $\mathcal{O}(n^3)$ svých kroků.

**1.2.19 Důsledek.** Předpokládejme, že program $P$ pro RAM splňuje podmínky z věty 1.2.17. Pak existuje Turingův stroj s jednou páskou, který má stejné chování jako $P$ a $n$ kroků programu $P$ simuluje pomocí $\mathcal{O}(n^6)$ svých kroků.