# Turingův stroj

Ještě než si sdělíme přesnou definici Turingova stroje, povíme si obecněji o jeho historii a významu pro moderní informatiku. Je pojmenován po svém „vynálezci“, britském matematikovi Alanu Turingovi (1912-1954), který je považován za zakladatele umělé inteligence. Kromě svého působení ve skupině kryptoanalytiků, kteří za 2. světové války spolupracovali na odhalení německých metod šifrování tajných zpráv, proslul i vyřčením podmínek tzv. Turingova testu. Jedná se o vcelku objektivní metodu stanovení, zda nějaký systém (např. počítač) je dostatečně inteligentní, aby nahradil člověka.

Vraťmme se zpátky k Turingovu stroji. Je to výpočetní model příslušný třídě rekurzivně spočetných jazyků, tedy jazyků typu 0 v Chomského hierarchii gramatik. Podobá se konečnému automatu, neboť disponuje vstupní páskou a konečně stavovou řídící jednotkou. Páska je však nekonečná, navíc čtecí i zapisovací, a Turingův stroj se po ní může pohybovat oběma směry. Díky tomu se jedná o výkonnější zařízení, jehož pásku si můžeme představit jako operační paměť, kde jednotlivá políčka jsou bloky paměti. Přístup k pásce je však bohužel sekvenční, tj. k přečtení informace na daném políčku potřebujeme projít všechna předcházející políčka.

Z výše uvedeného vyplývá, že model Turingova stroje se již velmi přibližuje dnešnímu počítači, pokud si představíme stavovou jednotku jako procesor a vstupní pásku jako operační paměť. Není to však jediný „abstraktní“ model, který byl matematiky ve 20. století navržen, aby pomocí něj mohli popsat výpočetní sílu dnešních počítačů. Uveďme např. rekurzivní funkce, Lambda kalkul, popř. RAM stroje. Všechny modely však mají jednu věc společnou – dá se dokázat, že mají stejnou výpočetní sílu. Turingův stroj byl však vybrán jako reprezentant této třídy modelů, protože ve své podstatě si jej můžeme nejlépe a nejjednodušeji představit.

Alan Turing společně s Alonzem Churchem však vyslovili ještě jednu zajímavou domněnku, která je vskutku revoluční a činí model Turingova stroje velmi významným. V tzv. Churchově tezi přesně popsali pojem algoritmus.

## Churchova teze

Každý proces, který lze intuitivně nazvat algoritmem, se dá realizovat na Turingově stroji.

## Poznámka.

Důkaz teze nebyl nikdy podán a pravděpodobně ani nebude. Důvodem je abstrakce slova algoritmus. Tento pojem není přesně definován, chápeme jej pouze intuitivně, proto s ním nelze operovat v matematickém důkazu.

## Definice – Turingův stroj

Turingův stroj (TM) je $M = (Q, \Sigma, \Gamma, \triangleright, \sqcup, \delta, q_0, q_{acc}, q_{rej})$, kde

- $Q$ je konečná množina stavů,
- $\Sigma$ je konečná množina vstupních symbolů (vstupní abeceda),
- $\Gamma$ je konečná množina pracovních symbolů (pracovní abeceda) – platí vztah $\Sigma \subseteq \Gamma$,
- $\triangleright \in \Gamma \setminus \Sigma$ je levá koncová značka,
- $\sqcup \in \Gamma \setminus \Sigma$ je symbol označující prázdné políčko,
- $\delta : \ (Q \setminus \{q_{acc}, q_{rej}\}) \times \Gamma \ \rightarrow \ Q \times \Gamma \times \{L, R\}$ je totální přechodová funkce,

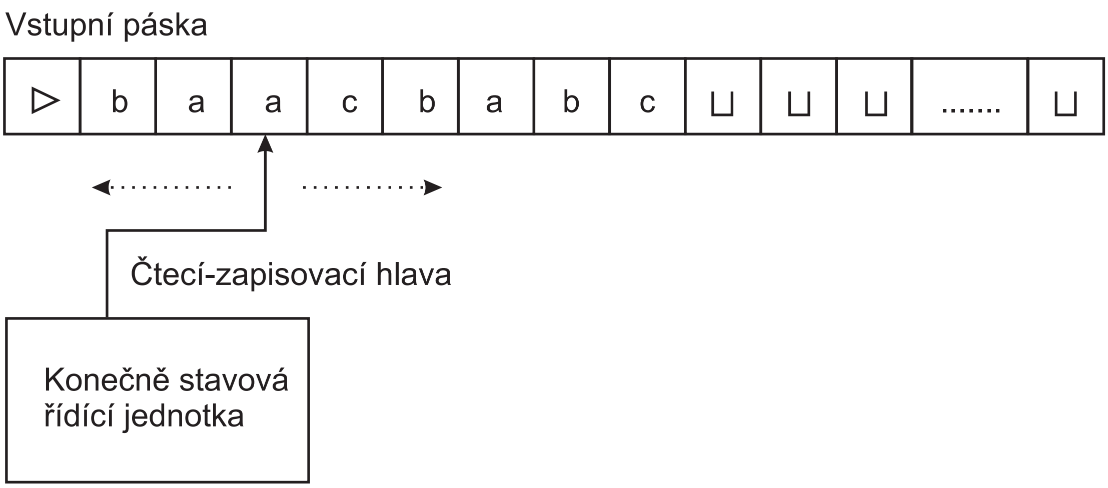
- $q_0 \in Q$ je počáteční stav,
- $q_{acc} \in Q$ je akceptující stav,
- $q_{rej} \in Q$ je zamítající stav.

### Poznámka.

Vstupní páska je jednosměrně nekonečná, tj. požadujeme, aby se čtecí-zapisovací hlava na levé koncové značce $\rhd$ mohla posunout pouze doprava a nepřepisovat tento symbol. Formálně: povolujeme pro libovolný stav $q \in Q$ pouze přechod $\delta(q, \rhd) = (p, \rhd, R)$, kde $p \in Q$ je též libovolný.

## Výpočet Turingova stroje

Představme si Turingův stroj na následujícím obrázku:




Mějme např. slovo $w = baacbabc$. Vstupní páska obsahuje symbol $\rhd$, který znamená začátek vstupní pásky. Poté zapíšeme dané slovo $w$. Zbytek vstupní pásky je prázdný, obsahuje tedy symboly $\sqcup$ (= prázdná políčka).

Principem fungování Turingova stroje je, že čtecí-zapisovací hlava se může pohybovat libovolným směrem, buď doleva nebo doprava. Vždy se posouvá na vstupní pásce o jedno políčko. Je požadováno, aby se hlava nedostala za levý okraj vstupní pásky, tj. aby v případě, že čte symbol $\rhd$, je možný posun pouze doprava. Navíc jsme v předchozí poznámce zmínili, že se levá koncová značka $\rhd$ nemůže přepsat.

### Konfigurace

**Označení:** Symbolem $\sqcup^\omega$ rozumíme libovolný (ne nutně konečný) počet prázdných políček, tedy

$$\sqcup^\omega = \sqcup \sqcup \sqcup \cdots \sqcup$$

### Definice – konfigurace

Konfigurací Turingova stroje $M = (Q, \Sigma, \Gamma, \rhd, \sqcup, \delta, q_0, q_{acc}, q_{rej})$ rozumíme libovolnou trojici $(q, z, n)$ z množiny $Q \times \{y\sqcup^\omega \mid y \in \Gamma^*\} \times \mathbb{N}_0$, kde

- prvek $q$ je stav z množiny $Q$,
- prvek $z$ je aktuální obsah vstupní pásky,
- prvek $n$ je pozice hlavy na pásce, 0 označuje pozici $\rhd$.

Je-li $w$ vstupní slovo pro $M$, pak

- **Počáteční konfigurací** rozumíme trojici $(q_0, \rhd w \sqcup^\omega, 0)$,
- **Akceptující konfigurací** rozumíme trojici $(q_{\text{acc}}, z, n)$,
- **Zamítající konfigurací** rozumíme trojici $(q_{\text{rej}}, z, n)$.

## Krok výpočtu

**Označení:** Je-li $z \in \Gamma$ řetěz objevující se na vstupní pásce, pak definujeme $z_n$ jako jeho $n$-tý symbol zleva. Tedy např. $\rhd$ je roven $z_0$.

Dále, $s^n_X(z)$ označuje řetězec $z'$ vzniklý ze řetězce $z$ tak, že $n$-tý symbol $z_n$ nahradíme za $X$.

## Definice – krok výpočtu

Na množině všech konfigurací Turingova stroje $M = (Q, \Sigma, \Gamma, \rhd, \sqcup, \delta, q_0, q_{acc}, q_{rej})$ definujeme relaci $\vdash_M$ takto:

1. $(p, z, n) \vdash_M (q, s^n_X(z), n + 1)$, pokud $\delta(p, z_n) = (q, X, R)$,
2. $(p, z, n) \vdash_M (q, s^n_X(z), n - 1)$, pokud $\delta(p, z_n) = (q, X, L)$,

kde $p, q \in Q, z \in \Gamma^*, n \in \mathbb{N}_0, X \in \Gamma$.

## Poznámka.

Výpočet na nějakém slově $w \in \Sigma^*$ je maximální (konečná i nekonečná) posloupnost konfigurací $K_0, K_1, \ldots$, kde $K_0$ je počáteční konfigurace pro $w$ a $K_i \vdash_M K_{i+1}$ pro všechna $i \geq 0$.

## Definice – přijímání či zamítání slov

1. Turingův stroj $M$ akceptuje slovo $w$, právě když výpočet $M$ na $w$ je konečný a jeho poslední konfigurace je akceptující.
2. Turingův stroj $M$ zamítá slovo $w$, právě když výpočet $M$ na $w$ je konečný a jeho poslední konfigurace je zamítající.
3. Turingův stroj $M$ pro slovo $w$ cyklí, právě když výpočet $M$ na $w$ je nekonečný.

Jazyk akceptovaný Turingovým strojem $M$ je množina

$$L(M) = \{w \in \Sigma^* \mid M \text{ akceptuje } w\}.$$

## Příklad 1.

Navrhneme Turingův stroj $M$ akceptující jazyk $L = \{a^n b^n c^n \mid n \in \mathbb{N}_0\}$. Výpočet na libovolném slově bude probíhat ve dvou fázích:

1. nejdříve zkontrolujeme, zda vstupní slovo $w$ je tvaru $a^*b^*c^*$. Poslouží nám k tomu stavy $q_0, q_1, q_2$.
2. Jakmile dojdeme až k prázdným políčkům, přepneme se do stavu $q_3$ a vrátíme se na začátek pásky, k symbolu $\triangleright$. Poté v cyklu opakujeme postup, ve kterém vždy přepíšeme symbolem $X$ jedno $a$, jedno $b$ i jedno $c$. A opět se vrátíme na začátek pásky. K zapamatování toho, jaký symbol máme hledat a přepisovat na $X$, nám poslouží stavy $q_4, q_5, q_6$.

Jestliže se na závěr vrátíme (pomocí stavu $q_3$) na začátek pásky a ve stavu $q_4$ budeme pořád číst pouze symboly $X$, až se dostaneme k prvnímu prázdnému políčku, pak slovo $w$ akceptujeme.

Nechť tedy

$$M = (Q, \{a, b, c\}, \{a, b, c, \triangleright, \sqcup, X\}, \triangleright, \sqcup, \delta, q_0, q_{acc}, q_{rej}),$$

kde $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_{acc}, q_{rej}\}$ a přechodová funkce $\delta$ je dána následující tabulkou:

| stav | $\triangleright$ | $a$ | $b$ | $c$ | $\sqcup$ | $X$ |
|---|---|---|---|---|---|---|
| $q_0$ | $(q_0, \triangleright, R)$ | $(q_0, a, R)$ | $(q_1, b, R)$ | $(q_2, c, R)$ | $(q_3, \sqcup, L)$ | $-$ |
| $q_1$ | $-$ | $(q_{rej}, -, -)$ | $(q_1, b, R)$ | $(q_2, c, R)$ | $(q_3, \sqcup, L)$ | $-$ |
| $q_2$ | $-$ | $(q_{rej}, -, -)$ | $(q_{rej}, -, -)$ | $(q_2, c, R)$ | $(q_3, \sqcup, L)$ | $-$ |
| $q_3$ | $(q_4, \triangleright, R)$ | $(q_3, a, L)$ | $(q_3, b, L)$ | $(q_3, c, L)$ | $(q_3, \sqcup, L)$ | $(q_3, X, L)$ |
| $q_4$ | $-$ | $(q_5, X, R)$ | $(q_{rej}, -, -)$ | $(q_{rej}, -, -)$ | $(q_{acc}, -, -)$ | $(q_4, X, R)$ |
| $q_5$ | $-$ | $(q_5, a, R)$ | $(q_6, X, R)$ | $(q_{rej}, -, -)$ | $(q_{rej}, -, -)$ | $(q_5, X, R)$ |
| $q_6$ | $-$ | $-$ | $(q_6, b, R)$ | $(q_3, X, L)$ | $(q_{rej}, -, -)$ | $(q_6, X, R)$ |

Ještě musíme vysvětlit existenci symbolu – v konfiguracích příslušných stavům $q_{rej}, q_{acc}$. V nich je nám jedno, jaký symbol na pásku umístíme, protože již nepokračujeme ve výpočtu a buď zamítáme nebo přijímáme. Za pomlčkou si tedy představte cokoliv.

Ukážeme si výpočet na slově $aabbcc$. Nejprve začínáme fází kontroly na tvar $a^*b^*c^*$:

$$\begin{aligned}(q_0, \triangleright aabbcc\sqcup^\omega, 0) &\vdash (q_0, \triangleright aabbcc\sqcup^\omega, 1) \vdash (q_0, \triangleright aabbcc\sqcup^\omega, 2) \vdash (q_0, \triangleright aabbcc\sqcup^\omega, 3) \vdash \\ &\vdash (q_1, \triangleright aabbcc\sqcup^\omega, 4) \vdash (q_1, \triangleright aabbcc\sqcup^\omega, 5) \vdash (q_2, \triangleright aabbcc\sqcup^\omega, 6) \vdash \\ &\vdash (q_2, \triangleright aabbcc\sqcup^\omega, 7)\end{aligned}$$

Dostáváme se k prvnímu prázdnému políčku (jsme na pozici 7). V tu chvíli se přepínáme do stavu $q_3$ a bez přepisování se vracíme zpátky k levé koncové značce $\triangleright$.

$$(q_2, \triangleright aabbcc\sqcup^\omega, 7) \vdash (q_3, \triangleright aabbcc\sqcup^\omega, 6) \vdash (q_3, \triangleright aabbcc\sqcup^\omega, 5) \vdash \cdots \vdash (q_3, \triangleright aabbcc\sqcup^\omega, 0)$$

Jsme-li ve stavu $q_3$ a na pozici patřící $\triangleright$, přepínáme se do stavu $q_4$ a začínáme první cyklus přepisování symbolů $a, b, c$ znakem $X$.

$$\begin{aligned}(q_3, \triangleright aabbcc\sqcup^\omega, 0) &\vdash (q_4, \triangleright aabbcc\sqcup^\omega, 1) \vdash (q_5, \triangleright Xabbcc\sqcup^\omega, 2) \vdash (q_5, \triangleright Xabbcc\sqcup^\omega, 3) \vdash \\ &\vdash (q_6, \triangleright XaXbcc\sqcup^\omega, 4) \vdash (q_6, \triangleright XaXbcc\sqcup^\omega, 5) \vdash (q_3, \triangleright XaXbXc\sqcup^\omega, 4)\end{aligned}$$

Jsme ve stavu $q_3$ a opět se vracíme k levé koncové značce $\triangleright$:

$$(q_3, \triangleright XaXbXc\sqcup^\omega, 4) \vdash (q_3, \triangleright XaXbXc\sqcup^\omega, 3) \vdash \cdots \vdash (q_3, \triangleright XaXbXc\sqcup^\omega, 0)$$

Začínáme nový cyklus přepisování symbolů $a, b, c$ znakem $X$:

$$\begin{aligned}(q_3, \rhd XaXbXc\sqcup^\omega, 0) &\vdash (q_4, \rhd XaXbXc\sqcup^\omega, 1) \vdash (q_4, \rhd XaXbXc\sqcup^\omega, 2) \vdash (q_5, \rhd XXXbXc\sqcup^\omega, 3) \vdash \\ &\vdash (q_5, \rhd XXXbXc\sqcup^\omega, 4) \vdash (q_6, \rhd XXXXXc\sqcup^\omega, 5) \vdash (q_6, \rhd XXXXXc\sqcup^\omega, 6) \vdash \\ &\vdash (q_3, \rhd XXXXXX\sqcup^\omega, 5)\end{aligned}$$

Pomocí stavu $q_3$ se opět dostaneme na začátek vstupní pásky:

$$(q_3, \rhd XXXXXX\sqcup^\omega, 5) \vdash (q_3, \rhd XXXXXX\sqcup^\omega, 4) \vdash \cdots \vdash (q_3, \rhd XXXXXX\sqcup^\omega, 0)$$

Na závěr se opět přepneme do stavu $q_4$ a protože máme na vstupu pouze symboly $X$, zůstáváme v něm až do konce.

$$(q_4, \rhd XXXXXX\sqcup^\omega, 0) \vdash (q_4, \rhd XXXXXX\sqcup^\omega, 1) \vdash \cdots \vdash (q_4, \rhd XXXXXX\sqcup^\omega, 7)$$

Až se dostaneme k prvnímu prázdnému políčku, můžeme akceptovat:

$$(q_4, \rhd XXXXXX\sqcup^\omega, 7) \vdash (q_{acc}, \rhd XXXXXX\sqcup^\omega, -)$$

## Rekurzivní a rekurzivně spočetné jazyky

Nyní se vrátíme k Chomského hierarchii gramatik. Uváděli jsme si 4 typy:

1. typ 0 (gramatika bez omezení) – frázová gramatika
2. typ 1 – kontextová gramatika
3. typ 2 – bezkontextová gramatika
4. typ 3 – regulární gramatika

Typy 2, 3 už známe. Typ 1 zde nebudeme podrobně rozebírat, popíšeme však typ 0. Platí následující věta:

### Věta 1 (vztah frázových gramatik a Turingova stroje).

Jazyk $L$ je rekurzivně spočetný (generovaný gramatikou typu 0), právě když existuje Turingův stroj $M$ takový, že $L = L(M)$.

### Poznámka.

Všimněte si tedy, že i pro gramatiky typu 0 máme odpovídající výpočetní model a odpovídající třídu jazyků.

### Definice – úplný Turingův stroj

Turingův stroj $M$ se vstupní abecedou $\Sigma$ se nazývá úplný, právě když pro každé slovo $w \in \Sigma^*$ výpočet $M$ zastaví, tj. je konečný. (Libovolné slovo $w$ je buď akceptováno nebo zamítnuto).

### Definice – rekurzivní jazyk

Jazyk $L$ se nazývá rekurzivní, právě když je akceptován nějakým úplným Turingovým strojem.

### Poznámka.

Pro přehlednost si nyní uvedeme výsledky našeho usilí. V tabulce naleznete gramatiky, třídy jazyků a výpočetní modely, které jsou vzájemně ekvivalentní. Platí, že třída na nižším řádku je vlastní podmnožinou (podtřídou) třídy na vyšším řádku.

| **Jazyky** | **Gramatiky (typ)** | **Automaty** |
|---|---|---|
| rekurzivně spočetné | frázové (0) | Turingovy stroje |
| rekurzivní | – | úplné Turingovy stroje |
| kontextové | kontextové (1) | lineárně ohraničené TM |
| bezkontextové | bezkontextové (2) | zásobníkové automaty |
| deterministické CFL | – | deterministické PDA |
| regulární | regulární (3) | konečné automaty |

## Rozhodnutelné a nerozhodnutelné problémy

Nakonec si zobecníme i naši dávnou otázku, zda nějaký problém lze či nelze rozhodnout. Nejdříve si jej zpřesníme:

Problém, zda daný objekt $O$ má vlastnost $P$, ztotožníme s množinou $\{O \mid O \text{ má vlastnost } P\}$. Např. problém, zda daná regulární gramatika $G$ je konečná, můžeme zapsat:

$$\{G \mid G \text{ je regulární a } \ L(G) \text{ je konečný}\}.$$

### Definice – rozhodnutelnost

Problém, zda objekt $O$ má vlastnost $P$, je

1. rozhodnutelný, právě když množina $\{O \mid O \text{ má vlastnost } P\}$ je rekurzivní.
2. nerozhodnutelný, právě když množina $\{O \mid O \text{ má vlastnost } P\}$ není rekurzivní.
3. částečně rozhodnutelný (semirozhodnutelný), právě když množina $\{O \mid O \text{ má vlastnost } P\}$ je rekurzivně spočetná.