# Cvičenie č. 10
# Turingove stroje

Peter Kostolányi

23. novembra 2022

## 1 Deterministické Turingove stroje

**Definícia 1.** *Deterministický Turingov stroj* je šestica $A = (K, \Sigma, \Gamma, \delta, q_0, F)$, kde $K$ je konečná množina stavov, $\Sigma$ je vstupná abeceda, $\Gamma \supseteq \Sigma$ je pracovná abeceda neobsahujúca špeciálny symbol $\mathbf{B} \notin \Gamma$ („blank“), $\delta\colon K \times (\Gamma \cup \{\mathbf{B}\}) \nrightarrow K \times \Gamma \times \{-1, 0, 1\}$ je *čiastočná* prechodová funkcia, $q_0 \in K$ je počiatočný stav a $F \subseteq K$ je množina akceptačných stavov.

Čiastočnosť prechodovej funkcie $\delta$ znamená, že pre niektoré dvojice $(q, c) \in K \times (\Gamma \cup \{\mathbf{B}\})$ môže byť jej výstup aj nedefinovaný – to sa niekedy označuje ako $\delta(q, c) = \perp$. Symbol $\nrightarrow$, použitý v uvedenej definícii, sa občas používa namiesto klasickej šípky na zdôraznenie čiastočnosti funkcie.

Deterministický Turingov stroj sa niekedy definuje aj ako špeciálny prípad nedeterministického Turingovho stroja – pre každé $q \in K$ a $c \in \Gamma$ je potom $\delta(q, c)$ najviac jednoprvková podmnožina množiny $K \times \Gamma \times \{-1, 0, 1\}$. Aj keď ide o formálne odlišnú definíciu, korešpondencia s tou našou by mala byť očividná. Možno sa tiež stretnúť s definíciou, pri ktorej $\mathbf{B} \in \Gamma$. V takom prípade je prechodovou funkciou zobrazenie $\delta\colon K \times \Gamma \nrightarrow K \times (\Gamma - \{\mathbf{B}\}) \times \{-1, 0, 1\}$.

Podobne ako pri konečných a zásobníkových automatoch, rozumieme aj pri Turingových strojoch pod konfiguráciou objekt nesúci informáciu o dosiaľ vykonanej časti výpočtu postačujúcu na to, aby bolo možné iba na základe nej vo výpočte pokračovať. Z toho je zrejmé, že v konfigurácii musí byť nejakým spôsobom zakódovaný stav, obsah pásky, ako aj pozícia čítacej hlavy. Objekt nesúci takúto informáciu možno definovať množstvom spôsobov, pričom v rôznych situáciách môžu byť výhodné rôzne formalizácie. Z tohto dôvodu konfiguráciu deterministického Turingovho stroja definujeme hneď v troch variantoch, pričom neskôr budeme vždy pracovať s variantom, ktorý sa bude javiť ako momentálne najvhodnejší.

**Definícia 2.** Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj.

*(i)* *Konfiguráciou* stroja $A$ možno nazvať trojicu $(q, w, n) \in K \times \mathbf{B}\Gamma^*\mathbf{B} \times \mathbb{N}$, kde $q$ je stav, $w$ je slovo reprezentujúce obsah zapísanej časti pásky s dvoma susednými symbolmi „blank“ a číslo $n \in \{0, \ldots, |w| - 1\}$ udáva pozíciu čítacej hlavy vzhľadom na začiatok zapísanej časti pásky.[1]

*(ii)* Nech $\upharpoonleft \notin \Gamma$. *Konfiguráciou* Turingovho stroja $A$ možno v takom prípade rozumieť aj dvojicu $(q, w) \in K \times (\upharpoonleft\mathbf{B}\Gamma^*\mathbf{B} \cup \mathbf{B}\Gamma^*\upharpoonleft\Gamma^*\mathbf{B})$, kde $q$ je stav a $w$ je slovo reprezentujúce obsah zapísanej časti pásky s dvoma susednými symbolmi „blank“ a pozíciou hlavy vyznačenou špeciálnym symbolom $\upharpoonleft$ – hlava v takom prípade číta písmeno nasledujúce bezprostredne za $\upharpoonleft$.

*(iii)* Nech $K \cap \Gamma = \emptyset$. *Konfiguráciou* stroja $A$ možno nazvať aj slovo $w \in K\mathbf{B}\Gamma^*\mathbf{B} \cup \mathbf{B}\Gamma^*K\Gamma^*\mathbf{B}$. Špeciálny symbol $\upharpoonleft$ z variantu *(ii)* je tu nahradený priamo stavom stroja $A$, ktorý v takom prípade nie je nutné udržiavať ako samostatnú zložku konfigurácie.

*Príklad* 1. Uvažujme konfiguráciu deterministického Turingovho stroja, ktorý je v stave $q$, na páske má zapísané slovo $aaba$ a ktorého hlava číta druhý symbol $a$. Vo variante *(i)* by sme túto konfiguráciu zapísali ako trojicu $(q, \mathbf{B}aaba\mathbf{B}, 2)$, vo variante *(ii)* by išlo o dvojicu $(q, \mathbf{B}a\upharpoonleft aba\mathbf{B})$ a vo variante *(iii)* o slovo $\mathbf{B}aqaba\mathbf{B}$. V prípade, že hlava číta prvý symbol „blank“ naľavo od zapísanej časti pásky, zapísali by sme konfiguráciu vo variante *(i)* ako $(q, \mathbf{B}aaba\mathbf{B}, 0)$, vo variante *(ii)* ako $(q, \upharpoonleft\mathbf{B}aaba\mathbf{B})$ a vo variante *(iii)* ako $q\mathbf{B}aaba\mathbf{B}$.

[1] Ak teda $w = \mathbf{B}a_1 \ldots a_m\mathbf{B}$ pre nejaké $m \in \mathbb{N}$ a hlava číta symbol $a_k$ pre $k \in [m]$, je $n = k$. V prípade, že hlava číta symbol $\mathbf{B}$ pred začiatkom slova $a_1 \ldots a_m$, je $n = 0$; pokiaľ číta symbol $\mathbf{B}$ za slovom $a_1 \ldots a_m$, je $n = m + 1$.

Bez ohľadu na variant sú okrem zapísanej časti pásky súčasťou konfigurácie aj dva susedné symboly $\mathbf{B}$. Keď totiž hlava Turingovho stroja spraví z prvého zapísaného políčka pásky krok doľava – prípadne z posledného zapísaného políčka krok doprava – bude v ďalšej konfigurácii čítať práve niektorý z týchto symbolov $\mathbf{B}$. Nikdy sa však nemôže stať, že by hlava čítala iné nezapísané políčko: prechodová funkcia $\delta$ totiž nemôže zapisovať $\mathbf{B}$ – a to ani vtedy, keď bol „blank" pôvodným obsahom políčka. Čítacia hlava sa tak nikdy nemôže príliš „vzdialiť" od zapísanej časti pásky.

Turingov stroj možno ekvivalentne definovať aj tak, aby mal možnosť zapisovať $\mathbf{B}$; konfiguráciu takéhoto stroja by však bolo nutné definovať opatrnejšie.

**Definícia 3.** Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. *Krok výpočtu* stroja $A$ je binárna relácia $\vdash_A$ na konfiguráciách tohto stroja definovaná nasledovne:

(*i*) Pre všetky $p, q \in K$, $n \in \mathbb{N}$, $a_1, \ldots, a_n, c \in \Gamma$, $k \in [n]$ a $d \in \{-1, 0, 1\}$ je

$$(p, \mathbf{B}a_1 \ldots a_{k-1}a_k a_{k+1} \ldots a_n\mathbf{B}, k) \vdash_A (q, \mathbf{B}a_1 \ldots a_{k-1}ca_{k+1} \ldots a_n\mathbf{B}, k + d)$$

práve vtedy, keď $\delta(p, a_k) = (q, c, d)$.

(*ii*) Pre všetky $p, q \in K$, $n \in \mathbb{N}$, $a_1, \ldots, a_n, c \in \Gamma$ a $d \in \{-1, 0, 1\}$ je

$$(p, \mathbf{B}a_1 \ldots a_n\mathbf{B}, 0) \vdash_A (q, \mathbf{B}ca_1 \ldots a_n\mathbf{B}, d + 1)$$

práve vtedy, keď $\delta(p, \mathbf{B}) = (q, c, d)$.

(*iii*) Pre všetky $p, q \in K$, $n \in \mathbb{N}$, $a_1, \ldots, a_n, c \in \Gamma$ a $d \in \{-1, 0, 1\}$ je

$$(p, \mathbf{B}a_1 \ldots a_n\mathbf{B}, n + 1) \vdash_A (q, \mathbf{B}a_1 \ldots a_nc\mathbf{B}, n + 1 + d)$$

práve vtedy, keď $\delta(p, \mathbf{B}) = (q, c, d)$.

(*iv*) V relácii $\vdash_A$ nie sú žiadne ďalšie dvojice konfigurácií.

V prípade, že je uvažovaný stroj $A$ zrejmý z kontextu, píšeme namiesto $\vdash_A$ iba $\vdash$.

Jazyk akceptovaný Turingovým strojom sa v literatúre definuje množstvom rôznych – ale vzájomne ekvivalentných – spôsobov. Podľa našej definície bude stroj akceptovať svoj vstup, kedykoľvek sa na ňom dokáže dostať do akceptačného stavu – bez ohľadu na to, či sa výpočet v tomto stave zastaví a či bol vstup dočítaný až do konca.

**Definícia 4.** Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je deterministický Turingov stroj. *Jazyk* akceptovaný strojom $A$ je daný ako

$$L(A) = \{w \in \Sigma^* \mid \exists q \in F\ \exists u \in \Gamma^*\ \exists k \in \mathbb{N} : (q_0, \mathbf{B}w\mathbf{B}, 1) \vdash^* (q, \mathbf{B}u\mathbf{B}, k)\}.$$

Podľa predchádzajúcej definície teda Turingov stroj akceptuje svoj vstup aj v prípade, že výpočet cez akceptačnú konfiguráciu iba „prejde", pričom skončí v neakceptačnej konfigurácii alebo neskončí vôbec (pod koncom výpočtu tu máme na mysli situáciu, keď sa stroj „zasekne", pretože na daný symbol nie je z daného stavu definovaný žiaden prechod).

Ľahko však vidieť, že naša definícia akceptovaného jazyka je ekvivalentná zdanlivo prísnejšej definícii požadujúcej *zastavenie* výpočtu v akceptačnom stave. Ak totiž výpočet vyhovuje takejto silnejšej podmienke akceptácie, triviálne vyhovuje aj podmienke z definície 7. Pre dôkaz opačným smerom si stačí uvedomiť, že prechody vedúce z akceptačných stavov sú zbytočné: ak sa stroj raz dostane do niektorého akceptačného stavu, nemusí už vo výpočte ďalej pokračovať, pretože o akceptovaní vstupu „už je rozhodnuté". Po odobratí takýchto zbytočných prechodov teda zjavne dostávame Turingov stroj akceptujúci rovnaký jazyk podľa pozmenenej definície.

Definíciu akceptovaného jazyka je možné ekvivalentne upraviť aj tak, aby bol stroj donútený pred akceptovaním prečítať celý vstup. Detaily konštrukcie tu prenechávame čitateľovi.

## 2 Nedeterministické Turingove stroje

Princíp nedeterminizmu je pri Turingových strojoch rovnaký ako pri konečných alebo zásobníkových automatoch: nedeterministický Turingov stroj môže mať pre daný stav a symbol definovaných vo všeobecnosti aj viacero rôznych prechodov, pričom v rámci kroku výpočtu z „kompatibilnej“ konfigurácie sa môže použiť ktorýkoľvek z nich. Na jednom vstupe teda môže existovať aj viacero výpočtov, pričom nedeterministický Turingov stroj svoj vstup akceptuje práve vtedy, keď *existuje aspoň jeden* akceptačný výpočet.

**Definícia 5.** *Nedeterministický Turingov stroj* je šestica $A = (K, \Sigma, \Gamma, \delta, q_0, F)$, kde $K$ je konečná množina stavov, $\Sigma$ je vstupná abeceda, $\Gamma \supseteq \Sigma$ je pracovná abeceda neobsahujúca špeciálny symbol $\mathbf{B} \notin \Gamma$ („blank“), $\delta\colon K \times (\Gamma \cup \{\mathbf{B}\}) \to 2^{K \times \Gamma \times \{-1,0,1\}}$ je prechodová funkcia, $q_0 \in K$ je počiatočný stav a $F \subseteq K$ je množina akceptačných stavov.

*Konfigurácie* nedeterministického Turingovho stroja definujeme navlas rovnako ako pri strojoch deterministických – možno použiť ľubovoľný z variantov definície 2. Málo prekvapivou úpravou definície pre deterministické Turingove stroje získame aj nasledujúcu definíciu *kroku výpočtu*.

**Definícia 6.** Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je nedeterministický Turingov stroj. *Krok výpočtu* stroja $A$ je binárna relácia $\vdash_A$ na konfiguráciách tohto stroja definovaná nasledovne:

(*i*) Pre všetky $p, q \in K$, $n \in \mathbb{N}$, $a_1, \ldots, a_n, c \in \Gamma$, $k \in [n]$ a $d \in \{-1, 0, 1\}$ je

$$(p, \mathbf{B}a_1 \ldots a_{k-1}a_ka_{k+1} \ldots a_n\mathbf{B}, k) \vdash_A (q, \mathbf{B}a_1 \ldots a_{k-1}ca_{k+1} \ldots a_n\mathbf{B}, k + d)$$

práve vtedy, keď $(q, c, d) \in \delta(p, a_k)$.

(*ii*) Pre všetky $p, q \in K$, $n \in \mathbb{N}$, $a_1, \ldots, a_n, c \in \Gamma$ a $d \in \{-1, 0, 1\}$ je

$$(p, \mathbf{B}a_1 \ldots a_n\mathbf{B}, 0) \vdash_A (q, \mathbf{B}ca_1 \ldots a_n\mathbf{B}, d + 1)$$

práve vtedy, keď $(q, c, d) \in \delta(p, \mathbf{B})$.

(*iii*) Pre všetky $p, q \in K$, $n \in \mathbb{N}$, $a_1, \ldots, a_n, c \in \Gamma$ a $d \in \{-1, 0, 1\}$ je

$$(p, \mathbf{B}a_1 \ldots a_n\mathbf{B}, n + 1) \vdash_A (q, \mathbf{B}a_1 \ldots a_nc\mathbf{B}, n + 1 + d)$$

práve vtedy, keď $(q, c, d) \in \delta(p, \mathbf{B})$.

(*iv*) V relácii $\vdash_A$ nie sú žiadne ďalšie dvojice konfigurácií.

V prípade, že je uvažovaný stroj $A$ zrejmý z kontextu, píšeme namiesto $\vdash_A$ iba $\vdash$.

Nasledujúca definícia *akceptovaného jazyka* je opäť rovnaká ako pri deterministických Turingových strojoch.

**Definícia 7.** Nech $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je nedeterministický Turingov stroj. *Jazyk* akceptovaný strojom $A$ je daný ako

$$L(A) = \{w \in \Sigma^* \mid \exists q \in F\ \exists u \in \Gamma^*\ \exists k \in \mathbb{N} : (q_0, \mathbf{B}w\mathbf{B}, 1) \vdash^* (q, \mathbf{B}u\mathbf{B}, k)\}.$$

## 3 Riešená úloha

**Úloha 1.** Skonštruujte deterministický alebo nedeterministický Turingov stroj akceptujúci jazyk $L = \{ww \mid w \in \{a,b\}^*\}$.

*Riešenie.* Skonštruujeme *nedeterministický* Turingov stroj $A$ pracujúci nasledovne: ak na začiatku výpočtu číta hlava symbol „blank“, stroj akceptuje, pretože vstupom je prázdne slovo; v opačnom prípade si stroj zapamätá prvý symbol vstupného slova a niektorý z jeho opakovaných výskytov nedeterministicky prehlási za začiatok druhej polovice slova. Následne už len stroj zistí, či sa sufix vstupného slova začínajúci týmto symbolom skutočne rovná prefixu končiacemu bezprostredne pred ním. To realizuje postupným porovnávaním jednotlivých symbolov oboch častí slova – už spracované symboly v prvej časti slova si pritom bude označovať čiarou a pre označenie spracovaných symbolov v druhej časti slova bude stroj používať dve čiary. V prípade nezhody sa stroj zasekne. Pokiaľ naopak stroj zistí u všetkých dvojíc symbolov zhodu, pričom sú v oboch častiach slova naozaj spracované všetky symboly, stroj akceptuje.

Nedeterministický Turingov stroj $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ teda bude daný nasledovne:

$$K = \{\mathrm{START}, \mathrm{INIT}_a, \mathrm{INIT}_b, \mathrm{BACK}, \mathrm{NEXT}, \langle a\rangle_1, \langle b\rangle_1, \langle a\rangle_2, \langle b\rangle_2, \mathrm{CHECK}, \mathrm{ACC}\},$$

$\Sigma = \{a, b\}$, $\Gamma = \{a, b, a', b', a'', b''\}$,

$$\begin{aligned}
\delta(\mathrm{START}, \mathbf{B}) &= \{(\mathrm{ACC}, a, 0)\}, & & \\
\delta(\mathrm{START}, a) &= \{(\mathrm{INIT}_a, a', 1)\}, & \delta(\mathrm{START}, b) &= \{(\mathrm{INIT}_b, b', 1)\}, \\
\delta(\mathrm{INIT}_a, a) &= \{(\mathrm{INIT}_a, a, 1), (\mathrm{BACK}, a'', 0)\}, & \delta(\mathrm{INIT}_a, b) &= \{(\mathrm{INIT}_a, b, 1)\}, \\
\delta(\mathrm{INIT}_b, a) &= \{(\mathrm{INIT}_b, a, 1)\}, & \delta(\mathrm{INIT}_b, b) &= \{(\mathrm{INIT}_b, b, 1), (\mathrm{BACK}, b'', 0)\}, \\
\delta(\mathrm{BACK}, a'') &= \{(\mathrm{BACK}, a'', -1)\}, & \delta(\mathrm{BACK}, b'') &= \{(\mathrm{BACK}, b'', -1)\}, \\
\delta(\mathrm{BACK}, a) &= \{(\mathrm{BACK}, a, -1)\}, & \delta(\mathrm{BACK}, b) &= \{(\mathrm{BACK}, b, -1)\}, \\
\delta(\mathrm{BACK}, a') &= \{(\mathrm{NEXT}, a', 1)\}, & \delta(\mathrm{BACK}, b') &= \{(\mathrm{NEXT}, b', 1)\}, \\
\delta(\mathrm{NEXT}, a) &= \{(\langle a\rangle_1, a', 1)\}, & \delta(\mathrm{NEXT}, b) &= \{(\langle b\rangle_1, b', 1)\}, \\
\delta(\mathrm{NEXT}, a'') &= \{(\mathrm{CHECK}, a'', 1)\}, & \delta(\mathrm{NEXT}, b'') &= \{(\mathrm{CHECK}, b'', 1)\}, \\
\delta(\langle a\rangle_1, a) &= \{(\langle a\rangle_1, a, 1)\}, & \delta(\langle a\rangle_1, b) &= \{(\langle a\rangle_1, b, 1)\}, \\
\delta(\langle a\rangle_1, a'') &= \{(\langle a\rangle_2, a'', 1)\}, & \delta(\langle a\rangle_1, b'') &= \{(\langle a\rangle_2, b'', 1)\}, \\
\delta(\langle b\rangle_1, a) &= \{(\langle b\rangle_1, a, 1)\}, & \delta(\langle b\rangle_1, b) &= \{(\langle b\rangle_1, b, 1)\}, \\
\delta(\langle b\rangle_1, a'') &= \{(\langle b\rangle_2, a'', 1)\}, & \delta(\langle b\rangle_1, b'') &= \{(\langle b\rangle_2, b'', 1)\}, \\
\delta(\langle a\rangle_2, a'') &= \{(\langle a\rangle_2, a'', 1)\}, & \delta(\langle a\rangle_2, b'') &= \{(\langle a\rangle_2, b'', 1)\}, \\
\delta(\langle a\rangle_2, a) &= \{(\mathrm{BACK}, a'', 0)\}, & & \\
\delta(\langle b\rangle_2, a'') &= \{(\langle b\rangle_2, a'', 1)\}, & \delta(\langle b\rangle_2, b'') &= \{(\langle b\rangle_2, b'', 1)\}, \\
\delta(\langle b\rangle_2, b) &= \{(\mathrm{BACK}, b'', 0)\}, & & \\
\delta(\mathrm{CHECK}, a'') &= \{(\mathrm{CHECK}, a'', 1)\}, & \delta(\mathrm{CHECK}, b'') &= \{(\mathrm{CHECK}, b'', 1)\}, \\
\delta(\mathrm{CHECK}, \mathbf{B}) &= \{(\mathrm{ACC}, a, 0)\}, & &
\end{aligned}$$

$\delta(q, c) = \emptyset$ pre všetky ostatné dvojice $(q, c) \in K \times (\Gamma \cup \{\mathbf{B}\})$, $q_0 = \mathrm{START}$ a $F = \{\mathrm{ACC}\}$.

Stroj najprv v počiatočnom stave START zistí, či je vstupné slovo prázdne; v takom prípade akceptuje. V opačnom prípade zistí, či je prvým symbolom vstupného slova $a$ alebo $b$; tento symbol si označí čiarou a v stave $\mathrm{INIT}_a$ resp. $\mathrm{INIT}_b$ nedeterministicky uhádne opakovaný výskyt toho istého symbolu na začiatku druhej polovice vstupného slova, ktorý si označí dvoma čiarami. Následne sa v stave BACK posunie nazad na prvý neoznačený symbol prvej časti a prepne sa do stavu NEXT. Stroj v stave NEXT obvykle začne porovnávať ďalšiu dvojicu symbolov v oboch častiach slova – symbol v prvej časti slova si označí čiarou, následne v stavoch $\langle a\rangle_1$ a $\langle a\rangle_2$ resp. $\langle b\rangle_1$ a $\langle b\rangle_2$ nájde príslušný symbol v druhej časti slova a v prípade zhody so zapamätaným symbolom ho označí dvoma čiarami. V stave BACK sa následne hlava opäť prestaví na prvý neoznačený symbol prvej časti slova a stav sa zmení na NEXT; môže sa tak začať porovnanie ďalšej dvojice symbolov. V prípade, že stroj narazí v stave NEXT na symbol označený dvoma čiarami, boli už všetky symboly prvej časti slova úspešne porovnané s ich náprotivkami v druhej časti. Zostáva už teda iba overiť správnosť nedeterministického rozhodnutia o začiatku druhej polovice slova – to znamená zistiť, či za označenými symbolmi druhej časti slova nenasledujú ešte nejaké neoznačené symboly. Túto kontrolu stroj realizuje v stave CHECK; v prípade úspechu sa prepne do akceptačného stavu ACC.

Poznamenajme ešte, že v prechodoch na prečítaný symbol $\mathbf{B}$ by sme mohli namiesto symbolu $a$ zapísať aj ľubovoľný iný symbol okrem $\mathbf{B}$ (ktorý Turingove stroje podľa ich definície zapisovať nemôžu). $\square$

Jazyk z predchádzajúcej úlohy možno veľmi podobne akceptovať aj *deterministickým* Turingovým strojom. Nedeterminizmus sme totiž v našej konštrukcii využívali iba pri hľadaní stredu vstupného slova – to však možno realizovať aj deterministicky, napríklad postupným označovaním prvého a posledného neoznačeného písmena vstupu. Pokračovať potom možno v princípe rovnako ako vyššie.

# 4 Ekvivalencia deterministických a nedeterministických strojov

Podobne ako sa každý deterministický konečný automat zvykne súčasne považovať aj za nedeterministický konečný automat, možno aj každý deterministický Turingov stroj chápať ako špeciálny nedeterministický Turingov stroj. Táto predstava síce v ani jednom prípade úplne nekorešponduje s formálnymi definíciami jednotlivých modelov, no celý nesúlad tkvie v skutočnosti, že kým výstupom prechodovej funkcie je v deterministickom variante nejaký objekt, v nedeterministickom variante je to množina takýchto objektov.

Korektná formalizácia korešpondencie medzi deterministickým a k nemu prislúchajúcim nedeterministickým strojom je teda nasledovná: deterministický Turingov stroj $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ môžeme chápať aj ako nedeterministický Turingov stroj $A' = (K, \Sigma, \Gamma, \delta', q_0, F)$, kde prechodová funkcia $\delta'$ je pre všetky $q \in K$ a $c \in \Gamma$ daná ako $\delta'(q, c) = \{\delta(q, c)\}$ kedykoľvek je hodnota $\delta(q, c)$ definovaná; v opačnom prípade je $\delta'(q, c) = \emptyset$. Evidentne $L(A') = L(A)$. Ku každému deterministickému Turingovmu stroju teda existuje ekvivalentný a v princípe totožný nedeterministický stroj.

Na dôkaz ekvivalencie deterministických a nedeterministických Turingových strojov teda stačí ukázať, že výpočet nedeterministického stroja možno odsimulovať aj na stroji deterministickom. To možno urobiť napríklad prehľadávaním stromu konfigurácii nedeterministického Turingovho stroja do šírky. Prehľadávanie do hĺbky použiť nemožno, keďže takýto strom konfigurácii je vo všeobecnosti nekonečný (vždy však má konečné vetvenie).

Nech teda $A = (K, \Sigma, \Gamma, \delta, q_0, F)$ je nedeterministický Turingov stroj. Princíp deterministického Turingovho stroja $A' = (K', \Sigma', \Gamma', \delta', q_0', F')$ simulujúceho stroj $A$ možno neformálne a v hrubých rysoch opísať nasledovne:

1. Ak $A'$ dostane na vstupe slovo $w$, upraví obsah pásky na počiatočnú konfiguráciu stroja $A$ na vstupe $w$. Zapísaná časť pásky teda bude mať obsah $\overline{B}q_0w\overline{B}$, kde $\overline{B}$ je „falošný blank“. V nasledujúcom budeme obsah pásky interpretovať ako FIFO front konfigurácií stroja $A$ oddelených špeciálnym symbolom #. Symboly $\overline{B}$ pritom budeme interpretovať ako symboly $\mathbf{B}$ susediace so zapísanou časťou pásky (ktoré sú podľa definície súčasťou konfigurácie). Na začiatku simulácie je teda obsahom frontu jediná konfigurácia $\mathbf{B}q_0w\mathbf{B}$.

2. Opakuj v nekonečnom cykle:

   2.1 Ak je front prázdny, ukonči cyklus a vstup $w$ zamietni.

   2.2 V opačnom prípade označme ako $C$ prvú konfiguráciu frontu.

   2.3 Ak je konfigurácia $C$ akceptačná (obsahuje akceptačný stav), ukonči cyklus a vstup $w$ akceptuj.

   2.4 V opačnom prípade pridaj na koniec frontu všetky konfigurácie $C'$ také, že $C \vdash_A C'$. Takýchto konfigurácií je vždy konečne veľa (ak je v konfigurácii $C$ stroj v stave $q$ a hlava číta písmeno $c$, ich počet je $|\delta(q, c)|$). Zmaž konfiguráciu $C$ zo začiatku frontu.

Z hľadiska implementácie na deterministickom Turingovom stroji je najmenej triviálnou časťou uvedeného algoritmu krok 2.4. Stroj môže pracovať napríklad tak, že si najprv v stave zapamätá všetky trojice z množiny $\delta(q, c)$, následne $|\delta(q, c)|$-krát skopíruje konfiguráciu $C$ na koniec frontu (pričom jednotlivé kópie pooddeľuje symbolom #) a napokon využije informácie zapamätané v stave na úpravu všetkých týchto kópií podľa trojíc z $\delta(q, c)$.