

# Turingův stroj — emulátor

Jiří Fišer, Petr Kubera

**Stručné zadání:** Vytvořte v jazyce C program — emulátor Turingova stroje, jehož program (přechodová tabulka) je uložen v souboru ve formátu CSV, počáteční stav pásky je čten ze standardního vstupu a koncový stav pásky je vypisován na výstup standardní. Dalším výstupem z programu může být HTML soubor, jenž bude obsahovat popis všech cyklů stroje (s výpisem stavů, použitých přechodových pravidel a grafickým zobrazením aktuální podoby pásky).

## 1 Turingův stroj – popis

Turingův stroj je matematický model jednoduchého počítače zavedený britským matematikem Turingem. I když se tento model výrazně liší od architektury současných počítačů, má stále velký význam v teorii algoritmů a programování (lze pomocí něj dokázat možnost resp. nemožnost konečného počítačového vyhodnocení určitých matematických problémů). Turingův stroj lze popsat zcela přesně pomocí formálního matematického zápisu, zde však dáme nejdříve přednost méně formálnímu popisu.

Předpokládejme, že existuje potenciálně nekonečná (orientovaná) páska, rozdělená na políčka, z nichž každé může být buď prázdné nebo obsahovat jeden symbol. Obecně to mohou být libovolné symboly, pro zjednodušení však předpokládejme, že se jedná o tisknutelné znaky (což můžeme učinit bez ztráty dostatečné obecnosti). Dále lze pro zjednodušení přijmout představu, že i prázdná políčka obsahují speciální symboly tzv. prázdné symboly. Prázdný symbol však musí být odlišitelný od běžných symbolů (tj. v našem zjednodušení od znaků) tj. musí být různý od libovolného z nich.

Dále předpokládejme, že existuje stroj, který je schopen jednotlivé znaky (symboly) číst z pásky a také jen na pásku zapisovat. Stroj pracuje v cyklu, v němž nejdříve přečte znak z pásky, poté na stejnou pozici zapíše nový znak (jenž se však nemusí lišit od znaku původního) a poté se přesune buď na políčko bezprostředně následující (posun vpřed) nebo na políčko bezprostředně předcházející (posun vzad). Teprve poté může pokračovat dalším cyklem (čtení, zápis, posun, atd.).

Jak se stroj zachová po přečtení znaku zachová závisí jednak na přečteném znaku tak na vnitřním stavu stroje. Stroj se může nacházet pouze v konečném počtu stavů (jednotlivé stavy tak lze označit identifikátory). Po provedení zápisu se může stav změnit. Tato změna, stejně jako zapsaný znak je deterministická, tj. je jednoznačně určena původním stavem a přečteným znakem (a ničím jiným).

Aby však byl celý běh stroje plně deterministický (tj. plně předpověditelný) musí být jednoznačně určen jeho počáteční stav i počáteční pozice pásky, tj. pozice z níž bude přečten první znak.

Navíc i když je páska potenciálně nekonečná, musí být běh stroje, alespoň v některých případech v čase ukončen (nekonečně běžící, resp. počítající stroje jsou nezajímavé, neboť výsledky chceme v konečném čase). Proto je možné některé ze stavů označit za tzv. koncové stavy, po jejichž dosažení je běh stroje zastaven (samozřejmě počáteční stav nemůže být zároveň koncový, neboť ani stroje nic nedělající nejsou příliš praktické).

Myšlený stroj, který jsme výše popsali je oním Turingovým strojem. Jeho vstupním i výstupním médiem je výše popsaná (myšlená) páska. Na počátku jeho běhu obsahuje konečný počet neprázdných symbolů (znaků),

kteře tvořĩ vstupnĩ informaci. Protože neprázdných symbolů je konečný počet existuje i na pásce pozice, za níž již jsou pouze prázdná políčka (=prázdné symboly) stejně jako pozice, u níž platí podobné tvrzení pro všechna políčka předcházející (tzv. horní resp. dolní mez vstupu). Jinak řečeno, vstup je vždy omezen na konečný úsek pásky. Navíc lze bez újmy na obecnosti požadovat, že počáteční pozice leží uvnitř daného úseku pásky a dokonce, že počáteční pozice je identická s dolní mezí vstupu.

Po skončení běhu stroje (tj. po dosažení koncového stavu<sup>1</sup>) obsahuje páska výstup (výstupní informaci), jež je opět omezena na konečnou část pásky (všechny políčka mimo tuto část jsou vždy prázdná). Turingův stroj tedy nahrazuje jednu konečnou posloupnost symbolů za jinou resp. při omezení symbolů na znaky, jeden řetězec za druhý.

## 2 Formální popis Turingova stroje

Turingův stroj je jednoznačně popsán pomocí následujících množin:

vstupně-výstupní abeceda symbolů  $\Sigma$  (včetně prázdného symbolu  $\epsilon$ )

množina stavů  $Q$

počáteční stav (jednoprvková množina)  $\{q_0\}$ , kde  $q_0 \in Q$

množina koncových stavů  $F$ , kde  $F \subset Q$

množina přechodových pravidel  $\delta$ , kde pravidlo je zobrazení z kartézského součinu  $(Q - F) \times \Sigma$  do kartézského součinu  $\Sigma \times Q \times \{-1, 1\}$ . Symbol  $-1$  označuje posun hlavy vzad, symbol  $1$  vpřed.

## 3 Emulátor Turingova stroje

Vstupně-výstupní abecedou jsou všechny alfanumerické znaky ASCII sady (a-z, A-Z, 0-9), znaky závorek (kulatých, hranatých a složených) a znaky základních aritmetických operací (+, -, \*, /). Prázdný symbol je reprezentován znakem „#“.

Stavy jsou identifikovány identifikátory, jejichž syntaxe odpovídá identifikátorům jazyka C (tj. smí obsahovat jen alfanumerické znaky a podtržítko a nesmí začínat číslicí). Délka identifikátorů není omezena. Koncové stavy jsou určeny tabulkou pravidel (viz dále).

### Tabulka pravidel

Množina pravidel je popsána tabulkou, jejíž sloupce jsou uvozeny vstupními symboly a řádky jmény stavů. Poslední sloupec může obsahovat klíčové slovo OTHERS, jež sdružuje všechny symboly vstupní abecedy (bez koncového), které nebyly jmenovitě uvedeny v předchozích hlavičkách sloupců. Cíle pravidel mají formát *cílový-stav: nový-znak: posun*, kde je posun čtecí hlavy identifikován znakem + (posun vpřed, +1) resp. - (posun vzad, -1). Všechny části pravidel však lze vynechat. Není-li uveden cílový stav, pak se po aplikaci pravidla nezmění. Při neuvedení nového znaku se nemění znak na pásce, při neuvedení posunu se provede posun vpřed. Oddělovací dvojtečky však vynechat nelze (tj. minimální pravidlo má tvar ::).

Není-li v jistém políčku tabulky uveden cíl pravidla pak, pokud bude při běhu Turingova stroje toto pravidlo požadováno, skončí běh stroje s chybou (program je ukončen a je vypsáno hlášení na standardní chybový výstup). Detailnější informace o problému by měla být zapsána i do logovacího HTML souboru (je-li aktivován).

Speciálním případem jsou stavy u nichž není uveden žádný cíl (tj. není specifikováno žádné výstupní pravidlo). Tyto stavy jsou koncové a jejich dosažení ukončuje program bez chybové hlášky.

Počáteční stav stroje je prvním (nejhořejším) stavem tabulky (druhá řádka po záhlaví vstupních znaků). Tento stav je povinný. Pokud je počáteční stav zároveň koncovým je program ukončen s příslušným chybovým hlášením.

<sup>1</sup> pokud je vůbec některý z koncových stavů dosažen

Tabulka užívaná strojem je čtena programem z textového souboru užívajícím CSV syntaxi, v níž jsou jednotlivé položky (i prázdné!) odděleny čárkou resp. středníkem<sup>2</sup> a řádky odřádkováním. Tabulky v CSV syntaxi lze získat exportem z tabulkového procesoru (*OpenOffice Calc*, *Gnumeric*, apod.). V textu tabulky jsou povoleny i mezery (ne však v cílech pravidel).

Pokud tabulka neodpovídá výše uvedené struktuře, resp. jestliže obsahuje nepřípustné elementy (např. nepřípustné znaky vstupně-výstupní abecedy, nepřípustné identifikátory, chybné nebo nejednoznačné cíle pravidel) musí program skončit s chybovým hlášením, ještě před provedením prvního kroku stroje. Chybové hlášení musí přesně specifikovat druh chyby i její pozici v textovém souboru (alespoň řádek).

## Páska

Páska stroje je nekonečná (omezená jen dostupnou pamětí) a na začátku je její relevantní část (mezi dolní a horní mezí vstupu) načtena ze standardního vstupu (ne ze souboru!) a po úspěšném skončení běhu je její relevantní část<sup>3</sup> vypsána na standardní výstup. Počáteční pozice je dána prvním znakem vstupního textu.

Při vstupu pásky (ještě před během stroje) musí být provedena kontrola, zda páska obsahuje je přípustné znaky (tj. znaky uvedené v záhlaví tabulky, resp. všechny přípustné znaky podle specifikace, obsahuje-li tabulka sloupec OTHERS). Kromě přípustných znaků je možné ve vstupním textu použít i bílé znaky (mezery, tabulátory, odřádkování), ty však musí být před zpracováním odstraněny.

## 4 HTML výstup mezistavů (HTML log)

Pro snadnější ladění programů pro Turingův stroj a vizualizaci jeho běhu, musí program nabízet možnost vytvoření logovacího (X)HTML souboru s popisem jednotlivých kroků Turingova stroje.

**Každý krok musí obsahovat tyto údaje (v daném pořadí):**

1. počáteční stav pásky (graficky za použití HTML tabulek), s vizuálním označením pozice čtecí hlavy
2. číslo kroku (první označen jedničkou)
3. aplikované pravidlo ve tvaru *počáteční-stav:vstupní-symbol -> koncový-stav:výstupní-symbol:posun* (bez krácení, s nápaditou grafickou úpravou)
4. koncový stav pásky (= počáteční stav dalšího kroku, pokud existuje)

Grafická reprezentace stavu pásky musí zobrazovat celou relevantní část pásky + aktuální pozici hlavy (není-li na v relevantní části). Pokud je hlava ve velké vzdálenosti od relevantní části (což je nejspíše chyba programu), nebo pokud relevantní část pásky přesáhne rozumnou velikost (např. 100 symbolů) může být stav vypsán ve zkrácené podobě (jeho tvar přenechávám Vaší invenci).

Pro zvýšení přehlednosti je povinné barevné označení všech políček pásky, jejíž symboly byly při dosavadním běhu programu pozměněny (tj. jejichž obsah byl alespoň jednou změněn).

## 5 Rozhraní programu

Program běžící podle výše uvedených specifikací musí mít toto minimální rozhraní na příkazovém řádku (nepovinný parametr je v hranatých závorkách):

```
turing -p tabulka-pravidel [ -l html-log ]
```

Páska je čtena ze standardního vstupu a vypisována na standardní výstup. Pokud má být počáteční stav čten ze souboru a výstup do souboru uložen lze použít přesměrování vstupu a výstupu např. takto:

```
turing -p program1.turing <input.txt >output.txt
```

<sup>2</sup>čárka a středníky se mohou střídát i v rámci jediného souboru

<sup>3</sup>ta se rozsahem může lišit od relevantní části při vstupu!

## 6 Nepovinná funkčnost

Následující přehled shrnuje některá rozšíření, jež nejsou povinná avšak jsou užitečná (a zvýší pravděpodobnost uznání seminární práce, jež ne zcela přesně splňuje výše uvedené specifikace):

- detekce některých jednoduchých situací vedoucích nekonečnému programu (cykly beze změn celkového stavu, nevratné opuštění relevantní části pásky)
- grafická reprezentace aktuálního stavu programu (GUI rozhraní), avšak při zachování povinné funkčnosti (textový vstup a výstup!)
- rozšířená syntaxe pravidel, vstupně výstupní abeceda se zachováním striktní zpětné kompatibility

## 7 Příklad Turingova stroje

Následující tabulka popisuje Turingův stroj, jež přijímá (na pásce) vstup složený ze znaků **a** a **b**<sup>4</sup>. Pokud má tento vstup tvar  $a^n b^n$  (tj. po  $n$ -znacích **a** následuje stejný počet znaků **b**) bude výstupem slovo **1** (= na pásce je kromě prázdných symbolů jediný znak a to **1**). Ve všech ostatních případech je výstupem slovo **0**.

Počáteční stav má jméno *odstran\_a*, koncovým je stav *konec*.

	a	b	#
odstran_a	nakonec:#:+	vymaz_vpravo:0:+	konec:1:
nakonec	::	::	odstran_b::-
odstran_b	vymaz_vlevo:0:-	nazacatek:#:-	konec:0:
nazacatek	::-	::-	odstran_a::+
vymaz_vpravo	:#:	:#:	konec::
vymaz_vlevo	:#:-	:#:-	konec::
konec			

Tatáž tabulka s rozepsáním všech cílů:

	a	b	#
odstran_a	nakonec:#:+	vymaz_vpravo:0:+	konec:1:+
nakonec	nakonec:a:+	nakonec:b:+	odstran_b:#:-
odstran_b	vymaz_vlevo:0:-	nazacatek:#:-	konec:0:+
nazacatek	nazacatek:a:-	nazacatek:b:-	odstran_a:#:+
vymaz_vpravo	vymaz_vpravo:#:+	vymaz_vpravo:#:+	konec:#:+
vymaz_vlevo	vymaz_vlevo:#:-	vymaz_vlevo:#:-	konec:#:+
konec			

---

<sup>4</sup>ohraničený zleva i zprava prázdnými symboly