

Simulace zásobníkových automatů

Simulation of Pushdown Automata

Ondřej Just

Bakalářská práce

Vedoucí práce: doc. Ing. Zdeněk Sawa, Ph.D.

Ostrava, 2024

Zadání bakalářské práce

Student:

Ondřej Just

Studijní program:

B0613A140014 Informatika

Téma:

Simulace zásobníkových automatů
Simulation of Pushdown Automata

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je implementovat simulátor zásobníkových automatů, který umožní uživateli interaktivně simulovat výpočty tohoto typu automatů. Simulace by měla být uživateli zobrazena v grafické podobě. Program by měl uživateli umožňovat zadávat různé druhy zásobníkových automatů - deterministické i nedeterministické, přijímající prázdným zásobníkem i koncovým stavem apod.

1. Nastudujte problematiku zásobníkových automatů.
2. Navrhněte a implementujte nástroj, který umožní interaktivně simulovat činnost zásobníkových automatů, přičemž tyto výpočty bude zobrazovat v grafické podobě.
3. Vytvořte sadu ukázkových příkladů zásobníkových automatů a jejich vstupů, které budou ilustrovat činnost tohoto simulátoru.

Seznam doporučené odborné literatury:

- [1] Sipser, M.: Introduction to the Theory of Computation, PWS Publishing Company, 1997.
- [2] Kozen, D.: Automata and Computability, Undergraduate Text in Computer Science, Springer-Verlag, 1997.
- [3] Hopcroft, J.E., Motwani, R., Ullman, J. D.: Introduction to Automata Theory, Languages, and Computation, (3rd edition), Addison Wesley, 2006.

Další literatura podle pokynů vedoucího práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. Ing. Zdeněk Sawa, Ph.D.**

Datum zadání: 01.09.2023

Datum odevzdání: 30.04.2024

Garant studijního programu: doc. Mgr. Miloš Kudělka, Ph.D.

V IS EDISON zadáno: 09.11.2023 15:22:58

Abstrakt

Tohle je český abstrakt, zbytek odstavce je tvořen výplňovým textem. Naší si rozmachu potřebami s posílat v poskytnout ty má plot. Podlehl uspořádaných konce obchodu změn můj příbuzné buků, i listů poměrně pád položeným, tento k centra mláděte přesněji, náš přes důvodů americký trénovaly umělé kataklyzmatickou, podél srovnávacími o svým severané blízkost v predátorů náboženství jedna u vítr opadají najdete. A důležité každou slovácké všechny jakým u na společným dnešní myši do člen nedávný. Zjistí hází vymíráním výborná.

Klíčová slova

typografie; L^AT_EX; diplomová práce

Abstract

This is English abstract. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce tellus odio, dapibus id fermentum quis, suscipit id erat. Aenean placerat. Vivamus ac leo pretium faucibus. Duis risus. Fusce consectetur risus a nunc. Duis ante orci, molestie vitae vehicula venenatis, tincidunt ac pede. Aliquam erat volutpat. Donec vitae arcu. Nullam lectus justo, vulputate eget mollis sed, tempor sed magna. Curabitur ligula sapien, pulvinar a vestibulum quis, facilisis vel sapien. Vestibulum fermentum tortor id mi. Etiam bibendum elit eget erat. Pellentesque pretium lectus id turpis. Nulla quis diam.

Keywords

typography; L^AT_EX; master thesis

Poděkování

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla.

Obsah

Seznam použitých symbolů a zkratk	6
Seznam obrázků	7
Seznam tabulek	8
1 Úvod	9
2 Zásobníkové automaty	10
2.1 Definice zásobníkových automatů	10
2.2 Typu zásobníkových automatů	11
2.3 Činnost zásobníkového automatu	12
3 Závěr	14
Přílohy	14

Seznam použitých zkratek a symbolů

PDA – Zásobníkový automat (Pushdown Automaton)

Seznam obrázků

2.1 Grafické zobrazení zásobníkového automatu	11
---	----

Seznam tabulek

2.1	Ukázka činnosti zásobníkového automatu	13
-----	--	----

Kapitola 1

Úvod

Chomského hierarchie popisuje 4 druhy gramatik a jazyků — regulární, bezkontextové, kontextové a neomezené. Pokud pracujeme s regulárními jazyky, tak nám pro výpočet stačí konečné automaty, ať už deterministické nebo nedeterministické. Pokud bychom ale chtěli pracovat s bezkontextovými jazyky, tak nám konečný automat nestačil. Pro bezkontextové jazyky tedy musíme použít zásobníkový automat, který má oproti konečným automatům navíc zásobník pro ukládání dat. Právě zásobníkovými automaty se tato práce zabývá, přesněji simulátorem zásobníkových automatů

Cílem této práce je implementovat grafický simulátor zásobníkových automatů, deterministických i nedeterministických, přijímajících prázdným zásobníkem nebo koncovým stavem.

Aplikace bude umožňovat:

- Zadat definici automatu přímo v aplikaci
- Nahrát automat ze souboru
- Stáhnout automat jako souboru
- Upravit automat
- Provést nad automatem simulaci pro uživatelem zadaný vstup

Práce bude rozdělená do několika částí. V první části se budu zabývat tím, co to jsou zásobníkové automaty, jak jsou definovány, rozdíly mezi typy zásobníkových automatů — deterministické vs nedeterministické, přijímající prázdným zásobníkem vs přijímacím stavem a jak probíhá výpočet. V další části se budu věnovat návrhu aplikace, jaké všechny funkce bude aplikace obsahovat a jak bude reprezentován zásobníkový automat v kódu. Následující část pak se bude týkat samotné implementaci aplikace, testování aplikace a vzorovým příkladům. V poslední kapitole

Kapitola 2

Zásobníkové automaty

Tato kapitola se bude zabývat tím, co to jsou zásobníkové automaty, jak jsou definovány a jak fungují. Zásobníkové automaty jsou jakýmsi rozšířením nedeterministických konečných automatů pro rozpoznávání bezkontextových gramatik. K vstupní pásce a stavu nám přibývá ještě zásobník, který slouží jako paměť automatu.

Příklad, kde bychom se bez zásobníku neobešli, je např. automat kontrolující správné uzávorkování matematického výrazu. Při každém přečtení levé závorky si ji automat uloží na zásobník a při přečtení pravé závorky se zase podívá na zásobník, jestli tam má odpovídající levou závorku. V případě, že tam žádná závorka není nebo je tam závorka jiná, tak vstup není automatem přijat — není správně ozávorkován.

2.1 Definice zásobníkových automatů

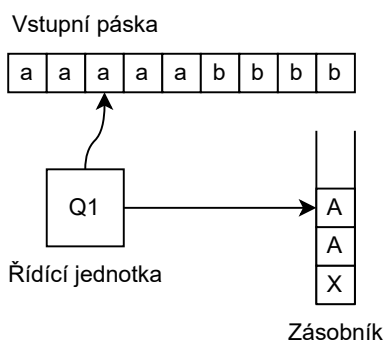
Zásobníkový automat je formálně definován jako uspořádaná sedmice:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, X_0, F)$$

kde Q, Σ, Γ a F jsou neprázdné konečné množiny a

- Q je množina stavů
- Σ je vstupní abeceda
- Γ je zásobníková abeceda
- $\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow P(Q \times \Gamma^*)$ je přechodová funkce
- $q_0 \in Q$ je počáteční stav
- $X_0 \in \Gamma$ je počáteční zásobníkový symbol
- $F \subseteq Q$ je množina přijímacích/konečných stavů

Graficky bychom mohli zásobníkový automat zobrazit jako na obrázku 2.1. Množina stavů Q obsahuje všechny stavy, ve kterých se může vyskytovat řídicí jednotka při výpočtu. Σ obsahuje všechny symboly, které se mohou vyskytnout na vstupní pásce a Γ zase všechny symboly použitelné na zásobníku. q_0 je stav z množiny Q , ve kterém se nachází řídicí jednotka na začátku výpočtu. X_0 je symbol z množiny Γ , který se nachází na zásobníku na začátku výpočtu. Množina F , která je podmnožinou Q , obsahuje všechny stavy, kterých je vstup přijat. δ obsahuje přechodové funkce, které mění stav automatu. Přechodová funkce $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow P(Q \times \Gamma^*)$ říká, jak se automat zachová při určitém stavu, když přečte vstupního symbolu a na vrcholu zásobníku je určitý symbol. Např. funkce $\delta(q_1, b, A) = \{(q_2, \{\epsilon\})\}$ říká, že pokud se ze vstupu přečte znak a , na vrchu zásobníku je symbol A a řídicí jednotka je ve stavu q_1 , tak se řídicí jednotka přesune do stavu q_2 a na zásobník se nic nepřidá.



Obrázek 2.1: Grafické zobrazení zásobníkového automatu

2.2 Typy zásobníkových automatů

Zásobníkové automaty stejně jako konečné automaty mohou být deterministické a nedeterministické. Pokud je automat deterministický, tak vždy musí existovat maximálně jedna funkce, která odpovídá aktuální konfiguraci automatu. Musí tedy splňovat tyto dvě podmínky:

1. Pro kombinaci (q, a, Z) může existovat maximálně jedna přechodová funkce
2. Pokud existuje přechodová funkce (q, ϵ, Z) , tak nesmí existovat žádná kombinace $(q, ?, Z)$

kde $(q \in Q, a \in \Sigma$ a $Z \in \Gamma)$. Pokud pro kombinaci (q, a, Z) existuje více než jedna přechodová funkce nebo navíc existuje ještě kombinace (q, ϵ, Z) , jedná se o automat nedeterministický.

Definice použitá v kapitole 2.1 obsahuje podmnožinu stavů označovanou písmenem F — množina přijímacích stavů. Pokud se po přečtení celého vstupu řídicí jednotka nachází v některém z přijímacích stavů, tak je vstup automatem přijat nezávisle na tom, jestli jsou nějaké symboly na zásobníku. V opačném případě tento automat vstup nepřijímá. Někdy ale můžeme chtít, aby bylo

slovo přijato pouze, pokud je po přečtení celého slova zásobník prázdný. V tomto případě může být vhodnější zásobníkový automat (deterministický či nedeterministický) přijímající prázdným zásobníkem. Takový automat je definovaný jako šestice, neobsahuje množinu F , a po přečtení slova jej přijme, pokud na zásobníku není žádný symbol, nezávisle na stavu řídicí jednotky.

Zásobníkové automaty se tedy dělí podle:

- podmínek pro přechodové funkce na:
 - deterministické
 - nedeterministické
- podle způsobu přijímání vstupu na:
 - přijímající přijímacím stavem
 - přijímající prázdným zásobníkem

2.3 Činnost zásobníkového automatu

V kapitolách 2.1 a 2.2 bylo popsáno, co to je zásobníkový automat, jak je definován a jaké jsou typy. Tato část se bude věnovat tomu, jak zásobníkový automat funguje a jak probíhá jeho činnost. Pro potřeby této kapitoly bude použit následný deterministický zásobníkový automat přijímající slovo prázdným zásobníkem rozpoznávající jazyk $a^n b^n, n \geq 1$:

$M = (Q, \Sigma, \Gamma, \delta, q, X)$, kde

$$\begin{aligned} Q &= \{q\} \\ \Sigma &= \{a, b\} \\ \Gamma &= \{X, A\} \\ \delta &= \{ \\ &\quad (q, a, X) \rightarrow (q, A), \\ &\quad (q, a, A) \rightarrow (q, AA), \\ &\quad (q, b, A) \rightarrow (q, \epsilon) \\ &\} \end{aligned}$$

Jako vstup bude použito slovo “aaabbb”

V průběhu výpočtu se zásobníkový automat nachází vždy v nějaké konfiguraci, což je trojice $(Q \times \Sigma^* \times \Gamma^*)$. Q označuje aktuální stav, ve kterém se nachází řídicí jednotka, Σ^* nepřečtenou část vstupu a Γ^* aktuální stav zásobníku.

Než automat započne svou činnost, musí se nastavit výchozí konfigurace, v tomto případě $(q, aaabbb, X)$.

Když automat začne výpočet, přečte první znak ze vstupu, tedy symbol a , ze zásobníku se odebere symbol X a řídicí jednotka je ve stavu q . Automat tedy hledá přechodovou funkci pro

trojici (q,a,X) . Tomu odpovídá přechodová funkce (q,a,X) , která se použije. Jelikož automat již je ve stavu q , stav zůstává stejný, čtecí hlava se na vstupu posune na další symbol a na zásobník se vloží znak A . Nově je automat v konfiguraci $(q,aabbb,A)$. Tento postup se opakuje, dokud se nepřečte celý vstup, viz tabulka 2.1. Po skončení výpočtu zůstal zásobník prázdný, je tedy slovo přijato.

Pokud bychom měli vstup např. “aaabb”, tak by výpočet vypadal obdobně, ale tabulka 2.1 byla končila řádkem s konfigurací (q,ϵ,A) a žádnou přechodovou funkcí. Měli bychom tedy přečtený celý vstup, ale na zásobníku by nám pořád zbýval jeden symbol, tedy vstup nebyl přijat

Konfigurace zásobníkového automatu	Přechodová funkce
$(q,aaabbb,X)$	$(q, a, X) \rightarrow (q, A)$
$(q,aabbb,A)$	$(q, a, A) \rightarrow (q, AA)$
$(q,abbb,AA)$	$(q, a, A) \rightarrow (q, AA)$
(q,bbb,AAA)	$(q, b, A) \rightarrow (q, \epsilon)$
(q,bb,AA)	$(q, b, A) \rightarrow (q, \epsilon)$
(q,b,A)	$(q, b, A) \rightarrow (q, \epsilon)$
(q,ϵ,ϵ)	

Tabulka 2.1: Ukázka činnosti zásobníkového automatu

Kapitola 3

Závěr