Theoretical Computer Science

Tutorial Week 11

Prof. Andrey Frolov

INNOPOLIS UNIVERSITY

Agenda

Non-determinism:

- NDFSA = DFSA
  - RegExp $\rightarrow$ NDFSA $\rightarrow$ DFSA (the previous week)
  - **DFSA $\rightarrow$ RegExp (Kleene's Algorithm)**
  - Chomsky Grammars Hierarchy (Regular grammars)
- NDPDA
  - Definition & Example
  - Chomsky Grammars Hierarchy (Context-Free grammars)
- TM
  - Definition & Example
  - Chomsky Grammars Hierarchy (Unrestricted grammars)

Kleene's Algorithm

Let $M = (Q, A, \delta, q_0, F)$ an FSA, where $Q = \{q_0, \ldots, q_n\}$.

Step $k = -1$

$$R_{ij}^{-1} = \begin{cases} a_1 \mid \ldots \mid a_m, & \text{if } i \neq j, \text{where } \delta(q_i, a_t) = q_j \\ a_1 \mid \ldots \mid a_m \mid \epsilon, & \text{if } i = j, \text{where } \delta(q_i, a_t) = q_j \\ \emptyset, & \text{otherwise} \end{cases}$$

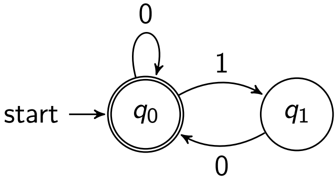
Step $k = 0, \ldots, n$

$$R_{ij}^{k} = R_{ik}^{k-1} \left(R_{kk}^{k-1}\right)^{*} R_{kj}^{k-1} \mid R_{ij}^{k-1}$$

Answer

$R_{0i_1}^{n} \mid \ldots \mid R_{0i_f}^{n}$, where $F = \{q_{i_1}, \ldots, q_{i_f}\}$ is the set of accept states

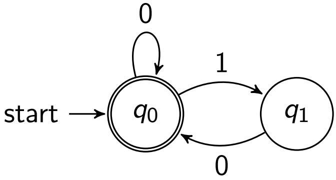
Kleene's Algorithm: Example 1



$$R_{ij}^{-1} = \begin{cases} a_1 \mid \ldots \mid a_m, & \text{if } i \neq j, \text{where } \delta(q_i, a_t) = q_j \\ a_1 \mid \ldots \mid a_m \mid \epsilon, & \text{if } i = j, \text{where } \delta(q_i, a_t) = q_j \\ \emptyset, & \text{otherwise} \end{cases}$$

$$\begin{aligned} R_{00}^{-1} &= 0 \mid \epsilon \\ R_{01}^{-1} &= 1 \\ R_{10}^{-1} &= 0 \\ R_{11}^{-1} &= \epsilon \end{aligned}$$

Kleene's Algorithm: Example 1

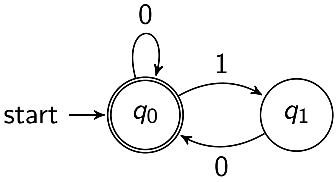


$$R_{ij}^{k} = R_{ik}^{k-1}(R_{kk}^{k-1})^{*}R_{kj}^{k-1} \mid R_{ij}^{k-1}$$

$$\begin{aligned} R_{00}^{-1} &= 0 \mid \epsilon \\ R_{10}^{-1} &= 0 \\ R_{01}^{-1} &= 1 \\ R_{11}^{-1} &= \epsilon \end{aligned}$$

$$\begin{aligned} R_{00}^{0} &= \underbrace{(0 \mid \epsilon)}_{R_{ik}^{k-1}}\underbrace{(0 \mid \epsilon)}_{R_{kk}^{k-1}}{}^{*}\,\underbrace{(0 \mid \epsilon)}_{R_{kj}^{k-1}} \mid \underbrace{(0 \mid \epsilon)}_{R_{ij}^{k-1}} = 0^{*} \\ R_{01}^{0} &= (0 \mid \epsilon)(0 \mid \epsilon)^{*}1 \mid 1 = 0^{*}1 \\ R_{10}^{0} &= 0(0 \mid \epsilon)^{*}(0 \mid \epsilon) \mid 0 = 00^{*} \\ R_{11}^{0} &= 0(0 \mid \epsilon)^{*}1 \mid \epsilon = 00^{*}1 \mid \epsilon \end{aligned}$$

Kleene's Algorithm: Example 1

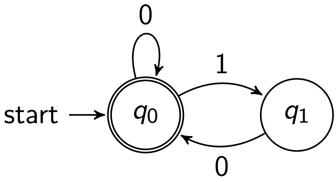


$$R_{ij}^{k} = R_{ik}^{k-1}\left(R_{kk}^{k-1}\right)^{*} R_{kj}^{k-1} \mid R_{ij}^{k-1}$$

$$\begin{aligned}
R_{00}^{0} &= (0 \mid \epsilon)(0 \mid \epsilon)^{*}(0 \mid \epsilon) \mid (0 \mid \epsilon) = 0^{*} \\
R_{01}^{0} &= (0 \mid \epsilon)(0 \mid \epsilon)^{*}1 \mid 1 = 0^{*}1 \\
R_{10}^{0} &= 0(0 \mid \epsilon)^{*}(0 \mid \epsilon) \mid 0 = 00^{*} \\
R_{11}^{0} &= 0(0 \mid \epsilon)^{*}1 \mid \epsilon = 00^{*}1 \mid \epsilon
\end{aligned}$$

$$R_{00}^{1} = 0^{*}1(00^{*}1 \mid \epsilon)^{*}00^{*} \mid 0^{*} = 0^{*}1(00^{*}1)^{*}00^{*} \mid 0^{*} =$$

Kleene's Algorithm: Example 1



$$R_{ij}^{k} = R_{ik}^{k-1}\left(R_{kk}^{k-1}\right)^{*} R_{kj}^{k-1} \mid R_{ij}^{k-1}$$

$$\begin{aligned}
R_{00}^{0} &= (0 \mid \epsilon)(0 \mid \epsilon)^{*}(0 \mid \epsilon) \mid (0 \mid \epsilon) = 0^{*} \\
R_{01}^{0} &= (0 \mid \epsilon)(0 \mid \epsilon)^{*}1 \mid 1 = 0^{*}1 \\
R_{10}^{0} &= 0(0 \mid \epsilon)^{*}(0 \mid \epsilon) \mid 0 = 00^{*} \\
R_{11}^{0} &= 0(0 \mid \epsilon)^{*}1 \mid \epsilon = 00^{*}1 \mid \epsilon
\end{aligned}$$

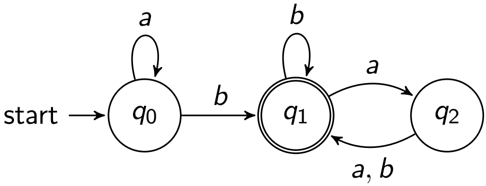
$$R_{00}^{1} = 0^{*}1(00^{*}1 \mid \epsilon)^{*}00^{*} \mid 0^{*} = 0^{*}1(00^{*}1)^{*}00^{*} \mid 0^{*} = (0^{*}100^{*})^{*}$$

Kleene's Algorithm: from FSA to Regular Expression

Description: Given an FSA $M = (Q, A, \delta, q_0, F)$ with $Q = \{q_0, \ldots, q_n\}$,

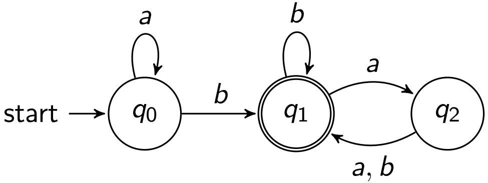
- R_{ij}^k are the sets of all strings that take M from state q_i to q_j without going through any state numbered lower than k,
- each set R_{ij}^k is represented by a regular expression,
- the algorithm computes R_{ij}^k step by step for $k = -1, 0, \ldots, n$,
- since there is no state numbered higher than n, the regular expression R_{0j}^n represents the set of all strings that take M from its start state q_0 to q_j.
  - If $F = \{q_{i_1}, \ldots, q_{i_f}\}$ is the set of accept states, the regular expression $R_{0i_1}^n \mid \ldots \mid R_{0i_f}^n$ represents the language accepted by M.

Kleene's Algorithm: Example 2 (-1)



$$
\begin{array}{lll}
R_{00}^{-1} = a \mid \epsilon & R_{10}^{-1} = \emptyset & R_{20}^{-1} = \emptyset \\
R_{01}^{-1} = b & R_{11}^{-1} = b \mid \epsilon & R_{21}^{-1} = a \mid b \\
R_{02}^{-1} = \emptyset & R_{12}^{-1} = a & R_{22}^{-1} = \epsilon
\end{array}
$$

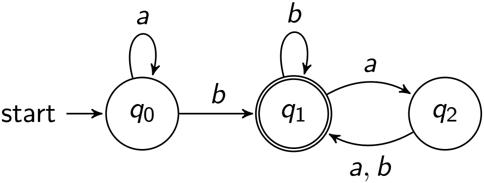
Kleene's Algorithm: Example 2 (0)



$$
\begin{array}{lll}
R^0_{00} = a^* & R^0_{10} = \emptyset & R^0_{20} = \emptyset \\
R^0_{01} = a^*b & R^0_{11} = b \mid \epsilon & R^0_{21} = a \mid b \\
R^0_{02} = \emptyset & R^0_{12} = a & R^0_{22} = \epsilon
\end{array}
$$

$$R^k_{ij} = R^{k-1}_{ik} \left(R^{k-1}_{kk} \right)^* R^{k-1}_{kj} \mid R^{k-1}_{ij}$$

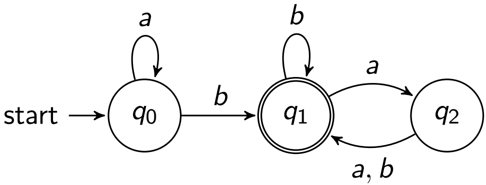
Kleene's Algorithm: Example 2 (1)



$$
\begin{array}{lll}
R^1_{00} = a^* & R^1_{10} = \emptyset & R^1_{20} = \emptyset \\
R^1_{01} = a^* bb^* & R^1_{11} = b^* & R^1_{21} = (a \mid b)b^* \\
R^1_{02} = a^* bb^* a & R^1_{12} = b^* a & R^1_{22} = (a \mid b)b^* a \mid \epsilon
\end{array}
$$

$$R^k_{ij} = R^{k-1}_{ik} \left(R^{k-1}_{kk}\right)^* R^{k-1}_{kj} \mid R^{k-1}_{ij}$$

Kleene's Algorithm: Example 2 (2)

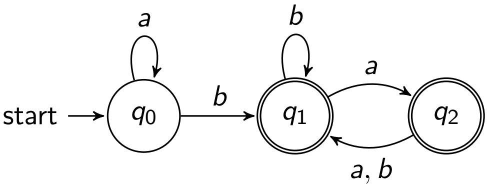


$$
\begin{array}{llllll}
R^1_{00} = & a^* & R^1_{10} = & \emptyset & R^1_{20} = & \emptyset \\
R^1_{01} = & a^*bb^* & R^1_{11} = & b^* & R^1_{21} = & (a \mid b)b^* \\
R^1_{02} = & a^*bb^*a & R^1_{12} = & b^*a & R^1_{22} = & (a \mid b)b^*a \mid \epsilon
\end{array}
$$

$$R^2_{01} = a^*bb^*a((a \mid b)b^*a \mid \epsilon)^*(a \mid b)b^* \mid a^*bb^* =$$

$$= a^*bb^*(a(a \mid b)b^*)^*$$

Kleene's Algorithm: Example 3



$R_{01}^2 = a^*bb^*(a(a \mid b)b^*)^*$
$R_{02}^2 = a^*bb^*(a(a \mid b)b^*)^*a$

Answer: $R_{01}^2 \mid R_{02}^2 = a^*bb^*(a(a \mid b)b^*)^*(\epsilon \mid a)$

Agenda

Non-determinism:

- NDFSA = DFSA
  - RegExp → NDFSA → DFSA (the previous week)
  - DFSA → RegExp (Kleene's Algorithm)
  - **Chomsky Grammars Hierarchy (Regular grammars)**
- NDPDA
  - Definition & Example
  - Chomsky Grammars Hierarchy (Context-Free grammars)
- TM
  - Definition & Example
  - Chomsky Grammars Hierarchy (Unrestricted grammars)

Grammar: definition

Definition

A grammar is a tuple

$$\langle V_N, V_T, P, S \rangle$$

where

- V_N is the non-terminal alphabet;
- V_T is the terminal alphabet;
- $P \subseteq (V^* \cdot V_N \cdot V^*) \times V^*$ is the (finite) set of rewriting rules of production, where $V = V_N \cup V_T$;
- $S \in V_N$ is a particular element called axiom or initial symbol.

Regular grammars (type 3)

Right regular grammar

A right regular grammar is a formal grammar $\langle V_N, V_T, P, S\rangle$ such that all the production rules in P are of one of the following forms:

1) $A \rightarrow s$, where $A \in V_N$ and $s \in V_T^*$;
2) $A \rightarrow sB$, where $A, B \in V_N$ and $s \in V_T^*$;

Left regular grammar

2*) $A \rightarrow Bs$, where $A, B \in V_N$ and $s \in V_T$;

$A \rightarrow a_1 A'_1, A'_1 \rightarrow a_2 A'_2, \ldots, A'_{k-1} \rightarrow a_k A'_k, A'_k \rightarrow \epsilon B \leftrightharpoons A \rightarrow sB$, where $s = a_1 a_2 \ldots a_k$.

Regular grammars (type 3)

Example 1

$$L_1 = \{(ab)^n \mid n \in \mathbb{N}\}$$

Rules

$$S \to \epsilon A$$
$$A \to abA$$
$$A \to \epsilon$$

$$S \to \epsilon A \to \epsilon\epsilon = \epsilon$$
$$S \to \epsilon A \to \epsilon abA \to \epsilon ab\epsilon = ab$$
$$S \to \epsilon A \to abA \to ababA \to \cdots (ab)^n A \to (ab)^n$$

Regular grammars (type 3)

Example 2

$$L_2 = \{xaay \mid x, y \in \{a, b\}^*\}$$

Rules

$$\begin{array}{lll} S \rightarrow \epsilon A & A \rightarrow aA & B \rightarrow aB \\ & A \rightarrow bA & B \rightarrow aB \\ & A \rightarrow aaB & B \rightarrow \epsilon \end{array}$$

Regular grammars (type 3)

Fact

$$DFSA = NFSA = RG$$

Agenda

Non-determinism:

- NDFSA $=$ DFSA
  - RegExp $\rightarrow$ NDFSA $\rightarrow$ DFSA (the previous week)
  - DFSA $\rightarrow$ RegExp (Kleene's Algorithm)
  - Chomsky Grammars Hierarchy (Regular grammars)
- NDPDA
  - **Definition & Example**
  - Chomsky Grammars Hierarchy (Context-Free grammars)
- TM
  - Definition & Example
  - Chomsky Grammars Hierarchy (Unrestricted grammars)

Non-deterministic Pushdown Automaton (NDPDA)

Definition: NDPDA

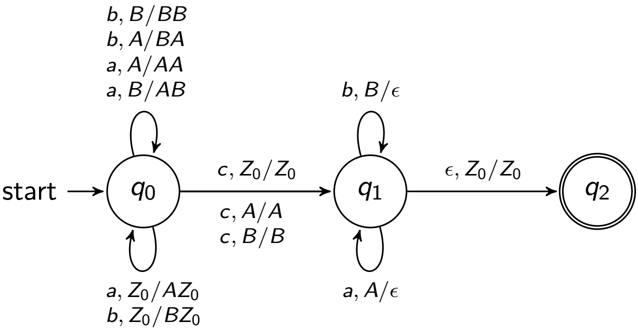
A NDPDA is a tuple $\langle Q, I, \Gamma, \delta, q_0, Z_0, F \rangle$, where $Q, I, \Gamma, q_0, Z_0, F$ are defined as in (D)PDA and the transition function is defined as

$$\delta : Q \times (I \cup \{\epsilon\}) \times \Gamma \rightarrow \mathbb{P}_{\mathtt{F}}(Q \times \Gamma^*)$$

where $\mathbb{P}_{\mathtt{F}}$ indicates finite subsets.

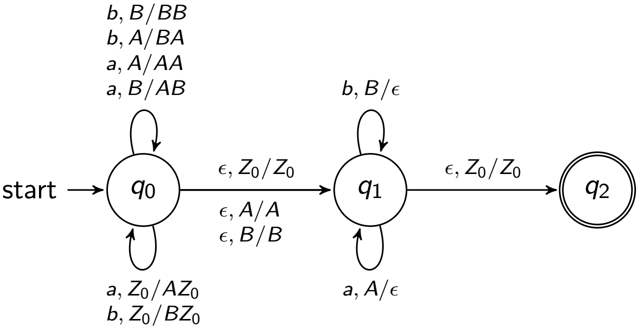
Deterministic PDA

$$L_1 = \{wcw^R \mid w \in \{a,b\}^*\}$$



NDPDA

$$L_2 = \{ww^R \mid w \in \{a, b\}^*\}$$



Agenda

Non-determinism:

- NDFSA $=$ DFSA
  - RegExp $\rightarrow$ NDFSA $\rightarrow$ DFSA (the previous week)
  - DFSA $\rightarrow$ RegExp (Kleene's Algorithm)
  - Chomsky Grammars Hierarchy (Regular grammars)
- NDPDA
  - Definition & Example
  - **Chomsky Grammars Hierarchy (Context-Free grammars)**
- TM
  - Definition & Example
  - Chomsky Grammars Hierarchy (Unrestricted grammars)

Context-Free grammars (type 2)

Regular grammar (Recall)

1) $A \to s$, where $A \in V_N$ and $s \in V_T^*$;
2) $A \to sB$, where $A, B \in V_N$ and $s \in V_T^*$;

Context-Free grammar

A Context-Free grammar is a formal grammar $\langle V_N, V_T, P, S \rangle$ such that all the production rules in P are the following forms:

$$A \to \beta,$$

where $A \in V_N$, $\beta \in (V_T \cup V_N)^*$.

Context-Free grammars (type 2)

Example 1

$$L_1 = \{a^n b^n \mid n \in \mathbb{N}\}$$

Rules

$$S \to \epsilon$$
$$S \to aSb$$

$$S \to \epsilon$$
$$S \to aSb \to ab$$
$$S \to aSb \to aaSbb \to \cdots a^n S b^n \to a^n b^n$$

Context-Free grammars (type 2)

Example 2

$$L_2 = \{ww^R \mid w \in \{a, b\}^*\}$$

Rules

$$S \to \epsilon$$
$$S \to aSa$$
$$S \to bSb$$

$$S \to aSa \to abSba \to abbSbba \to \cdots$$

Context-Free grammars (type 2)

Fact

$$DPDA \subsetneq NDPDA = CFG$$

Agenda

Non-determinism:

- NDFSA = DFSA
  - RegExp $\rightarrow$ NDFSA $\rightarrow$ DFSA (the previous week)
  - DFSA $\rightarrow$ RegExp (Kleene's Algorithm)
  - Chomsky Grammars Hierarchy (Regular grammars)
- NDPDA
  - Definition & Example
  - Chomsky Grammars Hierarchy (Context-Free grammars)
- TM
  - **Definition** & **Example**
  - Chomsky Grammars Hierarchy (Unrestricted grammars)

Turing Machine

Formal Definition

A Turing Machine (TM) with k-tapes is a tuple

$$T = \langle Q, I, \Gamma, \delta, q_0, Z_0, F \rangle$$

where

Q is a finite set of states;
I is the input alphabet;
Γ is the memory alphabet;
δ is the transition function;
$q_0 \in Q$ is the initial state;
$Z_0 \in \Gamma$ is the initial memory symbol;
$F \subseteq Q$ is the set of final states.

Deterministic & Non-Deterministic TM

Deterministic:

$$\delta : (Q - F) \times (I \cup \{\_\}) \times (\Gamma \cup \{\_\})^k \to Q \times (\Gamma \cup \{\_\})^k \times \{R, L, S\}^{k+1}$$

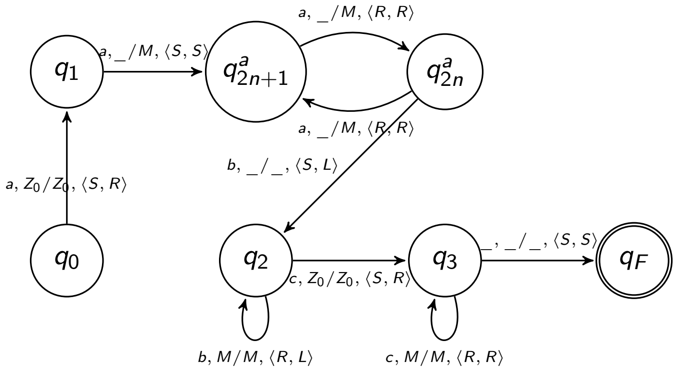
Definition: Non-Deterministic TM (NDTM)

A NDTM is a tuple $\langle Q, I, \Gamma, \delta, q_0, Z_0, F \rangle$, where $Q, I, \Gamma, q_0, Z_0, F$ are defined as in (D)TM and the transition function is defined as

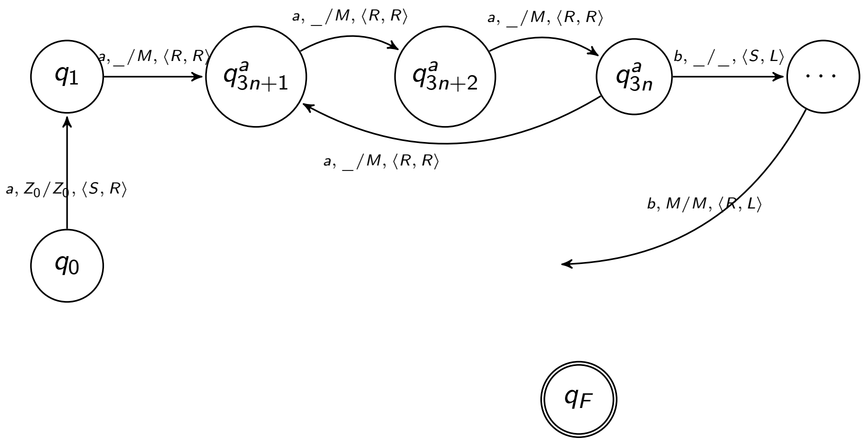
$$\delta : (Q - F) \times (I \cup \{\_\}) \times (\Gamma \cup \{\_\})^k \to \mathbb{P}_F \left(Q \times (\Gamma \cup \{\_\})^k \times \{R, L, S\}^{k+1} \right.$$

Example

The TM_1 recognises the language $L_1 = \{a^{2n}b^{2n}c^{2n} \mid n > 0\}$

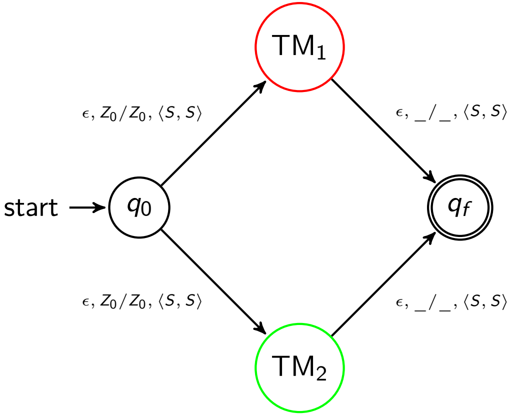


The TM_2 recognises the language $L_2 = \{a^{3n}b^{3n}c^{3n} \mid n > 0\}$



Example

The TM recognises the language
$L = \{a^{2n}b^{2n}c^{2n} \mid n > 0\} \cup \{a^{3n}b^{3n}c^{3n} \mid n > 0\}$



Agenda

Non-determinism:

- NDFSA $=$ DFSA
  - RegExp $\rightarrow$ NDFSA $\rightarrow$ DFSA (the previous week)
  - DFSA $\rightarrow$ RegExp (Kleene's Algorithm)
  - Chomsky Grammars Hierarchy (Regular grammars)
- NDPDA
  - Definition & Example
  - Chomsky Grammars Hierarchy (Context-Free grammars)
- TM
  - Definition & Example
  - **Chomsky Grammars Hierarchy (Unrestricted grammars)**

Unrestricted grammars (type 0)

Unrestricted grammar

A Unrestricted grammar is a formal grammar $\langle V_N, V_T, P, S \rangle$ such that all the production rules in P are the following forms:

$$\alpha \to \beta,$$

where $\alpha, \beta \in (V_T \cup V_N)^*$.

$AaB \to baC$

Unrestricted grammars (type 0)

Fact

$$NDTM = URG$$

Question

What about $DTM = NDTM$?

Thank you for your attention!