Theoretical computer science

Tutorial - week 11

April 1, 2021

INNOPOLIS UNIVERSITY

Agenda

- Non-determinism:
  - NDFSA to DFSA
  - PDA
  - TM

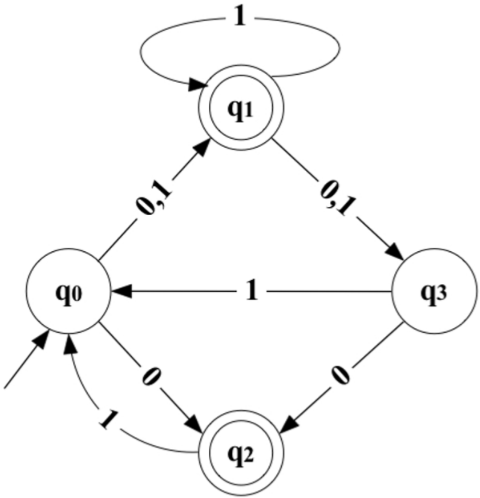
NDFSA to DFSA

Algorithm for NDFSA to DFSA

1. Create state table from the given NDFA
2. Create a blank state table under possible input alphabets for the equivalent DFA
3. Mark the start state of the DFA by q_0 (Same as the NDFA)
4. Find out the combination of States $q_0, q_1, ..., q_n$ for each possible input alphabet
5. Each time we generate a new DFA state under the input alphabet columns, we have to apply step 4 again, otherwise go to step 6
6. The states which contain any of the accepting states of the NDFA are the accepting states of the equivalent DFA

NDFSA to FSA: example

Let us consider the following NDFSA:



NDFSA to FSA: step 1

First, we build a transition table for NDFSA:

q	δ(q,0)	δ(q,1)
→q0	{q1,q2}	{q1}
*q1	{q3}	{q1, q3}
*q2	∅	{q0}
q3	{q2}	{q0}

NDFSA to FSA: step 2

Using table from previous slide, let us create a similar table, but this time for FSA. Initially, the table is empty:

q	δ(q,0)	δ(q,1)
...	...	...

NDFSA to FSA: step 3

We begin by adding the initial state and the set of states:

q	δ(q,0)	δ(q,1)
$\rightarrow$q0	{q1,q2}	{q1}
...	...	...

NDFSA to FSA: step 4

The initial state can take us to two new states that are not yet in the table. Note that we treat a set of states as a single state now!

q	δ(q,0)	δ(q,1)
→q0	{q1,q2}	{q1}
{q1,q2}	...	...
q1	...	...
...	...	...

The next step is to find possible states for $\{q1, q2\}$ and $q3$. For $q3$ it is trivial, you just need to look it up in the original NDFSA table. However, the transition for $\{q1, q2\}$ will be a union of sets:
$\delta(q1, q2, 0) = \delta(q1, 0) \cup \delta(q2, 0) = \{q3\}$
$\delta(q1, q2, 1) = \delta(q1, 1) \cup \delta(q2, 1) = \{q0, q1, q3\}$

NDFSA to FSA: step 5

The initial state can take us to two new states that are not yet in the table. Note that we treat a set of states as a single state now!

q	δ(q,0)	δ(q,1)
$\rightarrow$q0	{q1,q2}	{q1}
{q1,q2}	{q3}	{q0,q1,q3}
q1	...	...
...	...	...

The next step is to find possible states for $\{q1, q2\}$ and $q3$. For $q3$ it is trivial, you just need to look it up in the original NDFSA table. However, the transition for $\{q1, q2\}$ will be a union of sets:
$\delta(q1, q2, 0) = \delta(q1, 0) \cup \delta(q2, 0) = \{q3\}$
$\delta(q1, q2, 1) = \delta(q1, 1) \cup \delta(q2, 1) = \{q0, q1, q3\}$

NDFSA to FSA: step 5

Repeat the steps above until we have included all states from the original NDFSA and there are no new states.

q	δ(q,0)	δ(q,1)
$\rightarrow$q0	{q1,q2}	{q1}
{q1,q2}	{q3}	{q0,q1,q3}
q1	{q3}	{q1,q3}
...	...	...

NDFSA to FSA: step 6

After repeating previous steps and depicting final states (step 6) we will arrive to the following table:

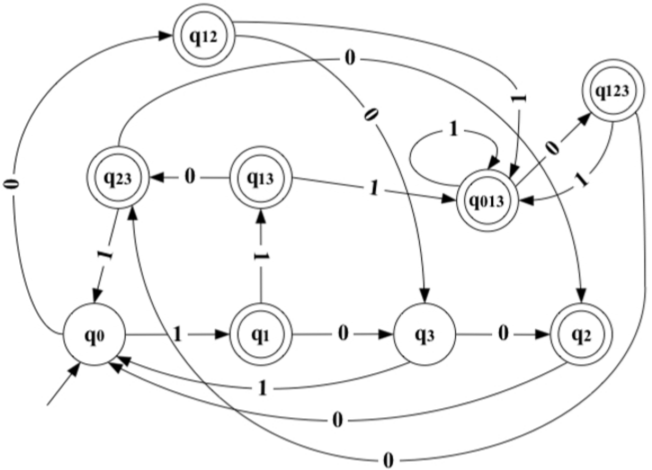
q	δ(q,0)	δ(q,1)
$\rightarrow$q0	{q1,q2}	{q1}
*{q1,q2}	{q3}	{q0,q1,q3}
*q1	{q3}	{q1,q3}
q3	{q2}	{q0}
*{q0,q1,q3}	{q1,q2,q3}	{q0,q1,q3}
*{q1,q3}	{q2,q3}	{q0,q1,q3}
*q2	$\varnothing$	{q0}
*{q1,q2,q3}	{q2,q3}	{q0,q1,q3}
*{q2,q3}	{q2}	{q0}

NDFSA to FSA: result (table representation)

q	δ(q,0)	δ(q,1)
$\rightarrow$q0	{q1,q2}	{q1}
*q12	{q3}	{q0,q1,q3}
*q1	{q3}	{q1,q3}
q3	{q2}	{q0}
*q013	{q1,q2,q3}	{q0,q1,q3}
*q13	{q2,q3}	{q0,q1,q3}
*q2	$\varnothing$	{q0}
*q123	{q2,q3}	{q0,q1,q3}
*q23	{q2}	{q0}

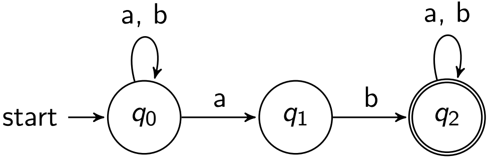
NDFSA to FSA: result (graphical representation)

Finally, we can build the resulting DFSA:



Example

Build DFSA from the NDFSA that recognizes the language:
$L_1 = \{x \in \{a, b\}^* \mid \ x \text{ contains ab}\}$



Deterministic PDA: recap

A Deterministic PDA – Formal Definition

A Deterministic Pushdown Automaton (DPDA)

A PDA $M = \langle Q, I, \Gamma, \delta, q_0, Z_0, F \rangle$ is deterministic if it satisfies both of the following conditions.

1. For every $q \in Q$, every $x \in I \cup \{\epsilon\}$, and every $\gamma \in \Gamma$, the set $\delta(q, x, \gamma)$ has at most one element.
2. For every $q \in Q$, every $x \in I$, and every $\gamma \in \Gamma$, the two sets $\delta(q, x, \gamma)$ and $\delta(q, \epsilon, \gamma)$ cannot both be non-empty.

Transition

Transitions between configurations ($\vdash$) depend on the transition function. It is the way to commute from a PDA snapshot to another.
There are 2 cases:

1. The transition function is defined for an input symbol.
2. The transition function is defined for an ϵ move.

Transition – Case 1

If $(q', \alpha) \in \delta(q, i, A)$ then

$$(q, x, \gamma) \vdash (q', x', \gamma')$$

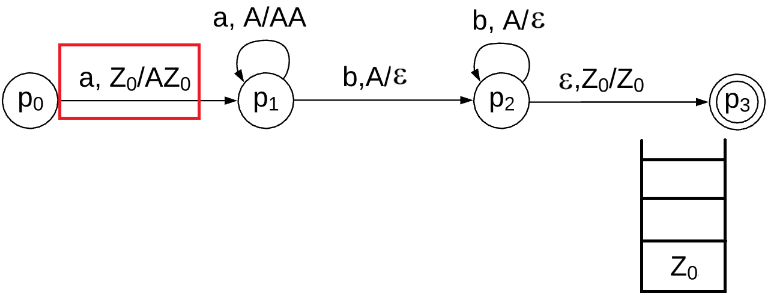
where (old snapshot)

- q is the current state
- $x = iy$
- $\gamma = A\beta$ (for some $\beta \in \Gamma^*$)

then (new snapshot)

- q' is the new state
- $x' = y$
- $\gamma' = \alpha\beta$

Transition – Case 1 (Graphical representation)



Transition – Case 2

If $(q', \alpha) \in \delta(q, \epsilon, A)$ then

$$(q, x, \gamma) \vdash (q', x', \gamma')$$

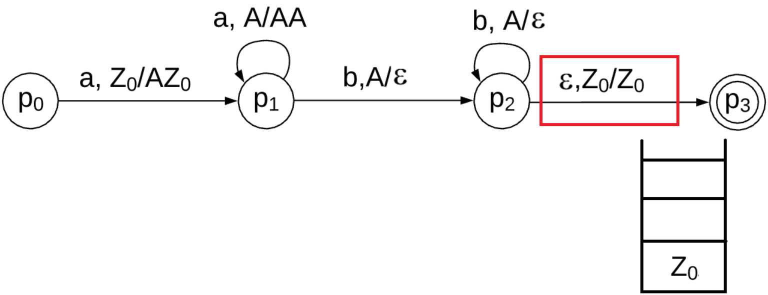
where (old snapshot)

- q is the current state
- $\gamma = A\beta$ (for some $\beta \in \Gamma^*$)

then (new snapshot)

- q' is the new state
- $x' = x$
- $\gamma' = \alpha\beta$

Transition – Case 2 (Graphical representation)



Non-deterministic PDA.

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

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

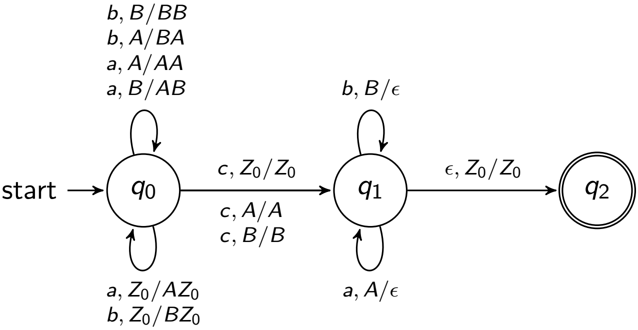
where $\mathbb{P}_{\mathtt{F}}$ indicates finite subsets.

Example: wcw^R

Detirministic PDA can accept $L_1 = \{wcw^R \mid w \in \{a, b\}^*\}$ where w^R is the reversed string w.

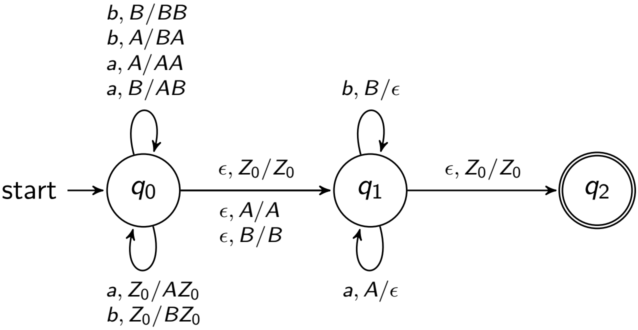
Example: wcw^R

Detirministic PDA can accept $L_1 = \{wcw^R \mid w \in \{a, b\}^*\}$ where w^R is the reversed string w.



Example: ww^R

NDPDA accepting $L_1 = \{ww^R \mid w \in \{a,b\}^*\}$ where w^R is the reversed string w.



Non-deterministic TM.

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.

Transition Function for Deterministic TM

The transition function for Deterministic TM

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

where elements of $\{R, L, S\}$ indicate "directions" of the head of the TM:

R : move the head one position to the right;
L : move the head one position to the left;
S : stand still.

Remarks:

- the transition function can be partial;
- no transition outgoing from the final states;
- the symbol $\_ \notin \Gamma \cup I$ is a special blank symbol on the tapes.

Transition Function for Non-Deterministic TM

To define a NDTM, we need to change the transition function (all the other elements remain as in a (D)TM):

Definition: 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

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

Acceptance: Among the various possible runs (with the same input) of the NDTM, it is sufficient that one of them succeeds to accept the input string.

Wrap up

- What have you learnt today?

Wrap up

- What have you learnt today?
- What for this could be useful?