San José State University
Department of Computer Science

Ahmad Yazdankhah
ahmad.yazdankhah@sjsu.edu
www.cs.sjsu.edu/~yazdankhah

Turing Machines

(Part 2)

Lecture 16

Day 17/31

CS 154

Formal Languages and Computability

Spring 2018

Agenda of Day 17

- About Teams
- Summary of Lecture 15
- Lecture 16: Teaching ...
  - Turing Machines (Part 2)
- Quiz 6 (Take-Home Exam)

Summary of Lecture 15: We learned ...

Standard Turing Machines (TMs)

- NPDAs are unable to accept some languages like $a^n b^n c^n$ and ww.
- The limitation of NPDAs is ...
  - ... we lose some data when we access the older data.
  - ... so, stack is not so flexible in storing and retrieving data.
- We introduced standard Turing machines to overcome this limitation.
- Standard TMs are deterministic.

- The main difference between TMs and NPDAs is ...
  - ... we have the ability to move the read/write head to the left or right.
- The transition condition of TMs is ...
  - ... input symbol.
- TMs halt iff ...
  - ... they have zero transition.

Any Question

Summary of Lecture 15: We learned ...

TMs

- The criteria of accepting strings for previous machines are ...

$$(h \wedge c \wedge f) \leftrightarrow a$$

- Consuming all input symbols is meaningless for TMs.
- So, theoretically, the logical representation of accepting strings is ...

$$(h \wedge f) \leftrightarrow a$$

- And for rejecting strings is ...

$$(\sim h \vee \sim f) \leftrightarrow \sim a$$

- But in practice, it is important to note that ...
- ... that is the TMs designers responsibility to make sure that the machine halts in an accepting state when all symbols are consumed.
- Otherwise, it should halts in a non-accepting state.

Any Question

Summary of Lecture 15: We learned ...

TMs

- For the first time, we observed a new phenomenon that happens in Turing machines ...
  - ... Infinite loops!
- This phenomenon never happened in the previous deterministic machines.
- This is the consequence of ...

  ... having freedom of moving the read-write head to the left or right.

- Recall that if a TM is in infinite-loop, the string it is processing, is considered as "rejected".
- When a machine is working for a long time, from an observer's point of view, ...
  - ... is it in an infinite loop? OR
  - ... it is in the middle of a very long computation?
- There is no known algorithm to answer this!

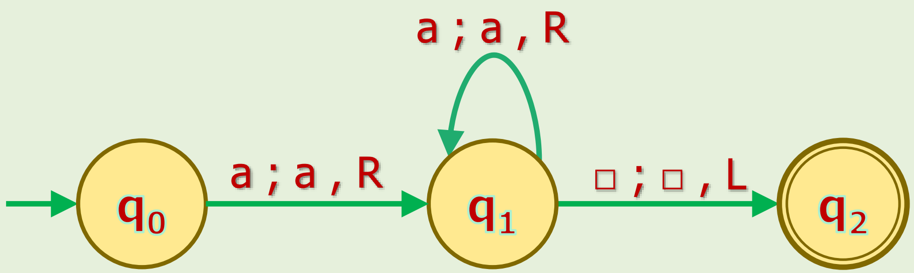
Any Question

TMs Design Examples

TMs Design Examples

Example 7

- Design a TM to accept $L = \{a^n : n \geq 1\}$ over $\Sigma = \{a, b\}$.
- Note that TMs don't like λ!

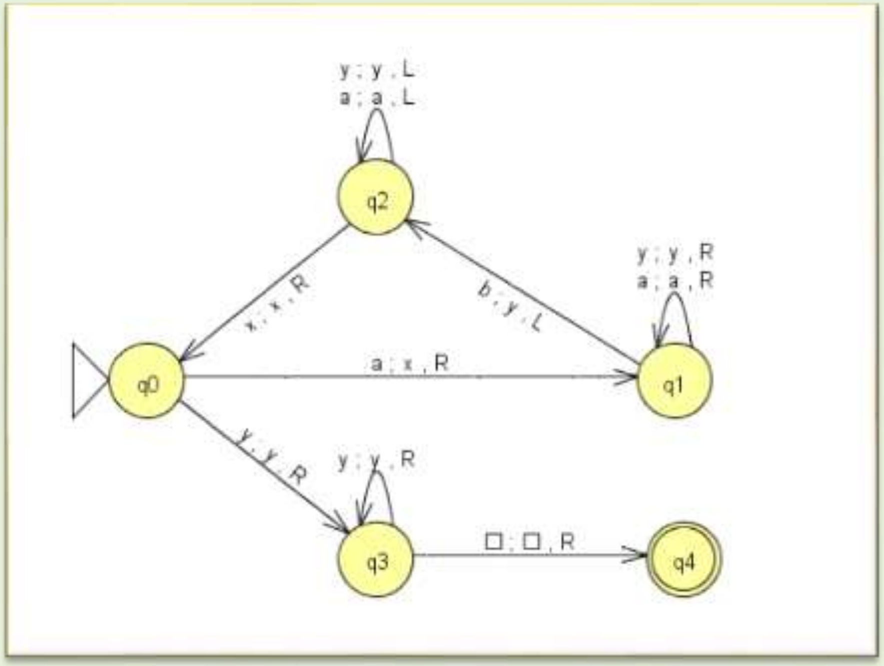


TMs Design Examples

Example 8

- Design a TM to accept our famous language $L = \{a^n b^n : n \geq 1\}$ over $\Sigma = \{a, b\}$.

Solution



- **Strategy**: For every a's, you should find one "b". So, we read the first "a" and mark it as read by replacing it to "x". Then we go right to find a corresponding "b" and mark it as "y".
We continue this process until we don't have any a's.
The string is accepted if then is no "b" as well.

Homework

- Design a TM for the following languages:
  1. $L = \{w \in \{a, b\}^+\}$
  2. $L = \{w \in \{a, b\}^+ : |w| = 2k, K \geq 0\}$
  3. $L = \{w \in \{a, b\}^+ : |w| = 2k+1, K \geq 0\}$
  4. $L = \{1^{2k} : k \geq 1\}$ over $\Sigma = \{1\}$
  5. $L = \{w \in \{a, b\}^+ : n_a(w) = n_b(w)\}$ //number of a's = number of b's
  6. $L = \{a^n b^n c^n : n \geq 1\}$
  7. $L = \{a^n b^m c^{nm} : n \geq 1, m \geq 1\}$
  8. $L = \{w\#w : w \in \{a, b\}^+\}$
  9. $L = \{w \in \{a, b\}^+ : |w| = 2k+1, K \geq 0$, w contains at least one a$\}$
  10. $L = \{ww : w \in \{a, b\}^+\}$

6. Definitions

6. Formal Definition of TMs

- A standard TM M is defined by the septuple (7-tuple):

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

- Where:
  - Q is a finite and nonempty set of states of the transition graph.
  - Σ is a finite and nonempty set of symbols called input alphabet.
  - Γ is a finite and nonempty set of symbols called tape alphabet.
  - δ is called transition function and is defined as:

$$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

δ may be partial xor total function.

  - $q_0 \in Q$ is the initial state of the transition graph.
  - $\square \in \Gamma$ is a special symbol called blank.
  - $F \subseteq Q$ is the set of accepting states of the transition graph.

6. Formal Definition of TMs

$$M = \left(\overset{\text{Set of States}}{Q}\ ,\ \underset{\text{Input Alphabet}}{\Sigma}\ ,\ \overset{\text{Tape Alphabet}}{\Gamma}\ ,\ \underset{\text{Transition Function}}{\delta}\ ,\ \overset{\text{Initial State}}{q_0}\ ,\ \underset{\text{Blank Symbol}}{\square}\ ,\ \overset{\text{Set of Final States}}{F} \right)$$

6. Formal Definition of TMs: Notes

1. L and R are called "move symbols".
   - They indicate whether the read-write head moves one cell left or right, after the new symbol has been written.

2. $\Sigma \subseteq \Gamma - \{\square\}$.
   - So, the input string cannot contain blank symbol.

3. There is no relationship between "determinism" and "total function".
   - A machine whose transition function is partial can be deterministic as long as it does not violate the definition of determinism.

TMs Transition Function Examples

Example 9

- Write the sub-rule of the following transition.

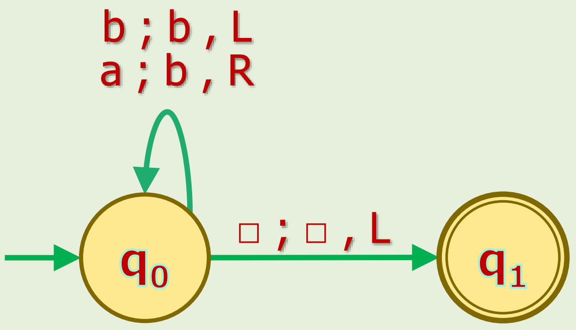
- $\delta(q_1, a) = (q_2, b, R)$

TMs Transition Function Examples

Example 10

- Write the δ of the following transition graph.

$$\delta:\begin{cases}\delta(q_0, a) = (q_0, b, R)\\ \delta(q_0, b) = (q_0, b, L)\\ \delta(q_0, \square) = (q_1, \square, L)\end{cases}$$



- Is the function total or partial?

7. TMs vs NPDAs

Can TMs Do Whatever NPDAs Can Do?

- Let's assume that we've constructed an NPDA for an arbitrary language L.
- Can we always construct a TM for L?

- To compare previous machines (i.e. DFAs, NFAS, NPDAs), we used the "formal definition conversion" technique .
- For this case, it is not so easy to do that.
- But there is another technique called "simulation".

- So, we convert the above question to:

Can we simulate NPDAs operations by TMs?

- Yes! How?

Can TMs Do Whatever NPDAs Can Do?

- Let M be an NPDA for the language L.
- We want to simulate M by an equivalent TM called M' such that:

  L(M) = L(M')

- The NPDA has several transitions and we should be able to simulate all of them by TM.
- We just show the simulation of one simple transition and leave the rest of them for the reader as exercise.

- I put the following document in Canvas for your reference:

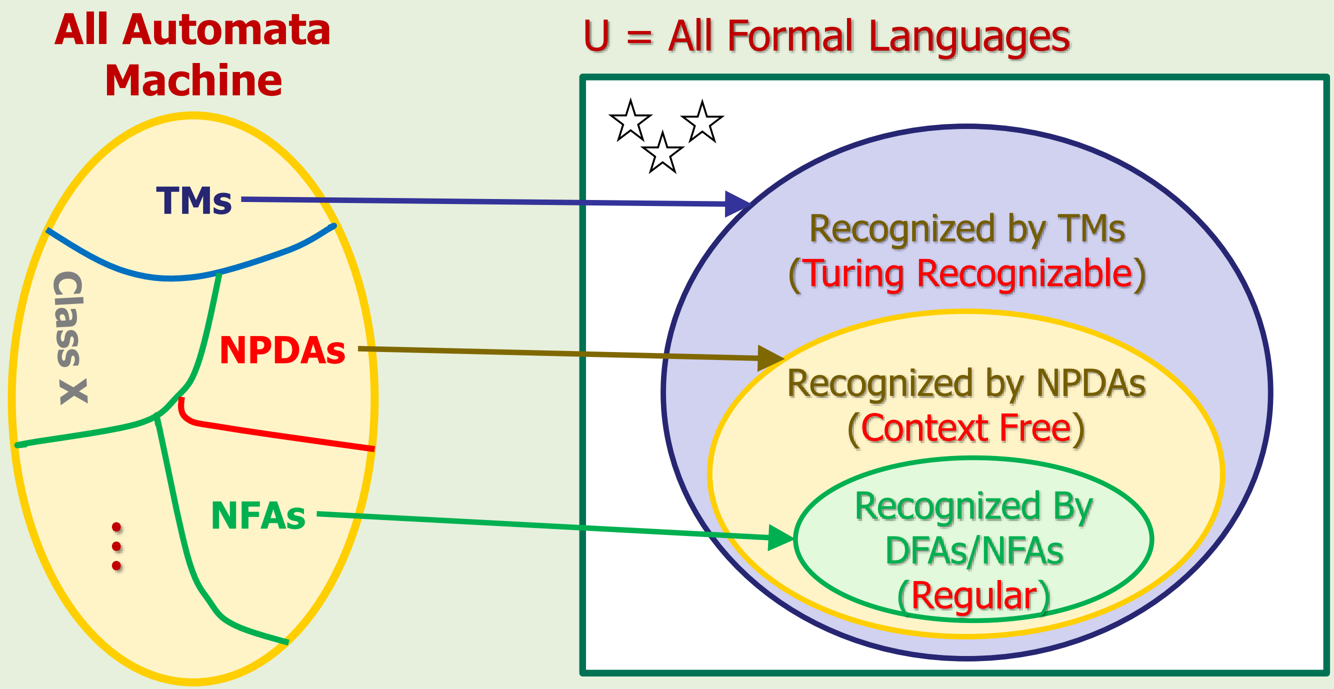
  Canvas → Files → Misc

  S18-Ahmad Y-CS154-NPDAs-Transition-Simulation.pdf

Can NPDAs Do Whatever TMs Can Do?

- Let's assume that we've constructed a TM for an arbitrary language L.
- Can we always construct an NPDA for L?
- No! Why?
- At least we know the following languages for which we constructed TMs but it was impossible to construct NPDAs.
  - $L = \{a^n b^n c^n : n \geq 1\}$
  - $L = \{ww : w \in \Sigma^*\}$
- Let's summarize our knowledge and figure out what would be the next step.

Machines and Languages Association

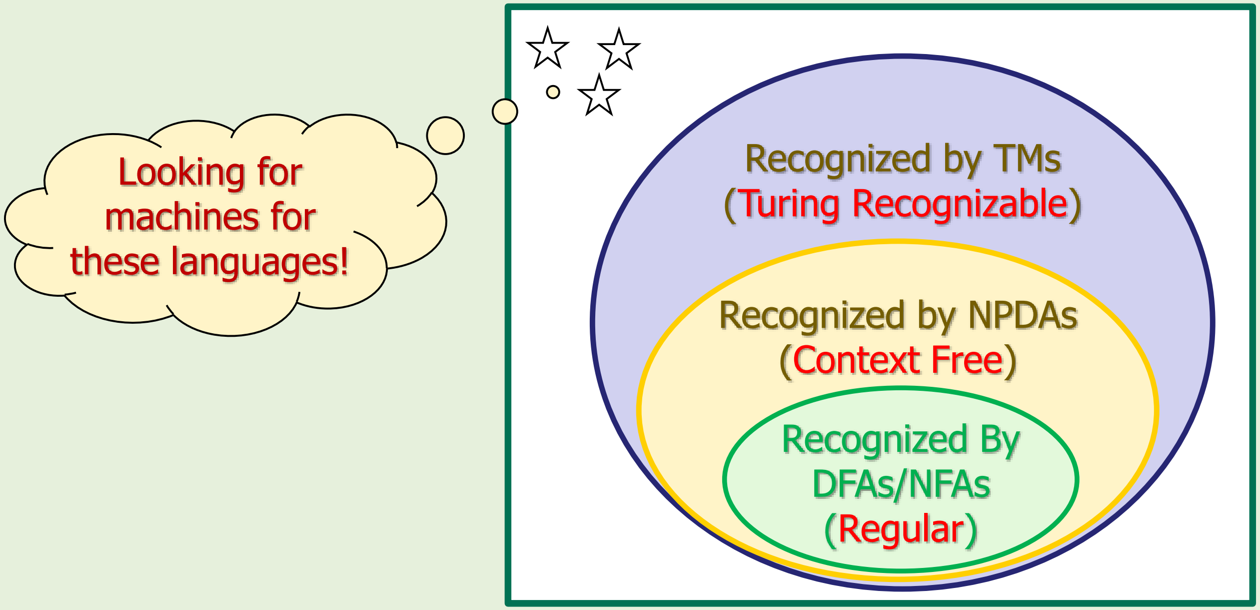


- The set of languages that NPDAs recognize is a proper subset of the set of languages that TMs recognize.
- So, TMs are more powerful than NPDAs.

8. What is the Next Step?

- TMs recognize some other non-regular languages called "Turing recognizable".
- But there are still languages that are not Turing recognizable!
- First, we need to find at least one of them, then we'll think about the next step!

U = All Formal Languages



NAME	Alan M. Turing		
SUBJECT	CS 154	TEST NO.	6
DATE	03/22/2018	PERIOD	1 , 2 , 3

TEST RECORD	
PART 1	123
PART 2	
TOTAL	

Your **list #** goes here!

Quiz 6

No Scantron

Take-Home Exam

Nice Videos

1. Turing machines explained visually
   https://www.youtube.com/watch?v=-ZS_zFg4w5k
2. A Turing machine – Overview
   https://www.youtube.com/watch?v=E3keLeMwfHY

References

1. Linz, Peter, "An Introduction to Formal Languages and Automata, 5th ed.," Jones & Bartlett Learning, LLC, Canada, 2012
2. Kenneth H. Rosen, "Discrete Mathematics and Its Applications, 7th ed.," McGraw Hill, New York, United States, 2012
3. Michael Sipser, "Introduction to the Theory of Computation, 3rd ed.," CENGAGE Learning, United States, 2013
   ISBN-13: 978-1133187790
4. Wikimedia Commons, https://commons.wikimedia.org/wiki/Category:Animations_of_machinery
5. https://en.wikipedia.org/wiki/Turing_Award
6. https://en.wikipedia.org/wiki/Alan_Turing