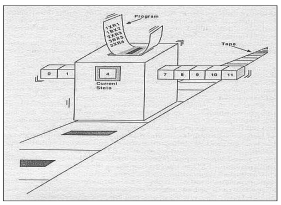
# CS 302 Lecture 15
# Turing Machine Robustness; Nondeterministic Turing Machines; Unrestricted Grammars

Qi Mi
Computer Science Department
University of Virginia



## Recap the Turing machines

- 7-tuple: $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$



- Possible modifications to Turing machines?



## TM with a different alphabet size

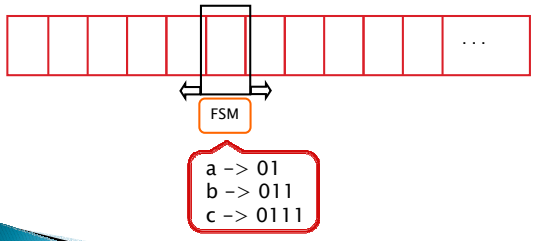
- Consider a Turing machine with an input alphabet of {a, b, c} and another with an input alphabet of {0, 1}. Which is more powerful?

$| \Sigma |$



## TM with a different alphabet size

- Idea: Use FSM to translate the encoding between different alphabets.






## TM with a different alphabet size

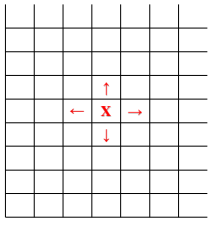
- Encoding
  - A process of transforming information from one format into another without loss of information.
- Example:
  - Binary representation of numbers
- Application:
  - Adding marker symbols that are not in the original alphabet when you design a TM will not change the power of TM.



## TM with a multidimensional tape

- A 2-dimensional tape

  {L, R, U, D}



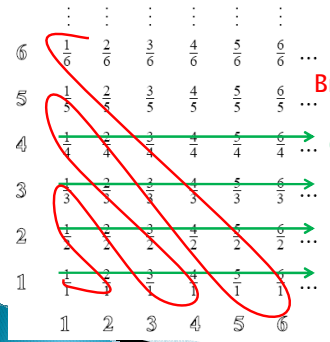

- Question: Is a TM with a 2-dimensional tape equivalent to one with an ordinary 1-dimensional tape?

## TM with a multidimensional tape

- *The set of rational numbers:*
  $Q = \{ p/q \mid p$ and $q$ are natural numbers and co-prime$\}$
- *The set of natural numbers:*
  $N = \{1, 2, 3, 4, 5, \ldots\}$
- True or False: $|Q| > |N|$?



## TM with a multidimensional tape



$|Q| = |N|$

Breadth-first search, instead of depth-first search

Dovetailing



## TM with a multidimensional tape

- Question: Recall that adjacent cells may become non-adjacent when we map a 2-dimensional tape to a 1-dimensional tape. How do we solve the issue of mapping the head movement between adjacent cells on a 2-dimensional tape to that on a 1-dimensional tape?



## TM with a multidimensional tape

- Map a 2-dimensional tape to an ordinary 1-dimensional tape.
- Map a k-dimensional tape to an ordinary 1-dimensional tape.
- Summary:
  - Dovetailing (interleaving)
  - Mapping (1-to-1 correspondence)



## Turing machine modifications

- A different alphabet size
- Multidimensional tape
- Doubly-infinite tape
- Multiple tapes
- Etc

Theorem: All these modifications do NOT increase the power of TM's. -- TM robustness

Question: What if a combination of the above?



## Designing Turing machines

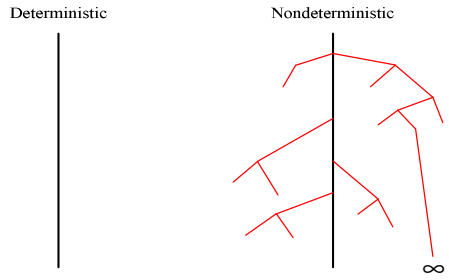
- Task: Design a Turing machine that can recognize $\{ww | w \in \Sigma^*\}$?

## Non-deterministic Turing machines

- A Turing machine is deterministic if:
  $\forall\ q \in Q,\ a \in \Gamma \quad |\delta(q,a)| \leq 1$
  i.e., no multiple choices allowed
- Otherwise, it is non-deterministic.
- A non-deterministic TM (NDTM) can have several choices of which state to proceed next in a computation.
- Many "next-moves":
  $\delta: Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R\}}$

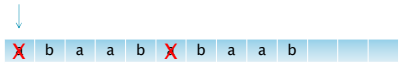


## Non-deterministic Turing machines



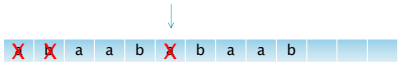



## Designing Turing machines






## Designing Turing machines






## Non-deterministic Turing machines

- Question: Is the set of languages that can be decided by NDTM's larger than that by DTM's?



## Non-deterministic Turing machines

- Simulate any non-deterministic TM N with a deterministic TM D.
- Three tapes: input tape, simulation tape, and address tape
- Have D try all possible branches of N using breadth-first search. (can't use depth-first search here)
- Conclusion: NDTMs and DTMs are equivalent in power.

## Unrestricted grammars

- An unrestricted grammar is a 4-tuple $G = (V, \Sigma, R, S)$ where
  - V is a finite set of variables
  - Σ (the alphabet) is a finite set of terminal symbols
  - R is the finite set of rules. Each rule is of the form $\alpha \rightarrow \beta$, where $\alpha \in (V \cup \Sigma)^+$ and $\beta \in (V \cup \Sigma)^*$
  - $S \in V$ is the start symbol.
- V and Σ are assumed to be disjoint.



## Unrestricted grammars

- In an unrestricted grammar (a.k.a. general grammar), the left hand side can include extra terminals and non-terminals.
  - Example: aSb → Tc
- Left hand side must include at least one non-terminal.



## Unrestricted grammars

- Example:
  A grammar that generates $\{a^i b^i c^i \mid i \geq 0\}$.
  $G = (V, \Sigma, R, S)$ where $V = \{S, A, C\}$, $\Sigma = \{a, b, c\}$
  $R = \{ S \rightarrow aAbc \mid \varepsilon$
  $A \rightarrow aAbC \mid \varepsilon$
  $Cb \rightarrow bC$
  $Cc \rightarrow cc \}$
  $S \Rightarrow aAbc \Rightarrow aaAbCbc \Rightarrow aabbCc \Rightarrow aabbcc$



## Unrestricted grammars

- Question: Are unrestricted grammars as powerful as Turing machines?



## Extra exercises

- True or False: $|R| > |[0, 1]|$
- Consider a PDA having a FIFO queue instead of a stack(i.e., write-only at the top, read-only at the bottom). Does this modification change the class of languages accepted by ordinary PDA's?