Foundations of Computing

Lecture 7

Arkady Yerukhimovich

February 6, 2024

Outline

Lecture 6 Review

- Nonregular languages
- Proving The NFA pumping lemma
- Using the pumping lemma

Lecture 6 Review

- Nonregular languages
- Proving The NFA pumping lemma
- Using the pumping lemma

Today

- Some more examples proving languages are not regular
- Going beyond regular languages

HW2 Problem 4

Let L be a regular language, prove that the following languages are regular.

1. $NOPREFIX(L) = \{w \in L|$ no proper prefix of w is a member of $L\}$
2. $NOEXTEND(L) = \{w \in L | w$ is not a proper prefix of any string in $L\}$

HW2 Problem 4

Let L be a regular language, prove that the following languages are regular.

1. $NOPREFIX(L) = \{w \in L |$ no proper prefix of w is a member of $L\}$
2. $NOEXTEND(L) = \{w \in L | w$ is not a proper prefix of any string in $L\}$

Example:

- $L = \{00, 11, 001, 101\}$
- $NOPREFIX(L) = \{00, 11, 101\}$
- $NOEXTEND(L) = \{11, 001, 101\}$

HW2 Problem 4

Let L be a regular language, prove that the following languages are regular.

1. $NOPREFIX(L) = \{w \in L |$ no proper prefix of w is a member of $L\}$

HW2 Problem 4

Let L be a regular language, prove that the following languages are regular.

1. $NOPREFIX(L) = \{w \in L|$ no proper prefix of w is a member of $L\}$
2. $NOEXTEND(L) = \{w \in L | w$ is not a proper prefix of any string in $L\}$

Outline

The Regular Language Pumping Lemma

Pumping Lemma

If L is a regular language, then there exists an integer p (the pumping length) where any string $w \in L$ such that $|w| \geq p$ can be divided into three pieces $w = xyz$ satisfying:

1. For each $i \geq 0$, $xy^iz \in L$
2. $|y| > 0$, and
3. $|xy| \leq p$

Outline

The Proof Procedure

To use the pumping lemma to prove that L is not regular, we do the following:

The Proof Procedure

To use the pumping lemma to prove that L is not regular, we do the following:

1. Assume that L is regular

The Proof Procedure

To use the pumping lemma to prove that L is not regular, we do the following:

1. Assume that L is regular
2. Use pumping lemma to guarantee pumping length p, s.t. all w with $|w| > p$ can be pumped – Note: proof must work for all p

The Proof Procedure

To use the pumping lemma to prove that L is not regular, we do the following:

1. Assume that L is regular
2. Use pumping lemma to guarantee pumping length p, s.t. all w with $|w| > p$ can be pumped – Note: proof must work for all p
3. Choose $w \in L$ with $|w| \geq p$

The Proof Procedure

To use the pumping lemma to prove that L is not regular, we do the following:

1. Assume that L is regular
2. Use pumping lemma to guarantee pumping length p, s.t. all w with $|w| > p$ can be pumped – Note: proof must work for all p
3. Choose $w \in L$ with $|w| \geq p$
4. Demonstrate that w cannot be pumped
    - For each possible division $w = xyz$ (with $|y| > 0$ and $|xy| \leq p$), find an integer i such that $xy^iz \notin L$

The Proof Procedure

To use the pumping lemma to prove that L is not regular, we do the following:

1. Assume that L is regular
2. Use pumping lemma to guarantee pumping length p, s.t. all w with $|w| > p$ can be pumped – Note: proof must work for all p
3. Choose $w \in L$ with $|w| \geq p$
4. Demonstrate that w cannot be pumped
   - For each possible division $w = xyz$ (with $|y| > 0$ and $|xy| \leq p$), find an integer i such that $xy^iz \notin L$
5. Contradiction!!!

Prior Examples

We've already seen how to prove:

- $L = \{0^n1^n | n \geq 0\}$ is not regular
- $L = \{w | w \text{ has an equal number of 0s and 1s}\}$ is not regular

In both proofs, we picked $w = 0^p1^p$
Easy to show that this string cannot be pumped

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

1. Assume L is regular, and let p be the pumping length

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

1. Assume L is regular, and let p be the pumping length
2. For any sufficiently long w, by pumping lemma, there must be a partition $w = xyz$ s.t. $xy^i z \in L$ for all i

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

1. Assume L is regular, and let p be the pumping length
2. For any sufficiently long w, by pumping lemma, there must be a partition $w = xyz$ s.t. $xy^i z \in L$ for all i
3. Goal: Show that for all partitions, $xy^i z \notin L$ for some $i \geq 0$

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

1. Assume L is regular, and let p be the pumping length
2. For any sufficiently long w, by pumping lemma, there must be a partition $w = xyz$ s.t. $xy^iz \in L$ for all i
3. Goal: Show that for all partitions, $xy^iz \notin L$ for some $i \geq 0$
   That is, $xy^iz = 0^{m'}1^{n'}$ with $m' = n'$.

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

1. Assume L is regular, and let p be the pumping length
2. For any sufficiently long w, by pumping lemma, there must be a partition $w = xyz$ s.t. $xy^i z \in L$ for all i
3. Goal: Show that for all partitions, $xy^i z \notin L$ for some $i \geq 0$
   That is, $xy^i z = 0^{m'} 1^{n'}$ with $m' = n'$.

Question

What w should we choose?

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^p 1^{p+1}$, then since $|xy| \leq p$, $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{p-(\alpha+\beta)} 1^{p+1}$

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^p 1^{p+1}$, then since $|xy| \leq p$, $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{p-(\alpha+\beta)} 1^{p+1}$
2. Consider what happens when we pump k times:

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^p 1^{p+1}$, then since $|xy| \leq p$, $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{p-(\alpha+\beta)} 1^{p+1}$
2. Consider what happens when we pump k times:

$$xy^k z = 0^{\alpha + k\beta + p - (\alpha+\beta)} 1^{p+1}.$$

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^p 1^{p+1}$, then since $|xy| \leq p$, $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{p-(\alpha+\beta)} 1^{p+1}$
2. Consider what happens when we pump k times:

$$xy^k z = 0^{\alpha + k\beta + p - (\alpha+\beta)} 1^{p+1}.$$

For this to give a contradiction we need

$$m' = n', \text{i.e. } \alpha + k\beta + p - (\alpha + \beta) = p + (k-1)\beta = p + 1$$

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^p 1^{p+1}$, then since $|xy| \leq p$, $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{p-(\alpha+\beta)} 1^{p+1}$
2. Consider what happens when we pump k times:

$$xy^k z = 0^{\alpha + k\beta + p - (\alpha+\beta)} 1^{p+1}.$$

For this to give a contradiction we need

$$m' = n', \text{i.e. } \alpha + k\beta + p - (\alpha + \beta) = p + (k-1)\beta = p + 1$$

Equivalently, we need

$$(k-1)\beta = 1$$

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^iz = 0^{m'}1^{m'}$.

1. Suppose we choose $w = 0^p 1^{p+1}$, then since $|xy| \leq p$, $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{p-(\alpha+\beta)}1^{p+1}$
2. Consider what happens when we pump k times:

   $$xy^kz = 0^{\alpha+k\beta+p-(\alpha+\beta)}1^{p+1}.$$

   For this to give a contradiction we need

   $$m' = n', \text{i.e. } \alpha + k\beta + p - (\alpha+\beta) = p + (k-1)\beta = p+1$$

   Equivalently, we need

   $$(k-1)\beta = 1$$
3. But, we can't control β, so this w does not work

Let's try again!!!

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^m 1^n$ with $m \geq p$, then $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{m-(\alpha+\beta)} 1^n$

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^m 1^n$ with $m \geq p$, then $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{m-(\alpha+\beta)} 1^n$
2. Consider what happens when we pump k times:

$$xy^k z = 0^{\alpha + k\beta + m - (\alpha+\beta)} 1^n.$$

A More Challenging Example

Consider $L = \{0^m1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^iz = 0^{m'}1^{m'}$.

1. Suppose we choose $w = 0^m1^n$ with $m \geq p$, then $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{m-(\alpha+\beta)}1^n$
2. Consider what happens when we pump k times:

$$xy^kz = 0^{\alpha+k\beta+m-(\alpha+\beta)}1^n.$$

We need a k s.t. $m + (k-1)\beta = n$ for a contradiction

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^m 1^n$ with $m \geq p$, then $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{m-(\alpha+\beta)} 1^n$
2. Consider what happens when we pump k times:

$$xy^k z = 0^{\alpha + k\beta + m - (\alpha+\beta)} 1^n.$$

We need a k s.t. $m + (k-1)\beta = n$ for a contradiction
Equivalently, we need $k = 1 + (n-m)/\beta$ to be an integer

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^m 1^n$ with $m \geq p$, then $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{m-(\alpha+\beta)} 1^n$
2. Consider what happens when we pump k times:

$$xy^k z = 0^{\alpha + k\beta + m - (\alpha+\beta)} 1^n.$$

   We need a k s.t. $m + (k-1)\beta = n$ for a contradiction
   Equivalently, we need $k = 1 + (n-m)/\beta$ to be an integer
3. We only know $\beta \leq p$, how can we guarantee $(n-m)$ is divisible by β?

A More Challenging Example

Consider $L = \{0^m 1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^i z = 0^{m'} 1^{m'}$.

1. Suppose we choose $w = 0^m 1^n$ with $m \geq p$, then $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{m-(\alpha+\beta)} 1^n$
2. Consider what happens when we pump k times:

$$xy^k z = 0^{\alpha + k\beta + m - (\alpha+\beta)} 1^n.$$

   We need a k s.t. $m + (k-1)\beta = n$ for a contradiction
   Equivalently, we need $k = 1 + (n-m)/\beta$ to be an integer
3. We only know $\beta \leq p$, how can we guarantee $(n-m)$ is divisible by β?

Hint: What number is divisible by all integers $\leq p$?

A More Challenging Example

Consider $L = \{0^m1^n | m \neq n\}$, prove L is not regular

Goal

Pick a w s.t. for all partitions $w = xyz$, for some $i \geq 0$, $xy^iz = 0^{m'}1^{m'}$.

1. Suppose we choose $w = 0^m1^n$ with $m \geq p$, then $x = 0^\alpha$, $y = 0^\beta$, $z = 0^{m-(\alpha+\beta)}1^n$
2. Consider what happens when we pump k times:
$$xy^kz = 0^{\alpha+k\beta+m-(\alpha+\beta)}1^n.$$
We need a k s.t. $m + (k-1)\beta = n$ for a contradiction
Equivalently, we need $k = 1 + (n-m)/\beta$ to be an integer
3. We only know $\beta \leq p$, how can we guarantee $(n-m)$ is divisible by β?
Hint: What number is divisible by all integers $\leq p$?
4. Set $n = 2(p!)$, $m = p!$, then $(n-m) = p!$ is divisible by β, so there is k s.t. $xy^kz \notin L$

Hints for Using the Pumping Lemma

To use the pumping lemma, need to do the following

- Identify what it means for $x \notin L$
- Choose w such that any valid split xyz can lead to a contradiction
- Prove that $w' = xy^k z \notin L$ form some k

Choosing w is often tricky, requires intuition and some trial and error.

Outline

Proving Non-Regularity Using Closure Properties

Consider $L = \{w | w \text{ has an equal number of 0s and 1s}\}$, prove L is not regular

A simpler proof:

Proving Non-Regularity Using Closure Properties

Consider $L = \{w | w \text{ has an equal number of 0s and 1s}\}$, prove L is not regular

A simpler proof:

1. We already proved that $L_1 = \{0^n 1^n | n \geq 0\}$ is nonregular

Proving Non-Regularity Using Closure Properties

Consider $L = \{w | w \text{ has an equal number of 0s and 1s}\}$, prove L is not regular

A simpler proof:

1. We already proved that $L_1 = \{0^n 1^n | n \geq 0\}$ is nonregular
2. Observe that $L_1 = L \cap 0^* 1^*$

Proving Non-Regularity Using Closure Properties

Consider $L = \{w | w \text{ has an equal number of 0s and 1s}\}$, prove L is not regular

A simpler proof:

1. We already proved that $L_1 = \{0^n 1^n | n \geq 0\}$ is nonregular
2. Observe that $L_1 = L \cap 0^* 1^*$
3. Easy to see that $0^* 1^*$ is regular
4. Since regular languages are closed under $\cap$, if L is regular then L_1 must be regular

Proving Non-Regularity Using Closure Properties

Consider $L = \{w | w \text{ has an equal number of 0s and 1s}\}$, prove L is not regular

A simpler proof:

1. We already proved that $L_1 = \{0^n 1^n | n \geq 0\}$ is nonregular
2. Observe that $L_1 = L \cap 0^* 1^*$
3. Easy to see that $0^* 1^*$ is regular
4. Since regular languages are closed under $\cap$, if L is regular then L_1 must be regular
5. Since we know L_1 is nonregular, this means that L must be nonregular

Using Closure Properties of Regular Languages

We have seen a number of closure properties of REs

1. Closure under complement: $\overline{L}$ is regular if L is
2. Closure under union: $L_1 \cup L_2$ is regular if L_1, L_2 are
3. Closure under intersection: $L_1 \cap L_2$ is regular if L_1, L_2 are
4. Closure under reversal: L^R is regular if L is
5. NOPREFIX, NOEXTEND
6. There are many more (e.g., set difference, cross product, ...)

Using Closure Properties of Regular Languages

We have seen a number of closure properties of REs

1. Closure under complement: $\overline{L}$ is regular if L is
2. Closure under union: $L_1 \cup L_2$ is regular if L_1, L_2 are
3. Closure under intersection: $L_1 \cap L_2$ is regular if L_1, L_2 are
4. Closure under reversal: L^R is regular if L is
5. NOPREFIX, NOEXTEND
6. There are many more (e.g., set difference, cross product, ...)

Important

- It is often much easier to prove non-regularity using closure properties
- Try this first before you turn to pumping lemma

Exercise

Prove that the following language is nonregular:

$$L = \{0^i 1^j 2^i 3^j | i, j > 0\}$$

How Can We Recognize Non-Regular Languages?

Let $L = \{0^n 1^n | n \geq 0\}$

Question

How can we build a machine to recognize this language?

How Can We Recognize Non-Regular Languages?

Let $L = \{0^n 1^n | n \geq 0\}$

Question

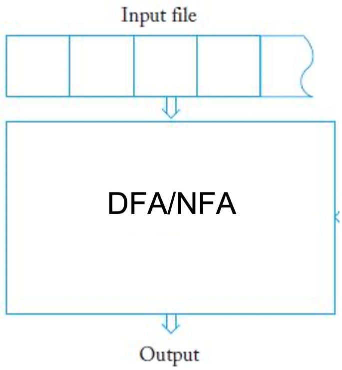
How can we build a machine to recognize this language?

Answer

Add some form of memory

Outline

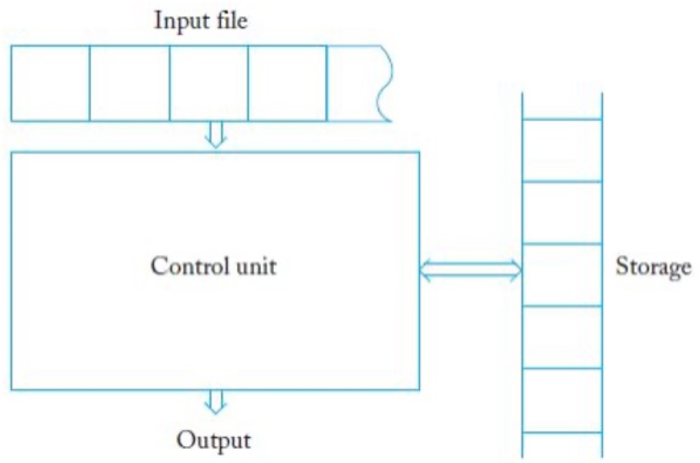
Let's Add Some Storage



Recall:

- An NFA/DFA has no external storage
- Only memory must be encoded in the finite number of states
- Can only recognize regular languages

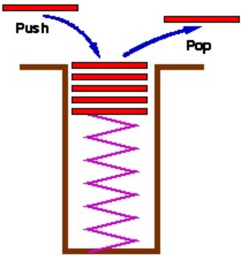
Let's Add Some Storage



Question

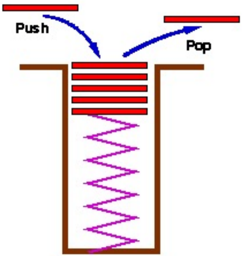
What kind of storage should we add?

A Stack



Let's add a Stack for storage

A Stack

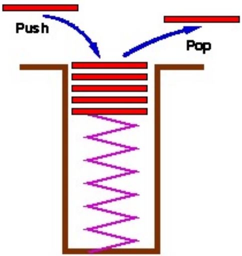


Let's add a Stack for storage

A stack has the following operations:

- push value - push a value onto the top of the stack

A Stack

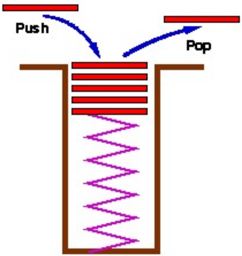


Let's add a Stack for storage

A stack has the following operations:

- push value - push a value onto the top of the stack
- pop value - pop the top item off the stack

A Stack

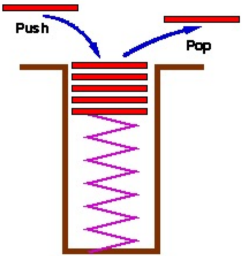


Let's add a Stack for storage

A stack has the following operations:

- push value - push a value onto the top of the stack
- pop value - pop the top item off the stack
- do nothing - denoted as ϵ

A Stack



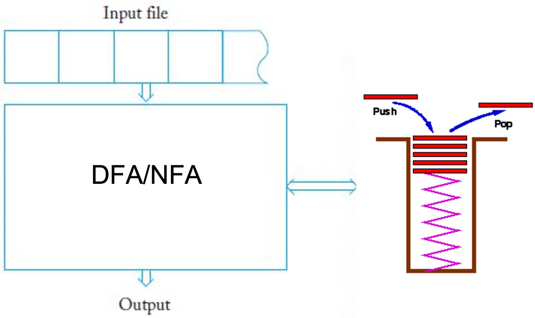
Let's add a Stack for storage

A stack has the following operations:

- push value - push a value onto the top of the stack
- pop value - pop the top item off the stack
- do nothing - denoted as ϵ

A stack is a Last-In First-Out (LIFO) data structure, that can hold an infinite amount of information

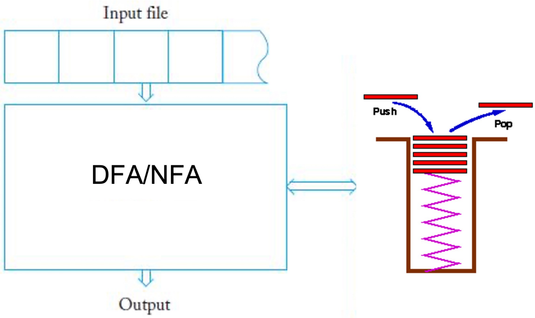
Pushdown Automata (PDA)



A PDA consists of:

- An NFA for a control unit
- A Stack for storage

Pushdown Automata (PDA)



A PDA consists of:

- An NFA for a control unit
- A Stack for storage

Question

Is this any more powerful than an NFA?

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

1. Read a symbol from the input tape

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

1. Read a symbol from the input tape
2. Optionally, pop a value from the Stack

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

1. Read a symbol from the input tape
2. Optionally, pop a value from the Stack
3. Use the input symbol and the Stack symbol to choose a next state

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

1. Read a symbol from the input tape
2. Optionally, pop a value from the Stack
3. Use the input symbol and the Stack symbol to choose a next state
4. Optionally, push a value onto the Stack

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

1. Read a symbol from the input tape
2. Optionally, pop a value from the Stack
3. Use the input symbol and the Stack symbol to choose a next state
4. Optionally, push a value onto the Stack

A PDA M accepts a string w if the NFA in the control stops in an accept state once all the input has been processed

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

1. Read a symbol from the input tape
2. Optionally, pop a value from the Stack
3. Use the input symbol and the Stack symbol to choose a next state
4. Optionally, push a value onto the Stack

A PDA M accepts a string w if the NFA in the control stops in an accept state once all the input has been processed

Observations:

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

1. Read a symbol from the input tape
2. Optionally, pop a value from the Stack
3. Use the input symbol and the Stack symbol to choose a next state
4. Optionally, push a value onto the Stack

A PDA M accepts a string w if the NFA in the control stops in an accept state once all the input has been processed

Observations:

- Since the control is an NFA, ϵ transitions are allowed

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

1. Read a symbol from the input tape
2. Optionally, pop a value from the Stack
3. Use the input symbol and the Stack symbol to choose a next state
4. Optionally, push a value onto the Stack

A PDA M accepts a string w if the NFA in the control stops in an accept state once all the input has been processed

Observations:

- Since the control is an NFA, ϵ transitions are allowed
- A PDA may choose not to touch the stack in a particular step

Computing With a PDA

Computing with a PDA

At each step, a PDA can do the following

1. Read a symbol from the input tape
2. Optionally, pop a value from the Stack
3. Use the input symbol and the Stack symbol to choose a next state
4. Optionally, push a value onto the Stack

A PDA M accepts a string w if the NFA in the control stops in an accept state once all the input has been processed

Observations:

- Since the control is an NFA, ϵ transitions are allowed
- A PDA may choose not to touch the stack in a particular step
- Unlike the case for DFA/NFA, deterministic PDA's are not equal to non-deterministic ones. We will only study non-deterministic PDAs.

An Example PDA

A PDA for $L = \{0^n 1^n | n \geq 0\}$

An Example PDA

A PDA for $L = \{0^n 1^n | n \geq 0\}$

Consider the following PDA "Algorithm"

An Example PDA

A PDA for $L = \{0^n 1^n | n \geq 0\}$

Consider the following PDA "Algorithm"

1. Read a symbol from the input

An Example PDA

A PDA for $L = \{0^n1^n | n \geq 0\}$

Consider the following PDA "Algorithm"

1. Read a symbol from the input
2. If it is a 0 and I have not seen any 1s, then push a 0 onto the stack

An Example PDA

A PDA for $L = \{0^n 1^n | n \geq 0\}$

Consider the following PDA "Algorithm"

1. Read a symbol from the input
2. If it is a 0 and I have not seen any 1s, then push a 0 onto the stack
3. If it is a 1, pop a value (a 0) from the stack

An Example PDA

A PDA for $L = \{0^n 1^n | n \geq 0\}$

Consider the following PDA "Algorithm"

1. Read a symbol from the input
2. If it is a 0 and I have not seen any 1s, then push a 0 onto the stack
3. If it is a 1, pop a value (a 0) from the stack
4. Accept if and only if the stack becomes empty when we read the last character

An Example PDA

A PDA for $L = \{0^n 1^n | n \geq 0\}$

Consider the following PDA "Algorithm"

1. Read a symbol from the input
2. If it is a 0 and I have not seen any 1s, then push a 0 onto the stack
3. If it is a 1, pop a value (a 0) from the stack
4. Accept if and only if the stack becomes empty when we read the last character
5. Reject if any of the following happen:
    - the stack becomes empty and the input is not done or

An Example PDA

A PDA for $L = \{0^n 1^n | n \geq 0\}$

Consider the following PDA "Algorithm"

1. Read a symbol from the input
2. If it is a 0 and I have not seen any 1s, then push a 0 onto the stack
3. If it is a 1, pop a value (a 0) from the stack
4. Accept if and only if the stack becomes empty when we read the last character
5. Reject if any of the following happen:
    - the stack becomes empty and the input is not done or
    - there are still 0s left on the stack when the last input is read or

An Example PDA

A PDA for $L = \{0^n 1^n | n \geq 0\}$

Consider the following PDA "Algorithm"

1. Read a symbol from the input
2. If it is a 0 and I have not seen any 1s, then push a 0 onto the stack
3. If it is a 1, pop a value (a 0) from the stack
4. Accept if and only if the stack becomes empty when we read the last character
5. Reject if any of the following happen:
    - the stack becomes empty and the input is not done or
    - there are still 0s left on the stack when the last input is read or
    - there are any 0s after the first 1

An Example PDA

A PDA for $L = \{0^n 1^n | n \geq 0\}$

Consider the following PDA "Algorithm"

1. Read a symbol from the input
2. If it is a 0 and I have not seen any 1s, then push a 0 onto the stack
3. If it is a 1, pop a value (a 0) from the stack
4. Accept if and only if the stack becomes empty when we read the last character
5. Reject if any of the following happen:
    - the stack becomes empty and the input is not done or
    - there are still 0s left on the stack when the last input is read or
    - there are any 0s after the first 1