CS420

Introduction to the Theory of Computation

Lecture 15: The pumping lemma; irregular languages

Tiago Cogumbreiro

Today we will learn...

- Introduce irregular languages
- Intuition of the Pumping lemma
- The Pumping lemma formally
- Proving a language to be irregular (with the Pumping lemma)
- Formally proving a language to be irregular (with Coq)

> Section 1.4 irregular Languages (ITC book)

What is a regular language?

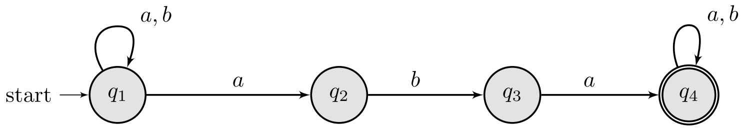
What is a regular language?

Definition 1.16

We say that L_1 is regular if there exists a DFA M such that $L(M) = L_1$.

Example 1

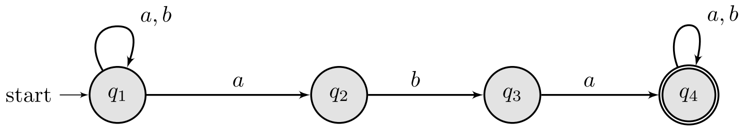
Let N_1 be the following NFA:



Is $L(N_1)$ regular?

Example 1

Let N_1 be the following NFA:

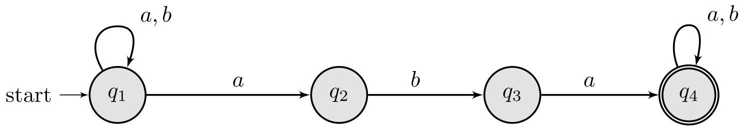


Is $L(N_1)$ regular?

Yes. **Proof:** we can convert N_1 into an equivalent DFA, which then satisfies Definition 1.16.

Example 1

Let N_1 be the following NFA:



Is $L(N_1)$ regular?

Yes. **Proof:** we can convert N_1 into an equivalent DFA, which then satisfies Definition 1.16.

Theorem

> We say that L_1 is regular, if there exits an NFA N such that $L(N) = L_1$

Example 2

Is $L(\mathbf{0} \cup \mathbf{1}^\star)$ regular?

Example 2

Is $L(0 \cup 1^\star)$ regular?

Yes. **Proof:** We have that $L(0 \cup 1^\star) = L(\mathrm{NFA}(0 \cup 1^\star))$, which is regular (from the previous theorem).

Example 2

Is $L(0 \cup 1^\star)$ regular?

Yes. **Proof:** We have that $L(0 \cup 1^\star) = L(\mathrm{NFA}(0 \cup 1^\star))$, which is regular (from the previous theorem).

Theorem

> We say that L_1 is regular, if there exits a regular expression R such that $L(R) = L_1$

What is a regular language?

1. A language is regular if there exists a DFA that recognizes it
2. A language is regular if there exists an NFA that recognizes it
3. A language is regular if there exists a Regex that recognizes it

Example

The language of strings that have a possibly empty sequence of n zeroes followed by a sequence of n ones.

$$L_4 = \{0^n 1^n \mid \forall n \colon n \geq 0\}$$

Is this language ***regular***?

Example

The language of strings that have a possibly empty sequence of n zeroes followed by a sequence of n ones.

$$L_4 = \{0^n 1^n \mid \forall n \colon n \geq 0\}$$

Is this language ***regular***?

How do we prove that a language is ***not*** regular?

Example

The language of strings that have a possibly empty sequence of n zeroes followed by a sequence of n ones.

$$L_4 = \{0^n 1^n \mid \forall n \colon n \geq 0\}$$

Is this language ***regular***?

How do we prove that a language is ***not*** regular?

The only way we know is by proving that there is no NFA/DFA/regex that can recognize such a language.

irregular languages

How do we prove that a language is not regular?

How do we prove a language is irregular?

- There are multiple ways to do it: pumping lemma, Myhill–Nerode theorem
- In this course, we will use of the pumping lemma.

Michael Rabin and Dana Scott (1959). "Finite Automata and Their Decision Problems" . IBM Journal of Research and Development. 3 (2): 114–125. DOI:10.1147/rd.32.0114.

Using the pumping lemma to prove irregularity

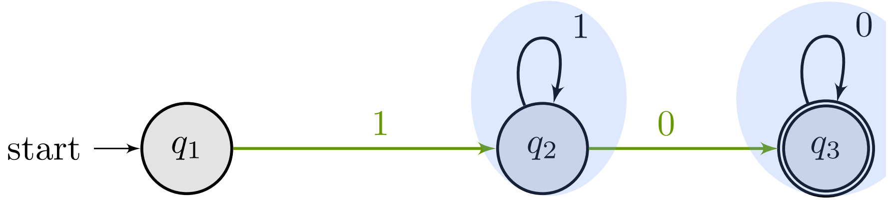
Pumping lemma

An intuition

> The pumping lemma tells us that **all** regular languages (that have a loop) have the following characteristics:

Every word in a regular language, $w \in L$, can be partitioned into three parts $w = xyz$:

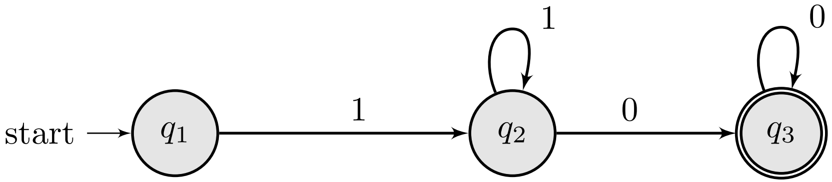
- a portion x before the first loop,
- a portion y that is one loop's iteration (nonempty), and
- a portion z that follows the first loop



Additionally, since y is a loop, then it may be omitted or replicated as many times as we want and that word will also be in the given language, that is $xy^i z \in L$

Pumping lemma

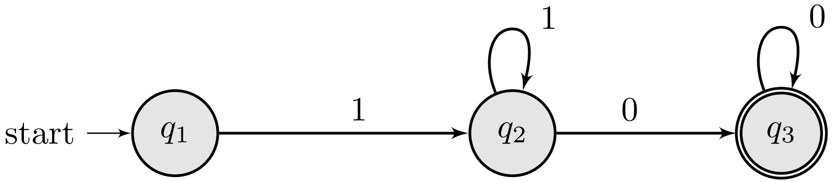
Pictorial intuition



You: Give me any string accepted by the automaton of at least size 3.

Pumping lemma

Pictorial intuition

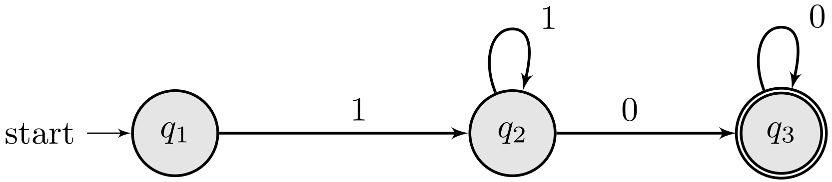


You: Give me any string accepted by the automaton of at least size 3.

Example: 100

Pumping lemma

Pictorial intuition



You: Give me any string accepted by the automaton of at least size 3.

Example: 100

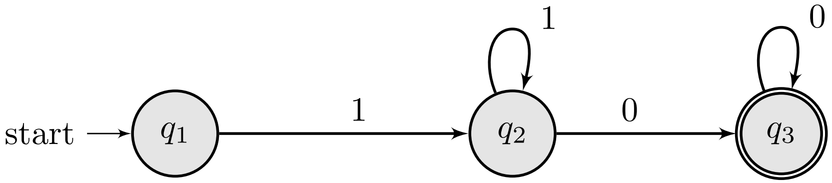
Me: I will partition 100 into three parts $100 = xyz$ such that $xy^i z$ is accepted for any i:

$$\underbrace{10}_{x}\underbrace{0}_{y}\underbrace{\epsilon}_{z}$$

- $xz = 10 \cdot \epsilon = 10$ is accepted
- $x\underline{yy}z = 1\underline{000}$ is accepted
- $x\underline{yyyy}z = 1\underline{00000}$ is accepted
- $x\underline{yyyyyy}z = 1\underline{0000000}$ is accepted

Pumping lemma

Pictorial intuition



You: Give me a string of **size 4**.

Example: 1100

Me: I will partition 1100 into three parts $1100 = xyz$ such that $xy^i z$ is accepted for any i:

$$\underbrace{1}_{x}\underbrace{1}_{y}\underbrace{00}_{z}$$

- $xz = 100$ is accepted
- $xyyz = 11100$ is accepted
- $xyyyyz = 1111100$ is accepted
- $xyyyyyyz = 111111100$ is accepted

The Pumping Lemma, formally

Pump language

We say that w is in language $\mathrm{Pump}(L, p)$ if:

1. You can partition w into three sections: $w = x \cdot y \cdot z$
2. The middle section y is nonempty
3. The first two sections have at most length p: $|x \cdot y| \leq p$
4. For any i, we have $x \cdot y^i \cdot z \in L$

```
Inductive Pump L p (w:word) : Prop :=
| pump_def:
  forall x y z,
  w = x ++ y ++ z →
  y <> [] →
  length (x ++ y) ≤ p →
  (forall i, In (x ++ pow y i ++ z) L) →
  Pump L p w.
```

Pumping lemma

```
Theorem pumping:
  forall L,
  Regular L →
  exists p, p ≥ 1 /\
  (forall w, In w L → length w ≥ p → In w (Pump L p)).
```

Intuition

> **Regular languages:** there exists a minimum length such that every word of that length is pump-able.
>
> 1. If L is regular
> 2. Then, there exists some p such that
> 3. Any word w with at least length p is in $\mathrm{Pump}(L, p)$

What about **regular** languages without loops?

Such languages have a maximum string length k. Pick the pumping length to be $k + 1$, now your pumping property is vacuously true.

But how do I use the pumping lemma?

Let us apply pumping to `Examples.L3 = "a" >> All >> "b"`.

```
Goal exists p : nat,
     p ≥ 1 /\
     (forall w : word,
      In w Examples.L3 →
      length w ≥ p → In w (Pump Examples.L3 p)).
Proof.
  apply (pumping _ l3_is_reg).
Qed.
```

How do you use the pumping lemma?

You cannot use the result because you don't know what p is:

```
Ha : p ≥ 1
Hb : forall w : word,
     In w Examples.L3 →
     length w ≥ p → In w (Pump Examples.L3 p)
______________________________________(1/1)
False
```

How do you use assumption `Hb`?

We use the pumping lemma to prove irregularity

Via the contrapositive

```
Goal (P → Q) → (~ Q → ~ P).
```

Deriving irregularity

From the Pumping lemma

- If L regular, then pump-able, aka: $\exists p, p \geq 1 \land (\forall w, w \in L \implies |w| \geq p \implies w \in \text{Pump}(L, p))$
- **Thus, if L is not pump-able, then L is not regular.**

What happens next?

1. To help you prove irregularity, we will define a $\neg\text{Pump}(L, p)$ without using the negation.
2. We show you how to use $\neg\text{Pump}(L, p)$ to conclude irregularity

What is a non-pump-able language?

The Clogs language

Language $\mathrm{Clogs}(L, p)$ is the reverse of $\mathrm{Pump}(L, p)$.

The Clogs language

```
Definition Clogs (L:language) p w :=
  forall (x y z:word),
    w = x ++ y ++ z →
    y <> [] →
    length (x ++ y) ≤ p →
    exists i,
    ~ In (x ++ (pow y i) ++ z) L.
```

Recall the Pump language

```
Inductive Pump L p (w:word) : Prop :=
| pump_def:
  forall x y z,
  w = x ++ y ++ z →
  y <> [] →
  length (x ++ y) ≤ p →
  (forall i, In (x ++ pow y i ++ z) L) →
  Pump L p w.
```

The Clogs language

Intuition

All strings that break the pumping property for language L

Language $\mathrm{Clogs}(L,p)$: every word that can be partitioned into three parts $w = xyz$, where $y \neq \epsilon$, and $|xy| \leq p$, and which we can conclude that $xy^i z \notin L$ for some i.

The Clogs language

```
Definition Clogs (L:language) p w :=
  forall (x y z:word),
    w = x ++ y ++ z →
    y <> [] →
    length (x ++ y) ≤ p →
    exists i,
    ~ In (x ++ (pow y i) ++ z) L.
```

Clogged language

We say that a language L is clogged at length p if:

1. There exists a word w of length p in L
2. And that word is in $\text{Clogs}(L, p)$

Formally

```
Inductive Clogged (L:language) p : Prop :=
| clogged_word:
  forall w,
  In w L →
  length w ≥ p →
  In w (Clogs L p) →
  Clogged L p.
```

irregular languages

If we can clog L for every length $p \geq 1$, then L is not regular.

```
Lemma not_regular:
  forall (L:language),
  (forall p, p ≥ 1 → Clogged L p) →
  ~ Regular L.
```

$\{0^n 1^n \mid \forall n \colon n \geq 0\}$ is irregular

Proving irregular languages

Theorem $L_1 = \{0^n 1^n \mid \forall n \colon n \geq 0\}$ is not regular.

Proof idea

Show that we can clog L with any p.

Q: How do we show that we can clog L?

1. Pick a word w that is in L
2. Show that $|w| \geq p$ where p is unknown
3. Show that w clogs L with p.

```
Inductive Clogged (L:language) p : Prop :
| clogged_word:
  forall w,
  In w L →
  length w ≥ p →
  In w (Clogs L p) →
  Clogged L p.
```

How do we clog a irregular language?

Intuition

Use the pumping length to your advantage.
We have:

1. $|w| \geq p$
2. $w = |xyz|$
3. $|xy| \leq p$

Idea for L_1

- If we pick $0^p 1^p$, then because of (3) $|xy| \leq p$ we get that y must consist of 0's only
- When we pump y once, thus $xyyz$, we have more 0's than 1's
- The pumped string is no longer has the same 0's than 1's

Theorem $L_1 = \{0^n 1^n \mid \forall n\colon n \geq 0\}$ is not regular.

Proof. We prove that the language above does not satisfy the pumping property, thus the language is not regular. Let p be the pumping length.

We pick $w = 0^p 1^p$ and must show that clog L:

1. $w \in \{0^n 1^n \mid \forall n\colon n \geq 0\}$, which holds by replacing n by p.
2. $|w| \geq p$, which holds since $|w| = 2p \geq p$.

Theorem $L_1 = \{0^n 1^n \mid \forall n \colon n \geq 0\}$ is not regular.

Proof. We prove that the language above does not satisfy the pumping property, thus the language is not regular. Let p be the pumping length.

We pick $w = 0^p 1^p$ and must show that clog L:

1. $w \in \{0^n 1^n \mid \forall n \colon n \geq 0\}$, which holds by replacing n by p.
2. $|w| \geq p$, which holds since $|w| = 2p \geq p$.
3. Finally, prove $w \in \text{Clogs}(L, p)$: given some x, y, z our assumptions are (H1) $w = xyz$, (H2) $|xy| \leq p$, and (H3) $|y| > 0$, we must prove that

$$\exists i, xy^i z \notin L_1$$

(We write in red what you need to prove)

Proof. (Continuation...)

Let $a + b = p$, where $xy = 0^a$ and $a, b \in \mathcal{N}_0$ (non-negative).

We can rewrite (H1) $w = xyz$ such that

$$(H_1) \quad w = \underbrace{0^p 1^p}_{xyz} = \underbrace{0^a}_{xy} \underbrace{0^b 1^{a+b}}_{z}$$

Proof. (Continuation...)

Let $a + b = p$, where $xy = 0^a$ and $a, b \in \mathcal{N}_0$ (non-negative).

We can rewrite (H1) $w = xyz$ such that

$$(H_1) \quad w = \underbrace{0^p 1^p}_{xyz} = \underbrace{0^a}_{xy} \underbrace{0^b 1^{a+b}}_{z}$$

Or, simply,

$$(H_1) \quad \underbrace{0^a}_{xy} \underbrace{0^b 1^{a+b}}_{z} = \underbrace{0^{|xy|}}_{xy} \underbrace{0^b 1^{|xy|+b}}_{z}$$

Proof. (Continuation...) We pick $i = 2$, so our goal is to show that

$$\underbrace{0^{|xyy|}}_{xyy}\underbrace{0^b 1^{|xy|+b}}_{z} \notin \{0^n 1^n \mid \forall n \colon n \geq 0\}$$

Proof. (Continuation...) We pick $i = 2$, so our goal is to show that

$$\underbrace{0^{|xyy|}}_{xyy}\underbrace{0^b 1^{|xy|+b}}_{z} \notin \{0^n 1^n \mid \forall n \colon n \geq 0\}$$

Thus, it is equivalent to show that

$$|xyy| + b \neq |xy| + b$$

We can simplify it with,

Proof. (Continuation...) We pick $i = 2$, so our goal is to show that

$$\underbrace{0^{|xyy|}}_{xyy}\underbrace{0^b 1^{|xy|+b}}_{z} \notin \{0^n 1^n \mid \forall n\colon n \geq 0\}$$

Thus, it is equivalent to show that

$$|xyy| + b \neq |xy| + b$$

We can simplify it with,

$$|xyy| + b - (|xy| + b) \neq |xy| + b - (|xy| + b)$$

And,

$$|y| \neq 0$$

Proof. (Continuation...) We pick $i = 2$, so our goal is to show that

$$\underbrace{0^{|xyy|}}_{xyy}\underbrace{0^b 1^{|xy|+b}}_{z} \notin \{0^n 1^n \mid \forall n \colon n \geq 0\}$$

Thus, it is equivalent to show that

$$|xyy| + b \neq |xy| + b$$

We can simplify it with,

$$|xyy| + b - (|xy| + b) \neq |xy| + b - (|xy| + b)$$

And,

$$|y| \neq 0$$

Which is trivially true since (H3) $|y| > 0$

$\{w \mid w \text{ has as many 0's as 1's}\}$ is not regular

Theorem $\{w \mid w \text{ has as many 0's as 1's}\}$ is not regular

Proof idea

1. **Adversary:** picks p such that $p \geq 0$

Theorem $\{w \mid w \text{ has as many 0's as 1's}\}$ is not regular

Proof idea

1. **Adversary:** picks p such that $p \geq 0$
2. **You:** Let us pick the same w as before
   $0^p 1^p \in A$ and $|w| \geq p$ (trivially holds)

Theorem $\{w \mid w \text{ has as many 0's as 1's}\}$ is not regular

Proof idea

1. **Adversary:** picks p such that $p \geq 0$
2. **You:** Let us pick the same w as before
   $0^p 1^p \in A$ and $|w| \geq p$ (trivially holds)
3. **Adversary:** decomposes w in xyz such that:
   $|y| > 0$ and $|xy| \leq p$

Theorem $\{w \mid w \text{ has as many 0's as 1's}\}$ is not regular

Proof idea

1. **Adversary:** picks p such that $p \geq 0$
2. **You:** Let us pick the same w as before
   $0^p1^p \in A$ and $|w| \geq p$ (trivially holds)
3. **Adversary:** decomposes w in xyz such that:
   $|y| > 0$ and $|xy| \leq p$
4. **You:** Let us pick $i = 2$:
   $i \geq 0$ (trivially holds)

Theorem $\{w \mid w \text{ has as many 0's as 1's}\}$ is not regular

Proof idea

1. **Adversary:** picks p such that $p \geq 0$
2. **You:** Let us pick the same w as before
   $0^p 1^p \in A$ and $|w| \geq p$ (trivially holds)
3. **Adversary:** decomposes w in xyz such that:
   $|y| > 0$ and $|xy| \leq p$
4. **You:** Let us pick $i = 2$:
   $i \geq 0$ (trivially holds)
5. **Goal:** **You:** show that $xyyz \notin A$

Why?

- We are responsible for picking w, which is the hardest part of the problem.
- By picking $0^p 1^p$, we replicate the proof we did in the previous exercise!

Theorem $L_2 = \{w \mid w \text{ has as many 0's as 1's}\}$ is not regular

Proof. We prove that the language above does not satisfy the pumping property, thus the language is not regular. Let p be the pumping length.

1. We pick $w = 0^p 1^p$ and must show that
   - $w \in L_2$, which holds since there are p 0's and p 1's.
   - $|w| \geq p$, which holds since $|w| = 2p \geq p$.
2. Finally, given some x, y, z our assumptions are (H1) $w = xyz$, (H2) $|xy| \leq p$, and (H3) $|y| > 0$, we must prove that

$$\exists i, xy^i z \notin L_2$$

(We write in red what you need to prove)

Proof. (Continuation...)

Let $p = a + b$ and $|xy| = a$. We pick $i = 2$ and show that

$$\underbrace{0^a}_{xy}\underbrace{0^{|y|}}_{y}\underbrace{0^b 1^{a+b}}_{z} \notin \{w \mid \forall n\colon n \text{ has as many 0's as 1's}\}$$

Proof. (Continuation...)

Let $p = a + b$ and $|xy| = a$. We pick $i = 2$ and show that

$$\underbrace{0^a}_{xy}\underbrace{0^{|y|}}_{y}\underbrace{0^b 1^{a+b}}_{z} \notin \{w \mid \forall n\colon n \text{ has as many 0's as 1's}\}$$

The goal below is equivalent:

$$a + |y| + b \neq a + b$$

Proof. (Continuation...)

Let $p = a + b$ and $|xy| = a$. We pick $i = 2$ and show that

$$\underbrace{0^a}_{xy}\ \underbrace{0^{|y|}}_{y}\ \underbrace{0^b 1^{a+b}}_{z} \notin \{w \mid \forall n\colon n \text{ has as many 0's as 1's}\}$$

The goal below is equivalent:

$$a + |y| + b \neq a + b$$

And can be simplified to

$$|y| \neq 0$$

Proof. (Continuation...)

Let $p = a + b$ and $|xy| = a$. We pick $i = 2$ and show that

$$\underbrace{0^a}_{xy}\underbrace{0^{|y|}}_{y}\underbrace{0^b 1^{a+b}}_{z} \notin \{w \mid \forall n\colon n \text{ has as many 0's as 1's}\}$$

The goal below is equivalent:

$$a + |y| + b \neq a + b$$

And can be simplified to

$$|y| \neq 0$$

Which is given by the hypothesis that $|y| > 0$.

$\{0^j 1^k \mid j > k\}$ is not regular

Theorem: $A = \{0^j 1^k \mid j > k\}$ is not regular

Proof idea

1. **Adversary:** picks p such that $p \geq 0$

Theorem: $A = \{0^j 1^k \mid j > k\}$ is not regular

Proof idea

1. **Adversary:** picks p such that $p \geq 0$
2. **You:** Let us pick $w = 0^{p+1}1^p$
   $0^{p+1}1^p \in A$ and $|w| \geq p$ (trivially holds)
3. **Adversary:** decomposes w in xyz such that:
   $|y| > 0$ and $|xy| \leq p$

Theorem: $A = \{0^j 1^k \mid j > k\}$ is not regular

Proof idea

1. **Adversary:** picks p such that $p \geq 0$
2. **You:** Let us pick $w = 0^{p+1}1^p$
   $0^{p+1}1^p \in A$ and $|w| \geq p$ (trivially holds)
3. **Adversary:** decomposes w in xyz such that:
   $|y| > 0$ and $|xy| \leq p$
4. **You:** Let us pick $i = 0$:
   $i \geq 0$ (trivially holds)
5. **Goal:** **You:** show that $xz \notin A$

Why?

- Ultimately, our goal is to show that $w \notin A$, thus that the exponent of 1 smaller or equal than the exponent of 0.
- Since the loop always appears on the left-hand side of the string, we should pick the smallest exponent possible that uses p and still $w \in A$. Thus, we pick $0^{p+1}1^p$.

Proof. We prove that the language above does not satisfy the pumping property, thus the language is not regular. Let p be the pumping length.

1. We pick $w = 0^{p+1}1^p \in A$. Let $|xy| + b = p$. We have $|xy| \leq p$ and that $w = 0^{p+1}1^p$.

Proof. We prove that the language above does not satisfy the pumping property, thus the language is not regular. Let p be the pumping length.

1. We pick $w = 0^{p+1}1^p \in A$. Let $|xy| + b = p$. We have $|xy| \leq p$ and that $w = 0^{p+1}1^p$.
2. We pick $i = 0$ and show that

$$xz \notin \{0^j 1^k \mid j > k\}$$

Proof. We prove that the language above does not satisfy the pumping property, thus the language is not regular. Let p be the pumping length.

1. We pick $w = 0^{p+1}1^p \in A$. Let $|xy| + b = p$. We have $|xy| \leq p$ and that $w = 0^{p+1}1^p$.
2. We pick $i = 0$ and show that

$$xz \notin \{0^j 1^k \mid j > k\}$$

3. Thus,

$$0^{|xy|-|y|+b+1}1^{|xy|+b} \notin \{0^j 1^k \mid j > k\}$$

Proof. We prove that the language above does not satisfy the pumping property, thus the language is not regular. Let p be the pumping length.

1. We pick $w = 0^{p+1}1^p \in A$. Let $|xy| + b = p$. We have $|xy| \leq p$ and that $w = 0^{p+1}1^p$.
2. We pick $i = 0$ and show that

$$xz \notin \{0^j 1^k \mid j > k\}$$

3. Thus,

$$0^{|xy|-|y|+b+1} 1^{|xy|+b} \notin \{0^j 1^k \mid j > k\}$$

4. So, we have to show that

$$|xy| - |y| + b + 1 \leq |xy| + b$$
$$|x| + 1 \leq |xy|$$
$$|y| \geq 1 \quad \text{which holds, since} |y| > 0$$