Languages and Machines

L1: Regular Languages

Jorge A. Pérez

Bernoulli Institute for Mathematics, Computer Science, and AI
University of Groningen, Groningen, the Netherlands

April 14, 2025

Preliminaries

- Induction for proofs and definitions
- Regular sets and languages
- Example of a proof
- Preview: Context-free languages

Basic Notation

- $x \in X, \quad X \subseteq Y$
- $\forall x \in X : P(x)$, $\exists x \in X : P(x)$
- $R \subseteq X \times Y$ is a relation between X and Y
- $x\, R\, y \equiv (x, y) \in R$
- $G = (V, E)$, with $E \subseteq V \times V$ is a directed graph
- R^* is the reflexive, transitive closure of relation R

Induction

The theory:

- Basis: $0 \in \mathbb{N}$
- Inductive (or recursive) step: if $n \in \mathbb{N}$ then $n + 1 \in \mathbb{N}$ too
- Closure: we only allow a finite number of steps ($\infty \notin \mathbb{N}$)

Induction

The theory:

- Basis: $0 \in \mathbb{N}$
- Inductive (or recursive) step: if $n \in \mathbb{N}$ then $n + 1 \in \mathbb{N}$ too
- Closure: we only allow a finite number of steps ($\infty \notin \mathbb{N}$)

The practice:
Given $f(n) = n(n+1)$ for all $n \in \mathbb{N}$, then $f(n)$ is even.

- Basis: for $n = 0$, we have that $f(n) = 0 \cdot 1 = 0$, which is even.
- Step: We must show that if $f(n)$ is even then $f(n+1)$ is even.

Induction

The theory:

- Basis: $0 \in \mathbb{N}$
- Inductive (or recursive) step: if $n \in \mathbb{N}$ then $n + 1 \in \mathbb{N}$ too
- Closure: we only allow a finite number of steps ($\infty \notin \mathbb{N}$)

The practice:

Given $f(n) = n(n+1)$ for all $n \in \mathbb{N}$, then $f(n)$ is even.

- Basis: for $n = 0$, we have that $f(n) = 0 \cdot 1 = 0$, which is even.
- Step: We must show that if $f(n)$ is even then $f(n+1)$ is even. Observe that

$$f(n+1) = (n+1)(n+2)$$

Induction

The theory:

- Basis: $0 \in \mathbb{N}$
- Inductive (or recursive) step: if $n \in \mathbb{N}$ then $n + 1 \in \mathbb{N}$ too
- Closure: we only allow a finite number of steps ($\infty \notin \mathbb{N}$)

The practice:

Given $f(n) = n(n + 1)$ for all $n \in \mathbb{N}$, then $f(n)$ is even.

- Basis: for $n = 0$, we have that $f(n) = 0 \cdot 1 = 0$, which is even.
- Step: We must show that if $f(n)$ is even then $f(n + 1)$ is even. Observe that

$$f(n + 1) = (n + 1)(n + 2) = n(n + 1) + 2(n + 1)$$

Induction

The theory:

- Basis: $0 \in \mathbb{N}$
- Inductive (or recursive) step: if $n \in \mathbb{N}$ then $n + 1 \in \mathbb{N}$ too
- Closure: we only allow a finite number of steps ($\infty \notin \mathbb{N}$)

The practice:
Given $f(n) = n(n+1)$ for all $n \in \mathbb{N}$, then $f(n)$ is even.

- Basis: for $n = 0$, we have that $f(n) = 0 \cdot 1 = 0$, which is even.
- Step: We must show that if $f(n)$ is even then $f(n+1)$ is even. Observe that

$$f(n+1) = (n+1)(n+2) = n(n+1) + 2(n+1) = f(n) + 2(n+1)$$

  Note: $f(n)$ is even (by IH) and $2(n+1)$ is also even (why?).
  Hence, $f(n+1)$ must be even too. This concludes the proof.

Induction is a **proof principle** and a tool for **defining mathematical objects**!

Strings and Languages

- Alphabet Σ: a finite set of indivisible elements (“letters”)
- Σ^*: the set of strings over Σ, defined recursively
- Language: a subset of Σ^*
- The empty string is denoted ϵ (read: epsilon)

Examples:

- Given $\Sigma = \{a, b\}$, elements of Σ^* include the empty string ϵ and non-empty strings such as ab, aaa, and $bbaba$
- Length: $|bbaba| = 5$.
- Symbol counts: $n_a(bbaba) = 2$

Operations on Strings

- Given strings u and v, the string uv is their concatenation.
  An associative operation: $(uv)w = u(vw)$.
- Derived concepts: substring, prefix, suffix.
- Replication ("exponentiation"): a string concatenated with itself.
- Given a string u, its reversal u^R is u written backwards

Examples:

- Given $u = ab$ and $v = ba$, their concatenation is $uv = abba$
- Replication: $a^3 = aaa$, $(ab)^2 = abab$.
- Reversal: $(abb)^R = bba$

An Inductive Definition

Question:
How to define the reversal of a string, inductively?

An Inductive Definition

Question:
How to define the reversal of a string, inductively?

Answer:
Let w be a finite string. We define w^R by induction on

An Inductive Definition

Question:
How to define the reversal of a string, inductively?

Answer:
Let w be a finite string. We define w^R by induction on $|w|$:

- Basis:
  In this case, $|w| = 0$. Then it must be the case that $w = \epsilon$.
  Therefore, $w^R = \epsilon$.

An Inductive Definition

Question:
How to define the reversal of a string, inductively?

Answer:
Let w be a finite string. We define w^R by induction on $|w|$:

- Basis:
  In this case, $|w| = 0$. Then it must be the case that $w = \epsilon$.
  Therefore, $w^R = \epsilon$.
- Step:
  In this case, $|w| = n \geq 1$

An Inductive Definition

Question:
How to define the reversal of a string, inductively?

Answer:
Let w be a finite string. We define w^R by induction on $|w|$:

- Basis:
  In this case, $|w| = 0$. Then it must be the case that $w = \epsilon$.
  Therefore, $w^R = \epsilon$.
- Step:
  In this case, $|w| = n \geq 1$ and so $w = u\, a$, with $|u| = n - 1$.
  Therefore, u^R is defined and so $w^R = a\, u^R$.

An Inductive Definition

Question:
How to define the reversal of a string, inductively?

Answer:
Let w be a finite string. We define w^R by induction on $|w|$:

- Basis:
  In this case, $|w| = 0$. Then it must be the case that $w = \epsilon$.
  Therefore, $w^R = \epsilon$.
- Step:
  In this case, $|w| = n \geq 1$ and so $w = u\,a$, with $|u| = n - 1$.
  Therefore, u^R is defined and so $w^R = a\,u^R$.

In what sense is this definition inductive?

Operations on Languages

- Operations on strings can be lifted to languages (sets of strings)
- Concatenation of languages X and Y:

  $$XY = \{uv \mid u \in X, v \in Y\}$$

  X^n denotes the concatenation of X with itself n times
  We define X^0 as $\{\epsilon\}$.
- The **Kleene star** of a set X, written X^*:

  $$X^* = \bigcup_{i=0}^{\infty} X^i$$

- The derived operator $+$, defined as: $X^+ = XX^*$

Operations on Languages

Examples:

- If $L = \{aa, bb\}$, $M = \{c, d\}$ then $LM = \{aac, aad, bbc, bbd\}$
- Powers:
  $\{a, b, ab\}^2 = \{aa, ab, aab, ba, bb, bab, aba, abb, abab\}$
- Kleene star:

$$\begin{aligned} \{a, b\}^* &= \{\epsilon\} \cup \{a, b\} \cup \{aa, ab, ba, bb\} \cup \{aaa, \ldots\} \cup \cdots \\ &= \{\epsilon, a, b, aa, ab, ba, bb, aaa, \ldots\} \end{aligned}$$

- Reversal: $\{ab, cd\}^R = \{ba, dc\}$

Regular Sets / Languages

- Recursively defined over an alphabet Σ from
  - $\emptyset$
  - $\{\epsilon\}$
  - $\{a\}$ for all $a \in \Sigma$

  by applying union, concatenation, and Kleene star.

Regular Expressions: A notation to denote regular languages

- Example: The regular expression

$$\mathtt{a}^*(\mathtt{c} \mid \mathtt{d})\,\mathtt{b}^*$$

  denotes the **regular set**

$$\{a\}^*(\{c\} \cup \{d\})\{b\}^*$$

- The regular expression of a set is not unique

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$?

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$?

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$?

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$? ✓
4. $aabb \in \mathtt{a} \mid (\mathtt{a} \mid \mathtt{b})^*\mathtt{a}$?

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$? ✓
4. $aabb \in \mathtt{a} \mid (\mathtt{a} \mid \mathtt{b})^*\mathtt{a}$? ✗
5. $aabb \in \mathtt{a}(\mathtt{ab})^*\mathtt{b}$?

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$? ✓
4. $aabb \in \mathtt{a} \mid (\mathtt{a} \mid \mathtt{b})^*\mathtt{a}$? ✗
5. $aabb \in \mathtt{a}(\mathtt{ab})^*\mathtt{b}$? ✓
6. $aabb \in \mathtt{a}(\mathtt{abb})^+$?

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$? ✓
4. $aabb \in \mathtt{a} \mid (\mathtt{a} \mid \mathtt{b})^*\mathtt{a}$? ✗
5. $aabb \in \mathtt{a}(\mathtt{ab})^*\mathtt{b}$? ✓
6. $aabb \in \mathtt{a}(\mathtt{abb})^+$? ✓
7. $aabb \in \mathtt{a}(\mathtt{abb})^+\mathtt{b}$?

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$? ✓
4. $aabb \in \mathtt{a} \mid (\mathtt{a} \mid \mathtt{b})^*\mathtt{a}$? ✗
5. $aabb \in \mathtt{a}(\mathtt{ab})^*\mathtt{b}$? ✓
6. $aabb \in \mathtt{a}(\mathtt{abb})^+$? ✓
7. $aabb \in \mathtt{a}(\mathtt{abb})^+\mathtt{b}$? ✗

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$? ✓
4. $aabb \in \mathtt{a} \mid (\mathtt{a} \mid \mathtt{b})^*\mathtt{a}$? ✗
5. $aabb \in \mathtt{a}(\mathtt{ab})^*\mathtt{b}$? ✓
6. $aabb \in \mathtt{a}(\mathtt{abb})^+$? ✓
7. $aabb \in \mathtt{a}(\mathtt{abb})^+\mathtt{b}$? ✗
8. $aabb \in \mathtt{a}^*(\mathtt{ba})^*\mathtt{b}^*$?

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$? ✓
4. $aabb \in \mathtt{a} \mid (\mathtt{a} \mid \mathtt{b})^*\mathtt{a}$? ✗
5. $aabb \in \mathtt{a}(\mathtt{ab})^*\mathtt{b}$? ✓
6. $aabb \in \mathtt{a}(\mathtt{abb})^+$? ✓
7. $aabb \in \mathtt{a}(\mathtt{abb})^+\mathtt{b}$? ✗
8. $aabb \in \mathtt{a}^*(\mathtt{ba})^*\mathtt{b}^*$? ✓

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$? ✓
4. $aabb \in \mathtt{a} \mid (\mathtt{a} \mid \mathtt{b})^*\mathtt{a}$? ✗
5. $aabb \in \mathtt{a}(\mathtt{ab})^*\mathtt{b}$? ✓
6. $aabb \in \mathtt{a}(\mathtt{abb})^+$? ✓
7. $aabb \in \mathtt{a}(\mathtt{abb})^+\mathtt{b}$? ✗
8. $aabb \in \mathtt{a}^*(\mathtt{ba})^*\mathtt{b}^*$? ✓
9. $aabb \in \mathtt{a}^*(\mathtt{ba})^+\mathtt{b}^*$?

Strings and Regular Expressions

1. $aabb \in (\mathtt{a}^*\mathtt{b}^*)\mathtt{b}$? ✓
2. $aabb \in (\mathtt{a}^* \mid \mathtt{b}^*)\mathtt{b}$? ✗
3. $aabb \in \mathtt{b} \mid (\mathtt{b} \mid \mathtt{a})^*$? ✓
4. $aabb \in \mathtt{a} \mid (\mathtt{a} \mid \mathtt{b})^*\mathtt{a}$? ✗
5. $aabb \in \mathtt{a}(\mathtt{ab})^*\mathtt{b}$? ✓
6. $aabb \in \mathtt{a}(\mathtt{abb})^+$? ✓
7. $aabb \in \mathtt{a}(\mathtt{abb})^+\mathtt{b}$? ✗
8. $aabb \in \mathtt{a}^*(\mathtt{ba})^*\mathtt{b}^*$? ✓
9. $aabb \in \mathtt{a}^*(\mathtt{ba})^+\mathtt{b}^*$? ✗

Exercise

Give a regular expression L over $\Sigma = \{a, b, c\}$ that contains every string not containing the substring "ab".

- Strings that do not contain a's are clearly acceptable:

$$(\mathtt{b} \mid \mathtt{c})^* \subseteq L$$

Exercise

Give a regular expression L over $\Sigma = \{a, b, c\}$ that contains every string not containing the substring "ab".

- Strings that do not contain a's are clearly acceptable:

$$(\mathtt{b} \mid \mathtt{c})^* \subseteq L$$

- Strings that contain precisely one a are also acceptable:

$$(\mathtt{b} \mid \mathtt{c})^* \mathtt{a} [\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*] \subseteq L$$

Exercise

Give a regular expression L over $\Sigma = \{a, b, c\}$ that contains every string not containing the substring "ab".

- Strings that do not contain a's are clearly acceptable:

$$(\mathtt{b} \mid \mathtt{c})^* \subseteq L$$

- Strings that contain precisely one a are also acceptable:

$$(\mathtt{b} \mid \mathtt{c})^*\mathtt{a}[\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*] \subseteq L$$

- Strings with a single group of one or more a's :

$$(\mathtt{b} \mid \mathtt{c})^*\mathtt{a}\mathtt{a}^*[\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*] \subseteq L$$

Exercise

Give a regular expression L over $\Sigma = \{a, b, c\}$ that contains every string not containing the substring "ab".

- Strings that do not contain a's are clearly acceptable:

$$(\mathtt{b} \mid \mathtt{c})^* \subseteq L$$

- Strings that contain precisely one a are also acceptable:

$$(\mathtt{b} \mid \mathtt{c})^*\mathtt{a}[\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*] \subseteq L$$

- Strings with a single group of one or more a's :

$$(\mathtt{b} \mid \mathtt{c})^*\mathtt{a}\mathtt{a}^*[\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*] \subseteq L$$

- Strings with two groups of a's:

$$(\mathtt{b} \mid \mathtt{c})^*\mathtt{a}\mathtt{a}^*\mathtt{c}(\mathtt{b} \mid \mathtt{c})^*\mathtt{a}\mathtt{a}^*[\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*] \subseteq L$$

Exercise

Give a regular expression L over $\Sigma = \{a, b, c\}$ that contains every string not containing the substring "ab".

We have seen that:

$$
\begin{aligned}
(\mathtt{b} \mid \mathtt{c})^* \mathtt{a} [\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*] &\subseteq L \\
(\mathtt{b} \mid \mathtt{c})^* \mathtt{a}\mathtt{a}^* [\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*] &\subseteq L \\
(\mathtt{b} \mid \mathtt{c})^* \mathtt{a}\mathtt{a}^* \mathtt{c}(\mathtt{b} \mid \mathtt{c})^* \mathtt{a}\mathtt{a}^* [\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*] &\subseteq L
\end{aligned}
$$

Continuing this line of reasoning we see that

$$
L = (\mathtt{b} \mid \mathtt{c})^* (\epsilon \mid [\mathtt{a}\mathtt{a}^* \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*]^* \mathtt{a}\mathtt{a}^* [\epsilon \mid \mathtt{c}(\mathtt{b} \mid \mathtt{c})^*])
$$

Proofs

- Q: When is a proof correct (enough)?
- A: When it convinces the reader!

Essential elements:

- What do you know?
- What do you want to prove?
- How are you going to prove it?
- The actual, step-by-step, proof—the **proof method**!

  Example: If we have A, then because of B we also have C. Now, because of C and D, we also have E.
- Conclusion! Finally, we see that we must indeed have Z.

Proofs

- Q: When is a proof correct (enough)?
- A: When it convinces the reader!

Essential elements:

- What do you know?
- What do you want to prove?
- How are you going to prove it?
- The actual, step-by-step, proof—the **proof method**!

  Example: If we have A, then because of B we also have C.
  Now, because of C and D, we also have E.
- Conclusion! Finally, we see that we must indeed have Z.

This week's tutorial is on different proof methods (direct, induction, case analysis, contradiction...)

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:

Well, x could not be larger, so the statement must be true.

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:
Well, x could not be larger, so the statement must be true. ✗

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:
Well, x could not be larger ***or smaller****, so the statement must be true.*

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:
Well, x could not be larger ***or smaller****, so the statement must be true.* ✗

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:
We must have $x = 0$.
Suppose $x < 0$*: picking* $y = 1$ *suffices to infer that* $0 \leq x$. *Hence,* $x \not< 0$.
Now suppose $x > 0$*: then picking* $y = x$ *allows us to infer that* $x < x$. *Hence,* $x \not> 0$.

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:
We must have $x = 0$.
Suppose $x < 0$*: picking* $y = 1$ *suffices to infer that* $0 \leq x$. *Hence,* $x \not< 0$.
Now suppose $x > 0$*: then picking* $y = x$ *allows us to infer that* $x < x$. *Hence,* $x \not> 0$. ✗

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:

- First consider $x < 0$. If we pick $y = 1$ then $y > 0$ and we should also have $0 \leq x$. This is clearly contradictory, so $x \not< 0$.
- If $x > 0$ would hold then picking $y = x$ would give us $y > 0$, and so $x < y$ would lead to the contradiction $x < x$. We thus conclude that $x \not> 0$.
- Clearly, we must now have $x = 0$. Indeed we see that if $x = 0$, then $0 \leq x < y$ holds for all $y > 0$.

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:

- First consider $x < 0$. If we pick $y = 1$ then $y > 0$ and we should also have $0 \leq x$. This is clearly contradictory, so $x \not< 0$.
- If $x > 0$ would hold then picking $y = x$ would give us $y > 0$, and so $x < y$ would lead to the contradiction $x < x$. We thus conclude that $x \not> 0$.
- Clearly, we must now have $x = 0$. Indeed we see that if $x = 0$, then $0 \leq x < y$ holds for all $y > 0$.

✗

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:
We proceed by case analysis on x. We consider three cases ($x < 0$, $x > 0$, and $x = 0$), and show that only $x = 0$ can be true:

1. First consider $x < 0$. If we pick $y = 1$ then $y > 0$ and we should also have $0 \leq x$. This is clearly contradictory, so $x \not< 0$.
2. If $x > 0$ would hold then picking $y = x$ would give us $y > 0$, and so $x < y$ would lead to the contradiction $x < x$. We thus conclude that $x \not> 0$.
3. Clearly, we must now have $x = 0$. Indeed we see that if $x = 0$, then $0 \leq x < y$ holds for all $y > 0$.

Q.E.D.

Example

- Given: $x \in \mathbb{R}$ satisfies $(\forall y \in \mathbb{R} : y > 0 \Rightarrow 0 \leq x < y)$
- To prove: $x = 0$

Proof:
We proceed by case analysis on x. We consider three cases ($x < 0$, $x > 0$, and $x = 0$), and show that only $x = 0$ can be true:

1. First consider $x < 0$. If we pick $y = 1$ then $y > 0$ and we should also have $0 \leq x$. This is clearly contradictory, so $x \not< 0$.
2. If $x > 0$ would hold then picking $y = x$ would give us $y > 0$, and so $x < y$ would lead to the contradiction $x < x$. We thus conclude that $x \not> 0$.
3. Clearly, we must now have $x = 0$. Indeed we see that if $x = 0$, then $0 \leq x < y$ holds for all $y > 0$.

Q.E.D. ✓

Proofs: Some Hints

What proof method/technique should you use?

- Direct proof difficult $\rightarrow$ Proof by contradiction
- Equivalence or set equality $\rightarrow$ Split into two implications
- Recursive definition $\rightarrow$ Proof by induction
- General case too hard $\rightarrow$ Case analysis
- Show something is *not* true $\rightarrow$ Contradiction + counter example

The way you present and structure your proofs is important!

Preview: Context-Free Languages

- Give a regular expression for $L = \{a^k b^k \mid k \in \mathbb{N}\}$

Preview: Context-Free Languages

- Give a regular expression for $L = \{a^k b^k \mid k \in \mathbb{N}\}$
- Impossible! The expression $\mathtt{a}^*\mathtt{b}^*$ does *not* work.
- Consider the grammar G given by

$$S \quad \rightarrow \quad \epsilon \mid aSb$$

- To show that $aabb \in L(G)$, we can write the derivation

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

- Equivalently, we can draw the corresponding *derivation tree*.

Taking Stock

- Basic notations
- Regular languages and regular notations
- Proofs
- There are non-regular languages:
  Context-free languages to the rescue!