

Domain Specific Formal Languages

Preliminaries

Francesco Tiezzi

University of Camerino
francesco.tiezzi@unicam.it

A.A. 2017/2018



Set Notation

$A \subseteq B$ every element of A is in B

$A \subset B$ if $A \subseteq B$ and there is one element of B not in A

$A \subseteq B$ and $B \subseteq A$ implies $A = B$

$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$ $(\bigcup_{i \in I} A_i)$

$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$ $(\bigcap_{i \in I} A_i)$

$A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}$

$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$ *ordered pairs* $(\times_{i=1}^n A_i)$

$2^A = \{X \mid X \subseteq A\}$ *powerset*

Set Notation

$A \subseteq B$ every element of A is in B

$A \subset B$ if $A \subseteq B$ and there is one element of B not in A

$A \subseteq B$ and $B \subseteq A$ implies $A = B$

$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$ $(\bigcup_{i \in I} A_i)$

$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$ $(\bigcap_{i \in I} A_i)$

$A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}$

$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$ *ordered pairs* $(\times_{i=1}^n A_i)$

$2^A = \{X \mid X \subseteq A\}$ *powerset*

Relations

$R \subseteq A \times B$ is a relation on sets A and B

$(R \subseteq \times_{i=1}^n A_i)$

$(a, b) \in R \equiv R(a, b) \equiv aRb$ notation

$Id_A = \{(a, a) \mid a \in A\}$ (identity)

$R^{-1} = \{(y, x) \mid (x, y) \in R\} \subseteq B \times A$ (inverse)

$R_1 \cdot R_2 = \{(x, z) \mid \exists y \in B. (x, y) \in R_1 \wedge (y, z) \in R_2\} \subseteq A \times C$ (composition)

Some basic constructions:

$$\begin{aligned} R^0 &= Id_A \\ R^{n+1} &= R \cdot R^n \\ R^* &= \bigcup_{n \geq 0} R^n \\ R^+ &= \bigcup_{n \geq 1} R^n \end{aligned}$$

Note that: $R^1 = R \cdot R^0 = R$, $R^* = Id_A \cup R^+$ and

$R^+ = \{(x, y) \mid \exists n, \exists x_1, \dots, x_n \text{ with } x_i R x_{i+1} (1 \leq i \leq n-1), x_1 = x, x_n = y\}$

Relations

$R \subseteq A \times B$ is a relation on sets A and B

$(R \subseteq \times_{i=1}^n A_i)$

$(a, b) \in R \equiv R(a, b) \equiv aRb$ notation

$Id_A = \{(a, a) \mid a \in A\}$ (identity)

$R^{-1} = \{(y, x) \mid (x, y) \in R\} \subseteq B \times A$ (inverse)

$R_1 \cdot R_2 = \{(x, z) \mid \exists y \in B. (x, y) \in R_1 \wedge (y, z) \in R_2\} \subseteq A \times C$ (composition)

Some basic constructions:

$$\begin{aligned} R^0 &= Id_A \\ R^{n+1} &= R \cdot R^n \\ R^* &= \bigcup_{n \geq 0} R^n \\ R^+ &= \bigcup_{n \geq 1} R^n \end{aligned}$$

Note that: $R^1 = R \cdot R^0 = R$, $R^* = Id_A \cup R^+$ and

$R^+ = \{(x, y) \mid \exists n, \exists x_1, \dots, x_n \text{ with } x_i R x_{i+1} (1 \leq i \leq n-1), x_1 = x, x_n = y\}$

Relations

$R \subseteq A \times B$ is a relation on sets A and B

$$(R \subseteq \times_{i=1}^n A_i)$$

$$(a, b) \in R \equiv R(a, b) \equiv aRb \quad \text{notation}$$

$$Id_A = \{(a, a) \mid a \in A\} \quad \text{(identity)}$$

$$R^{-1} = \{(y, x) \mid (x, y) \in R\} \subseteq B \times A \quad \text{(inverse)}$$

$$R_1 \cdot R_2 = \{(x, z) \mid \exists y \in B. (x, y) \in R_1 \wedge (y, z) \in R_2\} \subseteq A \times C \quad \text{(composition)}$$

Some basic constructions:

$$\begin{aligned} R^0 &= Id_A \\ R^{n+1} &= R \cdot R^n \\ R^* &= \bigcup_{n \geq 0} R^n \\ R^+ &= \bigcup_{n \geq 1} R^n \end{aligned}$$

Note that: $R^1 = R \cdot R^0 = R$, $R^* = Id_A \cup R^+$ and

$$R^+ = \{(x, y) \mid \exists n, \exists x_1, \dots, x_n \text{ with } x_i R x_{i+1} \ (1 \leq i \leq n-1), x_1 = x, x_n = y\}$$

Properties of Relations

Binary Relations

A binary relation $R \subseteq A \times A$ is (same set A)

reflexive: if $\forall x \in A, (x, x) \in R,$

symmetric: if $\forall x, y \in A, (x, y) \in R \Rightarrow (y, x) \in R,$

antisymmetric: if $\forall x, y \in A, (x, y) \in R \wedge (y, x) \in R \Rightarrow x = y;$

transitive: if $\forall x, y, z \in A, (x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$

Closure of Relations

$S = R \cup Id_A$ the reflexive closure of R

$S = R \cup R^{-1}$ the symmetric closure of R

$S = R^+$ the transitive closure of R

$S = R^*$ the reflexive and transitive closure of R

Properties of Relations

Binary Relations

A binary relation $R \subseteq A \times A$ is (same set A)

reflexive: if $\forall x \in A, (x, x) \in R,$

symmetric: if $\forall x, y \in A, (x, y) \in R \Rightarrow (y, x) \in R,$

antisymmetric: if $\forall x, y \in A, (x, y) \in R \wedge (y, x) \in R \Rightarrow x = y;$

transitive: if $\forall x, y, z \in A, (x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$

Closure of Relations

$S = R \cup Id_A$ the reflexive closure of R

$S = R \cup R^{-1}$ the symmetric closure of R

$S = R^+$ the transitive closure of R

$S = R^*$ the reflexive and transitive closure of R

Special Relations

A relation R is

- an **order** if it is reflexive, antisymmetric and transitive
- an **equivalence** if it is reflexive, symmetric and transitive
- a **preorder** if it is reflexive and transitive

Examples

- **orders**: less-than-or-equal-to (\leq) on \mathbb{R} , set inclusion (\subseteq), ...
- **equivalences**: equal-to ($=$) on \mathbb{R} , congruent-mod- n , ...
- **preorders**: reachability in directed graphs, some subtyping, ...

Kernel relation

- Given a preorder R its kernel, defined as $K = R \cap R^{-1}$, is an equivalence relation

Special Relations

A relation R is

- an **order** if it is reflexive, antisymmetric and transitive
- an **equivalence** if it is reflexive, symmetric and transitive
- a **preorder** if it is reflexive and transitive

Examples

- **orders**: less-than-or-equal-to (\leq) on \mathbb{R} , set inclusion (\subseteq), ...
- **equivalences**: equal-to ($=$) on \mathbb{R} , congruent-mod- n , ...
- **preorders**: reachability in directed graphs, some subtyping, ...

Kernel relation

- Given a preorder R its kernel, defined as $K = R \cap R^{-1}$, is an equivalence relation

Special Relations

A relation R is

- an **order** if it is reflexive, antisymmetric and transitive
- an **equivalence** if it is reflexive, symmetric and transitive
- a **preorder** if it is reflexive and transitive

Examples

- **orders**: less-than-or-equal-to (\leq) on \mathbb{R} , set inclusion (\subseteq), ...
- **equivalences**: equal-to ($=$) on \mathbb{R} , congruent-mod- n , ...
- **preorders**: reachability in directed graphs, some subtyping, ...

Kernel relation

- Given a preorder R its kernel, defined as $K = R \cap R^{-1}$, is an equivalence relation

Equivalence Classes and Quotient Set

Examples of **equivalence relations**: $R \subseteq A \times A$ (reflexive, symmetric, transitive)

Example: $R = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid (x \equiv y) \pmod{3}\}$

Equivalence Classes and Quotient Set

Examples of **equivalence relations**: $R \subseteq A \times A$ (reflexive, symmetric, transitive)

Example: $R = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid (x \equiv y) \pmod{3}\}$

$R(7, 7)$,

Equivalence Classes and Quotient Set

Examples of **equivalence relations**: $R \subseteq A \times A$ (reflexive, symmetric, transitive)

Example: $R = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid (x \equiv y) \pmod{3}\}$

$R(7, 7), R(7, 1), R(1, 7),$

Equivalence Classes and Quotient Set

Examples of **equivalence relations**: $R \subseteq A \times A$ (reflexive, symmetric, transitive)

Example: $R = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid (x \equiv y) \pmod{3}\}$

$R(7, 7), R(7, 1), R(1, 7), R(7, 10), R(1, 10)$

Equivalence Classes and Quotient Set

Examples of **equivalence relations**: $R \subseteq A \times A$ (reflexive, symmetric, transitive)

Example: $R = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid (x \equiv y) \pmod{3}\}$

$R(7, 7), R(7, 1), R(1, 7), R(7, 10), R(1, 10), \dots$

Equivalence Classes and Quotient Set

Examples of **equivalence relations**: $R \subseteq A \times A$ (reflexive, symmetric, transitive)

Example: $R = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid (x \equiv y) \pmod{3}\}$

$R(7, 7), R(7, 1), R(1, 7), R(7, 10), R(1, 10), \dots$

$[0] = \{0, 3, 6, 9, \dots\}$

equivalence classes:

$[1] = \{1, 4, 7, 10, \dots\}$

- *have a representative*

$[2] = \{2, 5, 8, 11, \dots\}$

- *are disjoint*

Equivalence Classes and Quotient Set

Examples of **equivalence relations**: $R \subseteq A \times A$ (reflexive, symmetric, transitive)

Example: $R = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid (x \equiv y) \pmod{3}\}$

$R(7, 7), R(7, 1), R(1, 7), R(7, 10), R(1, 10), \dots$

$[0] = \{0, 3, 6, 9, \dots\}$

equivalence classes:

$[1] = \{1, 4, 7, 10, \dots\}$

- have a representative

$[2] = \{2, 5, 8, 11, \dots\}$

- are disjoint

An **equivalence class** is a subset C of A such that

$x, y \in C \Rightarrow (x, y) \in R$ *consistent* and

$x \in C \wedge (x, y) \in R \Rightarrow y \in C$ *saturated*

Equivalence Classes and Quotient Set

Examples of **equivalence relations**: $R \subseteq A \times A$ (reflexive, symmetric, transitive)

Example: $R = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid (x \equiv y) \pmod{3}\}$

$R(7, 7), R(7, 1), R(1, 7), R(7, 10), R(1, 10), \dots$

$[0] = \{0, 3, 6, 9, \dots\}$

equivalence classes:

$[1] = \{1, 4, 7, 10, \dots\}$

- *have a representative*

$[2] = \{2, 5, 8, 11, \dots\}$

- *are disjoint*

The **quotient set** Q_A^R of A modulo R
is the set of equivalence classes induced by R on A

is a partition of A

Example: $R = \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid (x \equiv y) \pmod{3}\}$

$Q_{\mathbb{N}}^R = \{[0], [1], [2]\}$

Functions

Partial Functions

A *partial function* is a relation $f \subseteq A \times B$ such that

$$\forall x, y, z. (x, y) \in f \wedge (x, z) \in f \Rightarrow y = z$$

We denote partial function by $f : A \rightarrow B$

Total Functions

A (total) *function* is a partial function $f : A \rightarrow B$ such that

$$\forall x \exists y. (x, y) \in f$$

We denote total function by $f : A \rightarrow B$

Functions (total or partial) can be *monotone*, *continuous*, *injective*, *surjective*, *bijective*, *invertible*...

Functions

Partial Functions

A *partial function* is a relation $f \subseteq A \times B$ such that

$$\forall x, y, z. (x, y) \in f \wedge (x, z) \in f \Rightarrow y = z$$

We denote partial function by $f : A \rightarrow B$

Total Functions

A (total) *function* is a partial function $f : A \rightarrow B$ such that

$$\forall x \exists y. (x, y) \in f$$

We denote total function by $f : A \rightarrow B$

Functions (total or partial) can be *monotone, continuous, injective, surjective, bijective, invertible...*

Functions

Partial Functions

A *partial function* is a relation $f \subseteq A \times B$ such that

$$\forall x, y, z. (x, y) \in f \wedge (x, z) \in f \Rightarrow y = z$$

We denote partial function by $f : A \rightarrow B$

Total Functions

A (total) *function* is a partial function $f : A \rightarrow B$ such that

$$\forall x \exists y. (x, y) \in f$$

We denote total function by $f : A \rightarrow B$

Functions (total or partial) can be *monotone*, *continuous*, *injective*, *surjective*, *bijjective*, *invertible*...

Induction Principle

Mathematical Induction

To prove that $P(n)$ holds for every natural number $n \in \mathbb{N}$, prove

- 1 $P(0)$
- 2 for any $k \in \mathbb{N}$, $P(k)$ implies $P(k + 1)$

Example: Show that $sum(n) = \sum_{i=1}^n i = \frac{n(n+1)}{2}$ for every $n \in \mathbb{N}$

Induction Principle

Mathematical Induction

To prove that $P(n)$ holds for every natural number $n \in \mathbb{N}$, prove

- 1 $P(0)$
- 2 for any $k \in \mathbb{N}$, $P(k)$ implies $P(k + 1)$

Example: Show that $sum(n) = \sum_{i=1}^n i = \frac{n(n+1)}{2}$ for every $n \in \mathbb{N}$

Induction Principle

Mathematical Induction

To prove that $P(n)$ holds for every natural number $n \in \mathbb{N}$, prove

- 1 $P(0)$
- 2 for any $k \in \mathbb{N}$, $P(k)$ implies $P(k + 1)$

Example: Show that $sum(n) = \sum_{i=1}^n i = \frac{n(n+1)}{2}$ for every $n \in \mathbb{N}$

$$(1) \text{ } sum(0) = \frac{0(0+1)}{2} = 0$$

base case

Induction Principle

Mathematical Induction

To prove that $P(n)$ holds for every natural number $n \in \mathbb{N}$, prove

- 1 $P(0)$
- 2 for any $k \in \mathbb{N}$, $P(k)$ implies $P(k + 1)$

Example: Show that $sum(n) = \sum_{i=1}^n i = \frac{n(n+1)}{2}$ for every $n \in \mathbb{N}$

(1) $sum(0) = \frac{0(0+1)}{2} = 0$ *base case*

(2) to show: $\sum_{i=1}^n i = \frac{n(n+1)}{2}$ implies $\sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2}$

assume $sum(n) = \frac{n(n+1)}{2}$, for a generic n

$sum(n + 1) = sum(n) + (n + 1) =$ *properties of summation*

$= \frac{n(n+1)}{2} + (n + 1)$ *inductive hypothesis*

$= \frac{(n+1)(n+2)}{2}$ *qed*

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Playful digression

Some “advanced” proof methods

- 1 **Proof by obviousness:** So evident it need not to be mentioned
- 2 **Proof by general agreement:** All in favor?
- 3 **Proof by majority:** When general agreement fails
- 4 **Proof by plausibility:** It sounds good
- 5 **Proof by intuition:** I have this feeling. . .
- 6 **Proof by lost reference:** I saw it somewhere
- 7 **Proof by obscure reference:** It appeared in the Annals of Polish Math. Soc. (1854, in polish)
- 8 **Proof by logic:** It is on the textbook, hence it must be true
- 9 **Proof by intimidation:** Who is saying that it is false!?
- 10 **Proof by authority:** Don Knuth said it was true
- 11 **Proof by deception:** Everybody please turn their backs. . .
- 12 **Proof by divine word:** Lord said let it be true

Inductively Defined Sets

basis: the set I of initial elements of S

induction: rules R for constructing elements in S from elements in S

closure: S is the least set containing I and closed w.r.t. R

Natural numbers

$I = \{0\}$, R_1 : if $X \in S$ then $s(X) \in S$

$S = \{0, s(0), s(s(0)), \dots\}$

$S = Lists(\mathbb{N})$, lists of numbers in \mathbb{N}

$I = \{[]\}$, R_1 : if $X \in S$ and $n \in \mathbb{N}$ then $[n|X] \in S$

$S = \{[], [0], [1], [2], \dots, [0, 0], [0, 1], [0, 2], \dots, [1, 0], [1, 1], [1, 2], \dots\}$

n-ary trees

$I = \{\varepsilon\}$, R_1 : if $X_1, \dots, X_n \in S$ then $t(X_1, \dots, X_n) \in S$

$S = \{\varepsilon, t(\varepsilon), t(\varepsilon, \varepsilon), \dots, t(t(\varepsilon)), \dots, t(\varepsilon, t(t(\varepsilon), \varepsilon)), t(\varepsilon, \varepsilon, \varepsilon)), \dots\}$

Inductively Defined Sets

basis: the set I of initial elements of S

induction: rules R for constructing elements in S from elements in S

closure: S is the least set containing I and closed w.r.t. R

Natural numbers

$I = \{0\}$, R_1 : if $X \in S$ then $s(X) \in S$

$S = \{0, s(0), s(s(0)), \dots\}$

$S = Lists(\mathbb{N})$, lists of numbers in \mathbb{N}

$I = \{[]\}$, R_1 : if $X \in S$ and $n \in \mathbb{N}$ then $[n|X] \in S$

$S = \{[], [0], [1], [2], \dots, [0, 0], [0, 1], [0, 2], \dots, [1, 0], [1, 1], [1, 2], \dots\}$

n-ary trees

$I = \{\varepsilon\}$, R_1 : if $X_1, \dots, X_n \in S$ then $t(X_1, \dots, X_n) \in S$

$S = \{\varepsilon, t(\varepsilon), t(\varepsilon, \varepsilon), \dots, t(t(\varepsilon)), \dots, t(\varepsilon, t(t(\varepsilon), \varepsilon)), t(\varepsilon, \varepsilon, \varepsilon)), \dots\}$

Inductively Defined Sets

basis: the set I of initial elements of S

induction: rules R for constructing elements in S from elements in S

closure: S is the least set containing I and closed w.r.t. R

Natural numbers

$I = \{0\}$, R_1 : if $X \in S$ then $s(X) \in S$

$S = \{0, s(0), s(s(0)), \dots\}$

$S = Lists(\mathbb{N})$, lists of numbers in \mathbb{N}

$I = \{[]\}$, R_1 : if $X \in S$ and $n \in \mathbb{N}$ then $[n|X] \in S$

$S = \{[], [0], [1], [2], \dots, [0, 0], [0, 1], [0, 2], \dots, [1, 0], [1, 1], [1, 2], \dots\}$

n-ary trees

$I = \{\varepsilon\}$, R_1 : if $X_1, \dots, X_n \in S$ then $t(X_1, \dots, X_n) \in S$

$S = \{\varepsilon, t(\varepsilon), t(\varepsilon, \varepsilon), \dots, t(t(\varepsilon)), \dots, t(\varepsilon, t(t(\varepsilon), \varepsilon)), t(\varepsilon, \varepsilon, \varepsilon)), \dots\}$

Inductively Defined Sets

basis: the set I of initial elements of S

induction: rules R for constructing elements in S from elements in S

closure: S is the least set containing I and closed w.r.t. R

Natural numbers

$I = \{0\}$, R_1 : if $X \in S$ then $s(X) \in S$

$S = \{0, s(0), s(s(0)), \dots\}$

$S = Lists(\mathbb{N})$, lists of numbers in \mathbb{N}

$I = \{[]\}$, R_1 : if $X \in S$ and $n \in \mathbb{N}$ then $[n|X] \in S$

$S = \{[], [0], [1], [2], \dots, [0, 0], [0, 1], [0, 2], \dots, [1, 0], [1, 1], [1, 2], \dots\}$

n-ary trees

$I = \{\varepsilon\}$, R_1 : if $X_1, \dots, X_n \in S$ then $t(X_1, \dots, X_n) \in S$

$S = \{\varepsilon, t(\varepsilon), t(\varepsilon, \varepsilon), \dots, t(t(\varepsilon)), \dots, t(\varepsilon, t(t(\varepsilon), \varepsilon)), t(\varepsilon, \varepsilon, \varepsilon)), \dots\}$

Structural Induction

Let us consider a set S inductively defined by a set $C = \{c_1, \dots, c_n\}$ of constructors of arity $\{a_1, \dots, a_n\}$ with

- $I = \{c_i() \mid a_i = 0\}$
- R_i : if $X_1, \dots, X_{a_i} \in S$ then $c_i(X_1, \dots, X_{a_i}) \in S$

To prove that $P(x)$ holds for every $x \in S$, it is sufficient to prove that

- for every constructor $c_k \in C$ and
- for every $s_1, \dots, s_k \in S$, where k is the arity of c_k

$$P(s_1), \dots, P(s_k) \implies P(c_k(s_1, \dots, s_k))$$

Notice that the base case is the one dealing with **constructors of arity 0**
i.e. with **constants**

Structural Induction: example

Prove that $sum(\ell) \leq max(\ell) * len(\ell)$, for every $\ell \in Lists(\mathbb{N})$

where

- $sum(\ell)$ is the sum of the elements in the list ℓ
- $max(\ell)$ is the greatest element in ℓ (with $max([]) = 0$)
- $len(\ell)$ is the number of elements in ℓ

Structural Induction: example

Exercise: prove $sum(\ell) \leq max(\ell) * len(\ell)$, for every $\ell \in Lists(\mathbb{N})$

$$sum([]) = 0$$

$$len([]) = 0$$

$$sum([n|X]) = n + sum(X) \quad len([n|X]) = 1 + len(X)$$

$$max([]) = 0$$

$$max([n|X]) = n \quad \text{if } max(X) \leq n$$

$$max([n|X]) = max(X) \quad \text{if } n < max(X)$$

Structural Induction: example

Exercise: prove $sum(\ell) \leq max(\ell) * len(\ell)$, for every $\ell \in Lists(\mathbb{N})$

$$sum([]) = 0$$

$$len([]) = 0$$

$$sum([n|X]) = n + sum(X) \quad len([n|X]) = 1 + len(X)$$

$$max([]) = 0$$

$$max([n|X]) = n \quad \text{if } max(X) \leq n$$

$$max([n|X]) = max(X) \quad \text{if } n < max(X)$$

(1) $sum([]) \leq max([]) * len([])$

$$0 \leq 0 * 0$$

applying definitions

Structural Induction: example

Exercise: prove $sum(\ell) \leq max(\ell) * len(\ell)$, for every $\ell \in Lists(\mathbb{N})$

$$sum([]) = 0$$

$$len([]) = 0$$

$$sum([n|X]) = n + sum(X) \quad len([n|X]) = 1 + len(X)$$

$$max([]) = 0$$

$$max([n|X]) = n \quad \text{if } max(X) \leq n$$

$$max([n|X]) = max(X) \quad \text{if } n < max(X)$$

(1) $sum([]) \leq max([]) * len([])$

$$0 \leq 0 * 0$$

applying definitions

(2) assume $sum(\ell) \leq max(\ell) * len(\ell)$

inductive hyp.

prove $sum([n|\ell]) \leq max([n|\ell]) * len([n|\ell])$ for any $n \in \mathbb{N}$

Structural Induction: example

Exercise: prove $sum(\ell) \leq max(\ell) * len(\ell)$, for every $\ell \in Lists(\mathbb{N})$

$$sum([]) = 0$$

$$len([]) = 0$$

$$sum([n|X]) = n + sum(X) \quad len([n|X]) = 1 + len(X)$$

$$max([]) = 0$$

$$max([n|X]) = n \quad \text{if } max(X) \leq n \quad (a)$$

$$max([n|X]) = max(X) \quad \text{if } n < max(X)$$

(1) $sum([]) \leq max([]) * len([])$

$$0 \leq 0 * 0$$

applying definitions

(2) assume $sum(\ell) \leq max(\ell) * len(\ell)$

inductive hyp.

prove $sum([n|\ell]) \leq max([n|\ell]) * len([n|\ell])$ for any $n \in \mathbb{N}$

(a) $n + sum(\ell) \leq n * (1 + len(\ell))$ if $max(\ell) \leq n$ *applying definitions*

$$sum(\ell) \leq_{hyp} max(\ell) * len(\ell) \leq_{(a)} n * len(\ell) \quad QED$$

Structural Induction: example

Exercise: prove $sum(\ell) \leq max(\ell) * len(\ell)$, for every $\ell \in Lists(\mathbb{N})$

$$sum([]) = 0$$

$$len([]) = 0$$

$$sum([n|X]) = n + sum(X) \quad len([n|X]) = 1 + len(X)$$

$$max([]) = 0$$

$$max([n|X]) = n \quad \text{if } max(X) \leq n \quad \text{(a)}$$

$$max([n|X]) = max(X) \quad \text{if } n < max(X) \quad \text{(b)}$$

(1) $sum([]) \leq max([]) * len([])$

$$0 \leq 0 * 0$$

applying definitions

(2) assume $sum(\ell) \leq max(\ell) * len(\ell)$

inductive hyp.

prove $sum([n|\ell]) \leq max([n|\ell]) * len([n|\ell])$ for any $n \in \mathbb{N}$

(a) $n + sum(\ell) \leq n * (1 + len(\ell))$ if $max(\ell) \leq n$ *applying definitions*

$$sum(\ell) \leq_{hyp} max(\ell) * len(\ell) \leq_{(a)} n * len(\ell) \quad \text{QED}$$

(b) $n + sum(\ell) \leq max(\ell) + max(\ell) * len(\ell)$ if $n < max(\ell)$ *applying definitions*

$$A \leq B \text{ and } C \leq D \text{ imply } A + C \leq B + D \quad \text{QED}$$

Inference Systems

① I can be written as $\frac{}{t}$ (for any $t \in I$)

② R_i can be written as $\frac{p_1 \cdots p_n}{q}$

Meaning: $\vdash t$ and if $\vdash p_1, \dots, \vdash p_n$ then $\vdash q$

Example: rational numbers \mathbb{Q}

$$\frac{}{0 \in \mathbb{N}} \quad \frac{}{1 \in \mathbb{D}} \quad \frac{k \in \mathbb{N}}{k+1 \in \mathbb{N}} \quad \frac{k \in \mathbb{D}}{k+1 \in \mathbb{D}} \quad \frac{k \in \mathbb{N}, h \in \mathbb{D}}{k/h \in \mathbb{Q}}$$

A derivation:

$$\frac{\frac{0 \in \mathbb{N} \quad 1 \in \mathbb{D}}{1 \in \mathbb{N}} \quad 2 \in \mathbb{D}}{1/2 \in \mathbb{Q}}$$

$$\boxed{\vdash 1/2 \in \mathbb{Q}}$$

Question:

why do we need the rules in Red?

Inference Systems

① I can be written as $\frac{}{t}$ (for any $t \in I$)

② R_i can be written as $\frac{p_1 \cdots p_n}{q}$

Meaning: $\vdash t$ and if $\vdash p_1, \dots, \vdash p_n$ then $\vdash q$

Example: rational numbers \mathbb{Q}

$$\frac{}{0 \in N} \quad \frac{}{1 \in D} \quad \frac{k \in N}{k+1 \in N} \quad \frac{k \in D}{k+1 \in D} \quad \frac{k \in N, h \in D}{k/h \in \mathbb{Q}}$$

A derivation:

$$\frac{\frac{}{0 \in N} \quad \frac{}{1 \in D}}{1 \in N} \quad \frac{}{2 \in D}}{1/2 \in \mathbb{Q}}$$

$$\boxed{\vdash 1/2 \in \mathbb{Q}}$$

Question:
why do we
need the rules
in Red?

Inference Systems

① I can be written as $\frac{}{t}$ (for any $t \in I$)

② R_i can be written as $\frac{p_1 \cdots p_n}{q}$

Meaning: $\vdash t$ and if $\vdash p_1, \dots, \vdash p_n$ then $\vdash q$

Example: rational numbers \mathbb{Q}

$$\frac{}{0 \in N} \quad \frac{}{1 \in D} \quad \frac{k \in N}{k+1 \in N} \quad \frac{k \in D}{k+1 \in D} \quad \frac{k \in N, h \in D}{k/h \in \mathbb{Q}}$$

A derivation:

$$\frac{\frac{0 \in N}{1 \in N} \quad \frac{1 \in D}{2 \in D}}{1/2 \in \mathbb{Q}}$$

$\vdash 1/2 \in \mathbb{Q}$

Question:

why do we need the rules in Red?

More on Inductively Defined Sets

- $S_{I,R} = \{x \mid \vdash x\}$ *the set of all **finutely** derivable elements*
- $R(X) = \{y \mid \frac{x_1 \cdots x_n}{y} \text{ and } x_1, \dots, x_n \in X\}$ *one step derivation*

X is **closed** under R if $R(X) \subseteq X$ *called a (pre-)fixed point*

R is **monotonic** if $A \subseteq B \Rightarrow R(A) \subseteq R(B)$

$$\begin{aligned} S^0 &= R^0(\emptyset) &= \emptyset \\ S^1 &= R^1(\emptyset) &= R(\emptyset) \\ S^2 &= R^2(\emptyset) &= R(R(\emptyset)) \end{aligned} \qquad S^0 \subseteq S^1 \subseteq S^2 \subseteq \dots$$

$$\begin{aligned} &\vdots \\ S &\triangleq \bigcup_{i \in \mathbb{N}} S^i \end{aligned}$$

S closed under R $R(S) = S$ S least R -closed set

More on Inductively Defined Sets

- $S_{I,R} = \{x \mid \vdash x\}$ *the set of all **finutely** derivable elements*
- $R(X) = \{y \mid \frac{x_1 \cdots x_n}{y} \text{ and } x_1, \dots, x_n \in X\}$ *one step derivation*

X is **closed** under R if $R(X) \subseteq X$ *called a (pre-)fixed point*

R is **monotonic** if $A \subseteq B \Rightarrow R(A) \subseteq R(B)$

$$\begin{aligned} S^0 &= R^0(\emptyset) &= \emptyset \\ S^1 &= R^1(\emptyset) &= R(\emptyset) \\ S^2 &= R^2(\emptyset) &= R(R(\emptyset)) \end{aligned} \qquad S^0 \subseteq S^1 \subseteq S^2 \subseteq \dots$$

\vdots

$$S \triangleq \bigcup_{i \in \mathbb{N}} S^i$$

S closed under R $R(S) = S$ S least R -closed set

Constructing Inductively Defined Sets – an example

$$\text{fib}(0) = 0$$

$$\text{fib}(1) = 1$$

$$\text{fib}(n + 2) = \text{fib}(n + 1) + \text{fib}(n)$$

$$\text{fib} : \mathbb{N} \rightarrow \mathbb{N}$$

$$\frac{}{(0, 0) \in \text{Fib}} \quad \frac{}{(1, 1) \in \text{Fib}} \quad \frac{(n + 1, a) \in \text{Fib} \quad (n, b) \in \text{Fib}}{(n + 2, a + b) \in \text{Fib}}$$

$$R(X) = \{y \mid \frac{x_1 \cdots x_n}{y} \text{ and } x_1, \dots, x_n \in X\}$$

one step derivation

$$\begin{aligned} S^0 &= \emptyset &= \emptyset \\ S^1 &= R(S^0) &= \{(0, 0), (1, 1)\} \\ S^2 &= R(S^1) &= \{(0, 0), (1, 1), (2, 1)\} \\ S^3 &= R(S^2) &= \{(0, 0), (1, 1), (2, 1), (3, 2)\} \\ S^4 &= R(S^3) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3)\} \\ S^5 &= R(S^4) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5)\} \\ S^6 &= R(S^5) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8)\} \\ S^7 &= R(S^6) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8), (7, 13)\} \\ &\vdots & \end{aligned}$$

$$S^0 \subseteq S^1 \subseteq S^2 \subseteq \dots$$

a sequence of partial functions (under-) approximating *fib*

Constructing Inductively Defined Sets – an example

$$\text{fib}(0) = 0$$

$$\text{fib}(1) = 1$$

$$\text{fib}(n + 2) = \text{fib}(n + 1) + \text{fib}(n)$$

$$\text{fib} : \mathbb{N} \rightarrow \mathbb{N}$$

$$\frac{}{(0, 0) \in \text{Fib}} \quad \frac{}{(1, 1) \in \text{Fib}} \quad \frac{(n + 1, a) \in \text{Fib} \quad (n, b) \in \text{Fib}}{(n + 2, a + b) \in \text{Fib}}$$

$$R(X) = \{y \mid \frac{x_1 \cdots x_n}{y} \text{ and } x_1, \dots, x_n \in X\}$$

one step derivation

$$\begin{aligned} S^0 &= \emptyset &= \emptyset \\ S^1 &= R(S^0) &= \{(0, 0), (1, 1)\} \\ S^2 &= R(S^1) &= \{(0, 0), (1, 1), (2, 1)\} \\ S^3 &= R(S^2) &= \{(0, 0), (1, 1), (2, 1), (3, 2)\} \\ S^4 &= R(S^3) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3)\} \\ S^5 &= R(S^4) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5)\} \\ S^6 &= R(S^5) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8)\} \\ S^7 &= R(S^6) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8), (7, 13)\} \\ &\vdots & \end{aligned}$$

$$S^0 \subseteq S^1 \subseteq S^2 \subseteq \dots$$

a sequence of partial functions (under-) approximating *fib*

Constructing Inductively Defined Sets – an example

$$\text{fib}(0) = 0$$

$$\text{fib}(1) = 1$$

$$\text{fib}(n+2) = \text{fib}(n+1) + \text{fib}(n)$$

$$\text{fib} : \mathbb{N} \rightarrow \mathbb{N}$$

$$\frac{}{(0, 0) \in \text{Fib}} \quad \frac{}{(1, 1) \in \text{Fib}} \quad \frac{(n+1, a) \in \text{Fib} \quad (n, b) \in \text{Fib}}{(n+2, a+b) \in \text{Fib}}$$

$$R(X) = \{y \mid \frac{x_1 \cdots x_n}{y} \text{ and } x_1, \dots, x_n \in X\}$$

one step derivation

$$\begin{aligned} S^0 &= \emptyset &= \emptyset \\ S^1 &= R(S^0) &= \{(0, 0), (1, 1)\} \\ S^2 &= R(S^1) &= \{(0, 0), (1, 1), (2, 1)\} \\ S^3 &= R(S^2) &= \{(0, 0), (1, 1), (2, 1), (3, 2)\} \\ S^4 &= R(S^3) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3)\} \\ S^5 &= R(S^4) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5)\} \\ S^6 &= R(S^5) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8)\} \\ S^7 &= R(S^6) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8), (7, 13)\} \\ &\vdots & \end{aligned}$$

$$S^0 \subseteq S^1 \subseteq S^2 \subseteq \dots$$

a sequence of partial functions (under-) approximating *fib*

Constructing Inductively Defined Sets – an example

$$\text{fib}(0) = 0$$

$$\text{fib}(1) = 1$$

$$\text{fib}(n + 2) = \text{fib}(n + 1) + \text{fib}(n)$$

$$\text{fib} : \mathbb{N} \rightarrow \mathbb{N}$$

$$\frac{}{(0, 0) \in \text{Fib}} \quad \frac{}{(1, 1) \in \text{Fib}} \quad \frac{(n + 1, a) \in \text{Fib} \quad (n, b) \in \text{Fib}}{(n + 2, a + b) \in \text{Fib}}$$

$$R(X) = \{y \mid \frac{x_1 \cdots x_n}{y} \text{ and } x_1, \dots, x_n \in X\}$$

one step derivation

$$\begin{aligned} S^0 &= \emptyset &= \emptyset \\ S^1 &= R(S^0) &= \{(0, 0), (1, 1)\} \\ S^2 &= R(S^1) &= \{(0, 0), (1, 1), (2, 1)\} \\ S^3 &= R(S^2) &= \{(0, 0), (1, 1), (2, 1), (3, 2)\} \\ S^4 &= R(S^3) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3)\} \\ S^5 &= R(S^4) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5)\} \\ S^6 &= R(S^5) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8)\} \\ S^7 &= R(S^6) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8), (7, 13)\} \\ &\vdots & \end{aligned}$$

a sequence of partial functions (under-) approximating *fib*

$$S^0 \subseteq S^1 \subseteq S^2 \subseteq \dots$$

Constructing Inductively Defined Sets – an example

$$\text{fib}(0) = 0$$

$$\text{fib}(1) = 1$$

$$\text{fib}(n+2) = \text{fib}(n+1) + \text{fib}(n)$$

$$\text{fib} : \mathbb{N} \rightarrow \mathbb{N}$$

$$\frac{}{(0, 0) \in \text{Fib}} \quad \frac{}{(1, 1) \in \text{Fib}} \quad \frac{(n+1, a) \in \text{Fib} \quad (n, b) \in \text{Fib}}{(n+2, a+b) \in \text{Fib}}$$

$$R(X) = \{y \mid \frac{x_1 \cdots x_n}{y} \text{ and } x_1, \dots, x_n \in X\}$$

one step derivation

$$\begin{aligned} S^0 &= \emptyset &= \emptyset \\ S^1 &= R(S^0) &= \{(0, 0), (1, 1)\} \\ S^2 &= R(S^1) &= \{(0, 0), (1, 1), (2, 1)\} \\ S^3 &= R(S^2) &= \{(0, 0), (1, 1), (2, 1), (3, 2)\} \\ S^4 &= R(S^3) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3)\} \\ S^5 &= R(S^4) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5)\} \\ S^6 &= R(S^5) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8)\} \\ S^7 &= R(S^6) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8), (7, 13)\} \\ &\vdots & \end{aligned}$$

$$S^0 \subseteq S^1 \subseteq S^2 \subseteq \dots$$

a sequence of partial functions (under-) approximating *fib*

Constructing Inductively Defined Sets – an example

$$\text{fib}(0) = 0$$

$$\text{fib}(1) = 1$$

$$\text{fib}(n + 2) = \text{fib}(n + 1) + \text{fib}(n)$$

$$\text{fib} : \mathbb{N} \rightarrow \mathbb{N}$$

$$\frac{}{(0, 0) \in \text{Fib}} \quad \frac{}{(1, 1) \in \text{Fib}} \quad \frac{(n + 1, a) \in \text{Fib} \quad (n, b) \in \text{Fib}}{(n + 2, a + b) \in \text{Fib}}$$

$$R(X) = \{y \mid \frac{x_1 \cdots x_n}{y} \text{ and } x_1, \dots, x_n \in X\}$$

one step derivation

$$\begin{aligned} S^0 &= \emptyset &= \emptyset \\ S^1 &= R(S^0) &= \{(0, 0), (1, 1)\} \\ S^2 &= R(S^1) &= \{(0, 0), (1, 1), (2, 1)\} \\ S^3 &= R(S^2) &= \{(0, 0), (1, 1), (2, 1), (3, 2)\} \\ S^4 &= R(S^3) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3)\} \\ S^5 &= R(S^4) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5)\} \\ S^6 &= R(S^5) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8)\} \\ S^7 &= R(S^6) &= \{(0, 0), (1, 1), (2, 1), (3, 2), (4, 3), (5, 5), (6, 8), (7, 13)\} \end{aligned}$$

$$S^0 \subseteq S^1 \subseteq S^2 \subseteq \dots$$

⋮

a sequence of partial functions (under-) approximating *fib*

$$S \triangleq \bigcup_{i \in \mathbb{N}} S^i$$

this limit is exactly the (total) function *fib*

Languages

Strings over an alphabet

Let Γ be an alphabet (a finite nonempty set of symbols).

The set $Strings(\Gamma)$ is inductively defined as follows:

- $I = \Gamma \cup \{\varepsilon\}$,
- R_1 : if $x, y \in Strings(\Gamma)$ then $xy \in Strings(\Gamma)$
- xy is the concatenation of the strings x and y ($\varepsilon x = x\varepsilon = x$)
- Notation: $\Gamma^* = Strings(\Gamma)$ (star closure of an alphabet)

An example

$\Gamma = \{a, b\}$, $Strings(\Gamma) = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$

Languages

- A language on Γ is any subset $L \subseteq \Gamma^*$
- They can be defined inductively through formal grammars

Languages

Strings over an alphabet

Let Γ be an alphabet (a finite nonempty set of symbols).

The set $Strings(\Gamma)$ is inductively defined as follows:

- $I = \Gamma \cup \{\varepsilon\}$,
- R_1 : if $x, y \in Strings(\Gamma)$ then $xy \in Strings(\Gamma)$
- xy is the concatenation of the strings x and y ($\varepsilon x = x\varepsilon = x$)
- Notation: $\Gamma^* = Strings(\Gamma)$ (star closure of an alphabet)

An example

$\Gamma = \{a, b\}$, $Strings(\Gamma) = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$

Languages

- A language on Γ is any subset $L \subseteq \Gamma^*$
- They can be defined inductively through formal grammars

Languages

Strings over an alphabet

Let Γ be an alphabet (a finite nonempty set of symbols).

The set $Strings(\Gamma)$ is inductively defined as follows:

- $I = \Gamma \cup \{\varepsilon\}$,
- R_1 : if $x, y \in Strings(\Gamma)$ then $xy \in Strings(\Gamma)$
- xy is the concatenation of the strings x and y ($\varepsilon x = x\varepsilon = x$)
- Notation: $\Gamma^* = Strings(\Gamma)$ (star closure of an alphabet)

An example

$\Gamma = \{a, b\}$, $Strings(\Gamma) = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$

Languages

- A **language** on Γ is any subset $L \subseteq \Gamma^*$
- They can be defined inductively through **formal grammars**

Grammars

A **grammar** is a 4-tuple $G = \langle T, NT, S, P \rangle$ where

- ① **terminals** T
- ② **nonterminals** NT $(T \cap NT = \emptyset)$
- ③ **start symbol** $S \in NT$
- ④ **productions** $P \subseteq (T \cup NT)^* \times (T \cup NT)^*$
if $(u, v) \in P$ then u has at least a nonterminal symbol

Grammars

A **grammar** is a 4-tuple $G = \langle T, NT, S, P \rangle$ where

- ① **terminals** T
- ② **nonterminals** NT $(T \cap NT = \emptyset)$
- ③ **start symbol** $S \in NT$
- ④ **productions** $P \subseteq (T \cup NT)^* \times (T \cup NT)^*$
if $(u, v) \in P$ then u has at least a nonterminal symbol

(u, v) is also written as $u \rightarrow v$

Grammars

A **grammar** is a 4-tuple $G = \langle T, NT, S, P \rangle$ where

- ① **terminals** T
- ② **nonterminals** NT $(T \cap NT = \emptyset)$
- ③ **start symbol** $S \in NT$
- ④ **productions** $P \subseteq (T \cup NT)^* \times (T \cup NT)^*$
if $(u, v) \in P$ then u has at least a nonterminal symbol

(u, v) is also written as $u \rightarrow v$

$(u, v_1), (u, v_2), \dots, (u, v_n) \in P$ also written as

$$u \rightarrow v_1 \mid v_2 \mid \dots \mid v_n$$

or

$$u ::= v_1 \mid v_2 \mid \dots \mid v_n$$

Backus-Naur Normal Form (BNF)

Grammars – derivation relation

$$G = \langle T, N, S, P \rangle$$

$$\frac{s = lur \quad t = lvr \quad u \rightarrow v}{s \Rightarrow t} \quad \text{for any production } u \rightarrow v \text{ in } P$$

\Rightarrow^* is the reflexive and transitive closure of \Rightarrow

Grammars and Languages

The language generated by G is the following set of string of terminal symbols

$$L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

S

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

$\underline{S} \Rightarrow aB\underline{S}c$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

$\underline{S} \Rightarrow aB\underline{S}c \Rightarrow aBaB\underline{S}cc$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

$\underline{S} \Rightarrow aB\underline{S}c \Rightarrow aBaB\underline{S}cc \Rightarrow a\underline{Ba}Babccc$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

$\underline{S} \Rightarrow aB\underline{S}c \Rightarrow aBaB\underline{S}cc \Rightarrow a\underline{Ba}Babccc \Rightarrow$
 $\Rightarrow aaB\underline{B}abccc$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

$\underline{S} \Rightarrow aB\underline{S}c \Rightarrow aBaB\underline{S}cc \Rightarrow a\underline{Ba}Babccc \Rightarrow$
 $\Rightarrow aaB\underline{B}abccc \Rightarrow aa\underline{Ba}Bbccc$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

$\underline{S} \Rightarrow aB\underline{S}c \Rightarrow aBaB\underline{S}cc \Rightarrow a\underline{B}aBabccc \Rightarrow$
 $\Rightarrow aaB\underline{B}abccc \Rightarrow aa\underline{B}aBbccc \Rightarrow aaaB\underline{B}bccc$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

$\underline{S} \Rightarrow aB\underline{S}c \Rightarrow aBaB\underline{S}cc \Rightarrow a\underline{B}aBabccc \Rightarrow$
 $\Rightarrow aaB\underline{B}abccc \Rightarrow aa\underline{B}aBbccc \Rightarrow aaaB\underline{B}bccc \Rightarrow$
 $\Rightarrow aaa\underline{B}bbccc$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

$\underline{S} \Rightarrow aB\underline{S}c \Rightarrow aBaB\underline{S}cc \Rightarrow a\underline{B}aBabccc \Rightarrow$
 $\Rightarrow aaB\underline{B}abccc \Rightarrow aa\underline{B}aBbccc \Rightarrow aaaB\underline{B}bccc \Rightarrow$
 $\Rightarrow aaa\underline{B}bbccc \Rightarrow aaabbbccc \in \{a, b, c\}^*$

Grammars – example

$T = \{a, b, c\}$ $NT = \{S, B\}$ start symbol: S

$S \rightarrow aBSc \mid abc$ $Ba \rightarrow aB$ $Bb \rightarrow bb$

A derivation:

$\underline{S} \Rightarrow aB\underline{S}c \Rightarrow aBaB\underline{S}cc \Rightarrow a\underline{Ba}Babccc \Rightarrow$
 $\Rightarrow aaB\underline{B}abccc \Rightarrow aa\underline{Ba}Bbccc \Rightarrow aaaB\underline{B}bccc \Rightarrow$
 $\Rightarrow aaa\underline{B}bbccc \Rightarrow aaabbbccc \in \{a, b, c\}^*$

$L(G) = \{a^n b^n c^n \mid n \geq 1\}$

Abstract and Concrete Syntax

When providing the syntax of programming languages we need to worry about precedence of operators or grouping of statements to distinguish, e.g., between:

$$(3 + 4) * 5 \quad \text{and} \quad 3 + (4 * 5),$$

while p **do** $(c_1; c_2)$ and **(while** p **do** c_1); c_2

Thus, e.g., for arithmetic expressions we have grammars with parenthesis:

$$E ::= n \mid (E) \mid E + E \mid E - E \mid E * E \mid E / E$$

or more elaborate grammars specifying the precedence of operators (like the next one ...)

Abstract and Concrete Syntax

$E ::= E + T \mid E - T \mid T$	(expressions)
$T ::= T * P \mid T / P \mid P$	(terms)
$P ::= N \mid (E)$	(atomic expressions)
$N ::= DN \mid D$	(numbers)
$D ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$	(digits)

- When defining the semantics of programming languages, we are only concerned with the meaning of their constructs, not with the theory of how to write programs
- We thus resort to **abstract syntax** that leaves us the task of adding enough parentheses to programs to ensure they can be built-up in a unique way

Abstract syntax specifies the **parse trees** of a language; it is the job of concrete syntax to provide enough information through parentheses or precedence rules for a string to parse uniquely

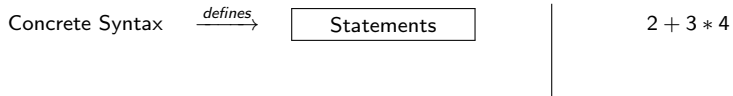
Abstract and Concrete Syntax

$E ::= E + T \mid E - T \mid T$	(expressions)
$T ::= T * P \mid T / P \mid P$	(terms)
$P ::= N \mid (E)$	(atomic expressions)
$N ::= DN \mid D$	(numbers)
$D ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$	(digits)

- When defining the semantics of programming languages, we are only concerned with the meaning of their constructs, not with the theory of how to write programs
- We thus resort to **abstract syntax** that leaves us the task of adding enough parentheses to programs to ensure they can be built-up in a unique way

Abstract syntax specifies the **parse trees** of a language; it is the job of concrete syntax to provide enough information through parentheses or precedence rules for a string to parse uniquely

From Parsing to Execution



From Parsing to Execution

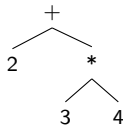
Concrete Syntax $\xrightarrow{\text{defines}}$ Statements

↓ Parse

Abstract Syntax $\xrightarrow{\text{defines}}$ Syntax Trees

2 + 3 * 4

↓



From Parsing to Execution

Concrete Syntax $\xrightarrow{\text{defines}}$ Statements

↓ Parse

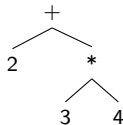
Abstract Syntax $\xrightarrow{\text{defines}}$ Syntax Trees

↓ Execute

Semantics $\xrightarrow{\text{defines}}$ Meaning of
Syntax Trees

2 + 3 * 4

↓



↓

14

Labelled Transition Systems

A labelled transition system is a 4-tuple $S = \langle Q, A, \rightarrow, q_0 \rangle$ such that

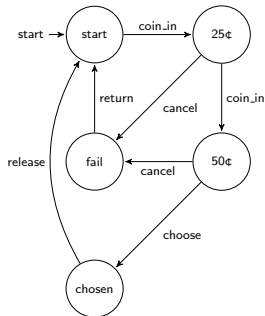
- ① **states** Q
- ② **actions** A
- ③ **transitions** $\rightarrow \subseteq Q \times A \times Q$
 $q \xrightarrow{a} q'$ denotes $(q, a, q') \in \rightarrow$
- ④ **initial state** $q_0 \in Q$

Labelled Transition Systems

A labelled transition system is a 4-tuple $S = \langle Q, A, \rightarrow, q_0 \rangle$ such that

- 1 **states** Q
- 2 **actions** A
- 3 **transitions** $\rightarrow \subseteq Q \times A \times Q$
 $q \xrightarrow{a} q'$ denotes $(q, a, q') \in \rightarrow$
- 4 **initial state** $q_0 \in Q$

Vending machine:



Labelled Transition Systems

A labelled transition system is a 4-tuple $S = \langle Q, A, \rightarrow, q_0 \rangle$ such that

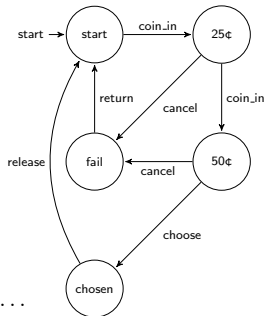
- 1 **states** Q
- 2 **actions** A
- 3 **transitions** $\rightarrow \subseteq Q \times A \times Q$
 $q \xrightarrow{a} q'$ denotes $(q, a, q') \in \rightarrow$
- 4 **initial state** $q_0 \in Q$

Semantics: traces

$\tau : a_0 a_1 a_2 a_3 a_4 a_5 a_6 \dots$

$\tau : \text{coin_in cancel return coin_in coin_in choose release } \dots$

Vending machine:



LTS-based Semantics of Arithmetic Expressions

$$\frac{m \circ n = k}{m \circ n \xrightarrow{\circ} k} \quad (\text{op}) \quad \frac{E_1 \xrightarrow{\circ'} E'_1}{E_1 \circ E_2 \xrightarrow{\circ'} E'_1 \circ E_2} \quad (\text{rl}) \quad \frac{E_2 \xrightarrow{\circ'} E'_2}{E_1 \circ E_2 \xrightarrow{\circ'} E_1 \circ E'_2} \quad (\text{rr})$$

LTS-based Semantics of Arithmetic Expressions

$$\frac{m \circ n = k}{m \circ n \xrightarrow{\circ} k} \quad (\text{op}) \qquad \frac{E_1 \xrightarrow{\circ'} E'_1}{E_1 \circ E_2 \xrightarrow{\circ'} E'_1 \circ E_2} \quad (\text{rl}) \qquad \frac{E_2 \xrightarrow{\circ'} E'_2}{E_1 \circ E_2 \xrightarrow{\circ'} E_1 \circ E'_2} \quad (\text{rr})$$

$$(4 + (7 * 3)) / (6 - 1) \xrightarrow{*} (4 + 21) / (6 - 1) \xrightarrow{+} 25 / (6 - 1) \xrightarrow{-} 25 / 5 \xrightarrow{/} 5$$

LTS-based Semantics of Arithmetic Expressions

$$\frac{m \circ n = k}{m \circ n \xrightarrow{\circ} k} \quad (\text{op}) \qquad \frac{E_1 \xrightarrow{\circ'} E'_1}{E_1 \circ E_2 \xrightarrow{\circ'} E'_1 \circ E_2} \quad (\text{rl}) \qquad \frac{E_2 \xrightarrow{\circ'} E'_2}{E_1 \circ E_2 \xrightarrow{\circ'} E_1 \circ E'_2} \quad (\text{rr})$$

$$(4 + (7 * 3)) / (6 - 1) \xrightarrow{*} (4 + 21) / (6 - 1) \xrightarrow{+} 25 / (6 - 1) \xrightarrow{-} 25 / 5 \xrightarrow{/} 5$$

$$\frac{\frac{7 * 3 = 21}{7 * 3 \xrightarrow{*} 21}}{4 + (7 * 3) \xrightarrow{*} 4 + 21}}{(4 + (7 * 3)) / (6 - 1) \xrightarrow{*} (4 + 21) / (6 - 1)}$$

$$\frac{4 + 21 = 25}{4 + 21 \xrightarrow{+} 25}}{(4 + 21) / (6 - 1) \xrightarrow{+} 25 / (6 - 1)}$$

similarly for $-$ and $/$

Finite State Automata as language recognizers

A *finite state automaton* M is a 5-tuple $M = \langle Q, \Gamma, \rightarrow, q_0, F \rangle$ s.t.

- ① **states** Q **finite !**
- ② **alphabet** Γ
- ③ **transitions** $\rightarrow \subseteq Q \times \Gamma \times Q$
 $q \xrightarrow{a} q'$ denotes $(q, a, q') \in \rightarrow$
- ④ **initial state** $q_0 \in Q$
- ⑤ **accepting states** $F \subseteq Q$

Finite State Automata as language recognizers

A *finite state automaton* M is a 5-tuple $M = \langle Q, \Gamma, \rightarrow, q_0, F \rangle$ s.t.

- 1 **states** Q **finite !**
- 2 **alphabet** Γ
- 3 **transitions** $\rightarrow \subseteq Q \times \Gamma \times Q$
 $q \xrightarrow{a} q'$ denotes $(q, a, q') \in \rightarrow$
- 4 **initial state** $q_0 \in Q$
- 5 **accepting states** $F \subseteq Q$

$$p \xRightarrow{w} q \quad \text{iff} \quad p \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} p_n = q \quad w = a_1 \cdots a_n$$

Finite State Automata as language recognizers

A *finite state automaton* M is a 5-tuple $M = \langle Q, \Gamma, \rightarrow, q_0, F \rangle$ s.t.

- 1 **states** Q **finite !**
- 2 **alphabet** Γ
- 3 **transitions** $\rightarrow \subseteq Q \times \Gamma \times Q$
 $q \xrightarrow{a} q'$ denotes $(q, a, q') \in \rightarrow$
- 4 **initial state** $q_0 \in Q$
- 5 **accepting states** $F \subseteq Q$

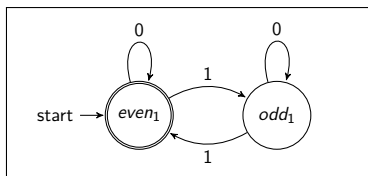
$$p \xRightarrow{w} q \quad \text{iff} \quad p \xrightarrow{a_1} p_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} p_n = q \quad w = a_1 \cdots a_n$$

Semantics of Finite State Automata

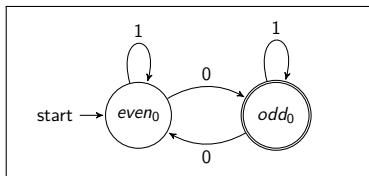
The language accepted by a Finite State Automata is the set:

$$L(M) = \{w \in \Gamma^* \mid q_0 \xRightarrow{w} q \text{ and } q \in F\}$$

Some Regular Bit-Strings – $\Gamma = \{0, 1\}$

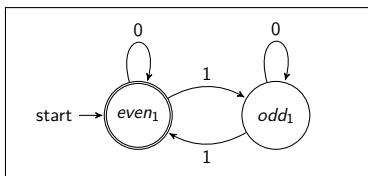


$$L(A_1) = \{w \mid \text{even number of 1's}\}$$

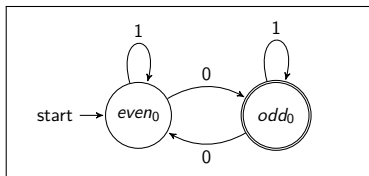


$$L(A_2) = \{w \mid \text{odd number of 0's}\}$$

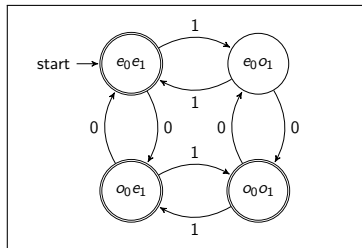
Some Regular Bit-Strings – $\Gamma = \{0, 1\}$



$L(A_1) = \{w \mid \text{even number of 1's}\}$

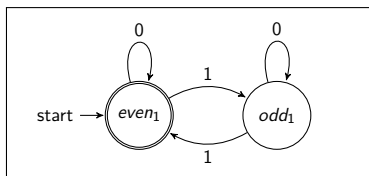


$L(A_2) = \{w \mid \text{odd number of 0's}\}$

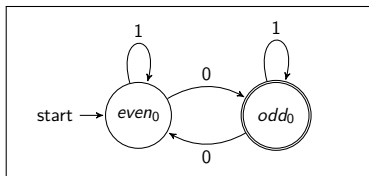


$L(A_1) \cup L(A_2)$

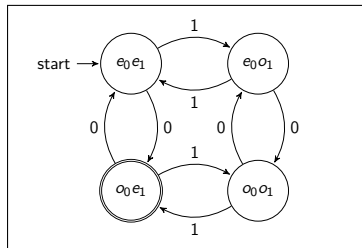
Some Regular Bit-Strings – $\Gamma = \{0, 1\}$



$L(A_1) = \{w \mid \text{even number of 1's}\}$

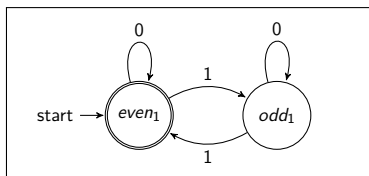


$L(A_2) = \{w \mid \text{odd number of 0's}\}$

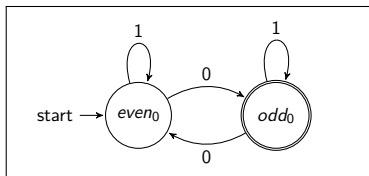


$L(A_1) \cap L(A_2)$

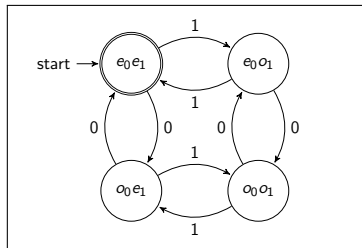
Some Regular Bit-Strings – $\Gamma = \{0, 1\}$



$L(A_1) = \{w \mid \text{even number of 1's}\}$

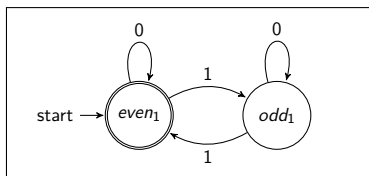


$L(A_2) = \{w \mid \text{odd number of 0's}\}$

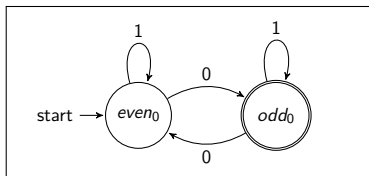


$L(A_1) \setminus L(A_2)$

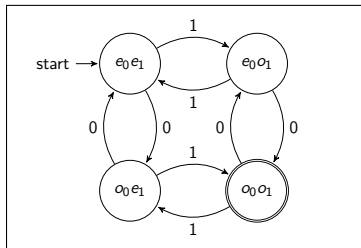
Some Regular Bit-Strings – $\Gamma = \{0, 1\}$



$L(A_1) = \{w \mid \text{even number of 1's}\}$

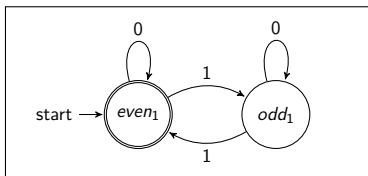


$L(A_2) = \{w \mid \text{odd number of 0's}\}$

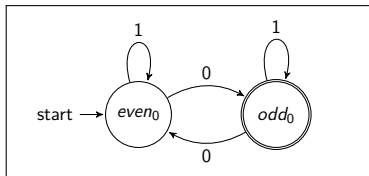


$L(A_2) \setminus L(A_1)$

Some Regular Bit-Strings – $\Gamma = \{0, 1\}$



$L(A_1) = \{w \mid \text{even number of 1's}\}$



$L(A_2) = \{w \mid \text{odd number of 0's}\}$

regular languages are closed
w.r.t. the operations of \cap , \cup ,
 \setminus , complement, reversal,
concatenation, star closure,
...

Regular Languages

Chomsky Hierarchy	Grammar Restriction	Language	Abstract Machine
Type 0	unrestricted	recursively enumerable	Turing machines
Type 1	$\alpha A \beta \rightarrow \alpha \gamma \beta$	context sensitive	linear bounded automata
Type 2	$A \rightarrow \gamma$	context free	nondeterministic pushdown automata
Type 3	$A \rightarrow a \quad A \rightarrow aB$	regular	finite state automata

with $A, B \in NT$, and $a \in T$ and $\alpha, \beta, \gamma \in (T \cup NT)^*$