Finite Automata Theory and Formal Languages

TMV027/DIT321– LP4 2016

Lecture 2
Ana Bove

March 22nd 2016

Overview of today's lecture:

- Recap on logic;
- Recap on sets, relations and functions;
- Central concepts of automata theory.

Propositional Logic

Definition: A *proposition* is an statement which is either *true* (T) or *false* (F).

Example: My name is Ana.

I come from Uruguay.

I have 3 children.

I can speak 4 different languages.

It is not always known what the *truth value* of a proposition is.

Goldbach's conjecture: Every even integer greater than 2 can be expressed as the sum of two primes.

Connective and Truth Tables

We can combine propositions by using *connectives*:

- $\neg$: negation, not
- $\wedge$: conjunction, and
- $\vee$: disjunction, or
- $\Rightarrow$: conditional, if-then, $\rightarrow$
- $\Leftrightarrow$: equivalence, if-and-only-if, $\leftrightarrow$

These are their *truth tables* (observe the conditional...):

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

Conditionals

Example: Consider the statement *if it rains then I take my umbrella*.

If it doesn't rain, should I take the umbrella?

Recall truth table for conditional:

it rains	I take my umbrella	it rains $\Rightarrow$ I take my umbrella
T	T	T
T	F	F
F	T	T
F	F	T

I could do what I want then!

The condition only says what *MUST* happen when it *DOES* rain!

Combined Propositions

Example: Express *either you study and pass the exam, or your don't pass the exam* with propositions and construct its truth table.

Let p be "you study".
Let q be "you pass the exam".

Then the sentence is expressed by $(p \wedge q) \vee \neg q$.

p	q	$p \wedge q$	$\neg q$	$(p \wedge q) \vee \neg q$
T	T	T	F	T
T	F	F	T	T
F	T	F	F	F
F	F	F	T	T

Tautologies and Logical Equivalence

Definition: A proposition that is always true is called a *tautology*.

Example: The *law of the excluded middle* is a tautology in classical logic

p	$\neg p$	$p \vee \neg p$
T	F	T
F	T	T

Definition: Two propositions are *logically equivalent* ($\equiv$) if they have the same truth table.

Example: $p \Rightarrow q \equiv \neg p \vee q$:

p	q	$p \Rightarrow q$	$\neg p$	$\neg p \vee q$
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Laws of (Classical) Logic

Equivalence:	$p \Leftrightarrow q \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$	
Implication:	$p \Rightarrow q \equiv \neg p \vee q$	
Double negation:	$\neg\neg p \equiv p$	
Idempotent:	$p \wedge p \equiv p$	$p \vee p \equiv p$
Commutative:	$p \wedge q \equiv q \wedge p$	$p \vee q \equiv q \vee p$
Associative:	$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	
	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	
Distributive:	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	
	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	
de Morgan:	$\neg(p \wedge q) \equiv \neg p \vee \neg q$	$\neg(p \vee q) \equiv \neg p \wedge \neg q$
Identity:	$p \wedge T \equiv p$	$p \vee F \equiv p$
Annihilation:	$p \wedge F \equiv F$	$p \vee T \equiv T$
Inverse:	$p \wedge \neg p \equiv F$	$p \vee \neg p \equiv T$
Absorption:	$p \wedge (p \vee q) \equiv p$	$p \vee (p \wedge q) \equiv p$

Exercise: Construct the truth tables and check the logical equivalences!

Statements with Variables

By using variables we could talk about any element in a domain.

Example: Consider the following property for $x \in \mathbb{N}$ (Natural numbers):

$$x > 4 \Rightarrow x > 2$$

When statements have variables we are actually working on *predicate logic*.

Reasoning in predicate logic is more complicated since variables can range over an infinite set of values.

Predicate Logic

Definition: A *predicate* is a statement with one or more variables.

When we assign values to all variable in a predicate we get a proposition.

Definition: The expressions *for all* ($\forall$) and *exists* ($\exists$) are called *quantifiers*.

Example: Express the following 2 statements in predicate logic:

- For every number x there is a number y such that x is equal to y
  $\forall x.\exists y.x = y$
- There is a number x such that for every number y then x is equal to y
  $\exists x.\forall y.x = y$

Are they the same statement?

More Laws of (Classical) Logic

We have that

$$\neg\forall x.P(x) \equiv \exists x.\neg P(x)$$

and

$$\neg\exists x.P(x) \equiv \forall x.\neg P(x)$$

Sets

Definition: A *set* is a collection of well defined and distinct objects or elements.

A set might be finite or infinite.

Sets can be described/defined in different ways:

Enumeration: mainly finite sets, sometimes with help of ...

WeekDays = {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday}

OddNat $= \{1, 3, 5, 7, \ldots\}$

Characteristic Property: OddNat $= \{x \in \mathbb{N} \mid x$ is odd$\}$.

Operations on Other Sets: $A \cup B$, $A \cap B$, ... (see slide 12)

Inductive Definitions: More on this later ...

$\vdots$

Membership on Sets

Definition: We denote that x is an *element* of set A by $x \in A$.

It is important to determine whether $x \in A$ or $x \notin A$.
However this is not always possible.

Example: Let P be the set of programs that always terminate.

Can we always be sure if a certain program $pgr \in P$?

Russell's paradox: Let $R = \{x \mid x \notin x\}$.

Then $R \in R \Leftrightarrow R \notin R$!

Some Operations and Properties on Sets

Union: $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$.

Intersection: $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$.

Cartesian Product: $A \times B = \{(x, y) \mid x \in A \text{ and } y \in B\}$.
Observe this is a collection of ordered pairs! $(x, y) \neq (y, x)$.

Difference: $S - A = \{x \mid x \in S \text{ and } x \notin A\}$.
When the set S is known, $S - A$ is written $\overline{A}$ and is called the complement.
$S - A$ is sometimes denoted $S \backslash A$ and $\overline{A}$ is sometimes denoted A'.

Subset: $A \subseteq B$ if for all $x \in A$ then $x \in B$.

Equality: $A = B$ if $A \subseteq B$ and $B \subseteq A$.

Proper Subset: $A \subset B$ if $A \subseteq B$ and $A \neq B$.

Some Particular Sets

Empty set: $\emptyset$ is the set with no elements.
We have $\emptyset \subseteq S$ for any set S.

Singleton sets: Sets with only one element: $\{p_0\}$, $\{p_1\}$.

Finite sets: Set with a finite number n of elements:
$\{p_1, \ldots, p_n\} = \{p_1\} \cup \ldots \cup \{p_n\}$.

Power sets: $\mathcal{P}ow(S)$ the set of all subsets of the set S.
$\mathcal{P}ow(S) = \{A \mid A \subseteq S\}$.
Observe that $\emptyset \in \mathcal{P}ow(S)$ and $S \in \mathcal{P}ow(S)$.
Also, if $|S| = n$ then $|\mathcal{P}ow(S)| = 2^n$.

Note: $\emptyset \neq \{\emptyset\}$!!

Algebraic Laws for Sets

Idempotent:	$A \cup A = A$		$A \cap A = A$
Commutative:	$A \cup B = B \cup A$		$A \cap B = B \cap A$
Associative:	$(A \cup B) \cup C = A \cup (B \cup C)$		
	$(A \cap B) \cap C = A \cap (B \cap C)$		
Distributive:	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$		
	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$		
de Morgan:	$\overline{(A \cup B)} = \overline{A} \cap \overline{B}$		$\overline{(A \cap B)} = \overline{A} \cup \overline{B}$
Laws for $\emptyset$:	$A \cup \emptyset = A$		$A \cap \emptyset = \emptyset$
Laws for Universe:	$A \cup U = U$		$A \cap U = A$
Complements:	$\overline{\overline{A}} = A$	$A \cup \overline{A} = U$	$A \cap \overline{A} = \emptyset$
	$\overline{U} = \emptyset$	$\overline{\emptyset} = U$	
Absorption:	$A \cup (A \cap B) = A$		$A \cap (A \cup B) = A$

Exercise: Prove the equality of the sets by showing the double inclusion!

Relations

Definition: A (binary) *relation* R between two sets A and B is a subset of $A \times B$, that is, $R \subseteq A \times B$.

Notation: $(a, b) \in R$, $a\,R\,b$, $R(a, b)$, (a, b) satisfies R.

Definition: A relation R over a set S, that is $R \subseteq S \times S$, is

Reflexive if $\forall a \in S.\ a\,R\,a$;

Symmetric if $\forall a, b \in S.\ a\,R\,b \Rightarrow b\,R\,a$;

Transitive if $\forall a, b, c \in S.\ a\,R\,b \wedge b\,R\,c \Rightarrow a\,R\,c$.

Definition: If S has an equality relation $= \subseteq S \times S$ and $R \subseteq S \times S$ then R is antisymmetric if $\forall a, b \in S.\ a\,R\,b \wedge b\,R\,a \Rightarrow a = b$.

Example of Relations

Let $S = \{1, 2, 3\}$ and let $= \subseteq S \times S$ be as expected.
Which of these relations are reflexive, symmetric, antisymmetric, and/or transitive?

Play at `kahoot.it`!

- $R_1 = \emptyset$ — *Symmetric, Antisymmetric, Transitive*
- $R_2 = \{(1, 2)\}$ — *Antisymmetric, Transitive*
- $R_3 = \{(1, 2), (2, 3)\}$ — *Antisymmetric*
- $R_4 = \{(1, 2), (2, 3), (1, 3)\}$ — *Antisymmetric, Transitive*
- $R_5 = \{(1, 2), (2, 1)\}$ — *Symmetric*
- $R_6 = \{(1, 2), (2, 1), (1, 1)\}$ — *Symmetric*
- $R_7 = \{(1, 2), (2, 1), (1, 1), (2, 2)\}$ — *Symmetric, Transitive*
- $R_8 = \{(1, 2), (2, 1), (1, 1), (2, 2), (3, 3)\}$ — *Reflexive, Symm, Trans*

Equivalent Relations and Partial Orders

Definition: A relation R over a set S that is reflexive, symmetric and transitive is called an *equivalence relation* over S.

Example: $=$ is an equivalence over $\mathbb{N}$.

Definition: A relation R over a set S that is reflexive, antisymmetric and transitive is called a *partial order* over S.

Example: $\leqslant$ is a partial order over $\mathbb{N}$ but $<$ is not!.

Definition: A relation R over a set S is called a *total order* over S if:

- R is a partial order;
- $\forall a, b \in S.\ a\, R\, b \vee b\, R\, a$.

Example: $\leqslant$ is a total order over $\mathbb{N}$.

Partitions

Definition: A set P is a *partition* over the set S if:

- Every element of P is a non-empty subset of S

$$\forall C \in P.\ C \neq \emptyset \wedge C \subseteq S;$$

- Elements of P are pairwise disjoint

$$\forall C_1, C_2 \in P.\ C_1 \neq C_2 \Rightarrow C_1 \cap C_2 = \emptyset;$$

- The union of the elements of P is equal to S

$$\bigcup_{C \in P} C = S.$$

Equivalent Classes

Let R be an equivalent relation over S.

Definition: If $a \in S$, then the *equivalent class* of a in S is the set defined as $[a] = \{b \in S \mid a\,R\,b\}$.

Lemma: $\forall a, b \in S,\ [a] = [b]$ *iff* $a\,R\,b$.

Theorem: *The set of all equivalence classes in S w.r.t. R form the* *quotient* *partition over S.*

Notation: This partition is denoted as S/R.

Example: The rational numbers $\mathbb{Q}$ can be formally defined as the equivalence classes of the quotient set $\mathbb{Z} \times \mathbb{Z}^+/\sim$, where $\sim$ is the equivalence relation defined by $(m_1, n_1) \sim (m_2, n_2)$ iff $m_1 n_2 =_{\mathbb{Z}} m_2 n_1$.

Functions

Definition: A *function* f from A to B is a relation $f \subseteq A \times B$ such that, given $x \in A$ and $y, z \in B$, if $x\,f\,y$ and $x\,f\,z$ then $y = z$.

Notation: If f is a function from A to B we write $f : A \to B$.

Notation: That $x\,f\,y$ is usually written as $f(x) = y$.

Example: $\mathrm{sq} : \mathbb{Z} \to \mathbb{N}$ such that $\mathrm{sq}(n) = n^2$.

Observe that $\mathrm{sq}(2) = 4$ and $\mathrm{sq}(-2) = 4$.

Domain, Codomain, Range and Image

Let $f : A \to B$.

Definition: The sets A and B are called the *domain* and the *codomain* of the function, respectively.

Definition: The set $\mathrm{Dom}(f)$ or Dom_f for which the *function is defined* is given by $\{x \in A \mid \exists y \in B. f(x) = y\} \subseteq A$.

We will also refer to $\mathrm{Dom}(f)$ as the domain of f.

Definition: The set $\{y \in B \mid \exists x \in A. f(x) = y\} \subseteq B$ is called the *range* or *image* of f and denoted $\mathrm{Im}(f)$ or Im_f.

Example: The image of sq is NOT all $\mathbb{N}$ but $\{0, 1, 4, 9, 16, 25, 36, \ldots\}$.

Total and Partial Functions

Let $f : A \to B$.

Definition: If $\text{Dom}(f) = A$ then f is called a *total* function.

Example: sq is a total function.

Definition: If $\text{Dom}(f) \subset A$ then f is called a *partial* function.

Example: $\text{sqr} : \mathbb{N} \to \mathbb{N}$ such that $\text{sqr}(n) = \sqrt{n}$ is a partial function.

Note: In some cases it is not known if a function is partial or total.

Example: It is not known if $\text{collatz} : \mathbb{N} \to \mathbb{N}$ is total or not.

$$\begin{array}{l} \text{collatz}(0) = 1 \\ \text{collatz}(1) = 1 \end{array} \qquad \text{collatz}(n) = \begin{cases} \text{collatz}(n/2) & \text{if } n \text{ even} \\ \text{collatz}(3n+1) & \text{if } n \text{ odd} \end{cases}$$

Injective or One-to-one Functions

Let $f : A \to B$.

Definition: f is called an *injective* or *one-to-one* function if $\forall x, y \in A. f(x) = f(y) \Rightarrow x = y$.

Alternatively:

Definition: f is called an *injective* or *one-to-one* function if $\forall x, y \in A. x \neq y \Rightarrow f(x) \neq f(y)$.

Exercise: Prove that $\text{double} : \mathbb{N} \to \mathbb{N}$ such that $\text{double}(n) = 2n$ is injective.

The Pigeonhole Principle

"If you have more pigeons than pigeonholes and each pigeon flies into some pigeonhole, then there must be at least one hole with more than one pigeon."

More formally: if $f : A \to B$ and $|\text{Dom}_f(A)| > |B|$ then f cannot be *injective*.
That is, there must exist $x, y \in A$ such that $x \neq y$ and $f(x) = f(y)$.

This principle is often used to show the existence of an object without building this object explicitly.

Example: In a room with at least 13 people, at least 2 of them are born the same month.

Surjective or Onto Functions

Let $f : A \to B$.

Definition: f is called an *surjective* or *onto* function if
$\forall y \in B.\exists x \in A.f(x) = y$.

Note: If f is surjective then $\text{Im}(f) = B$.

Exercise: Prove that $\text{f} : \mathbb{R} \to \mathbb{R}$ such that $\text{f}(n) = 2n + 1$ is surjective.

Bijective and Inverse Functions

Definition: A function that is both injective and surjective is called a *bijective* function.

Definition: If $f : A \to B$ is a bijective function, then there exists an *inverse* function $f^{-1} : B \to A$ such that $\forall x \in A.f^{-1}(f(x)) = x$ and $\forall y \in B.f(f^{-1}(y)) = y$.

Exercise: Is $\mathrm{g} : \mathbb{Z} \to \mathbb{Z}$ such that $\mathrm{g}(n) = 2n + 1$ bijective?

Exercise: Which is the inverse of $\mathrm{f} : \mathbb{R} \to \mathbb{R}$ such that $\mathrm{f}(n) = 2n + 1$?

Lemma: *If* $f : A \to B$ is a bijective function, then $f^{-1} : B \to \mathrm{Dom}_f(A)$ is also bijective.

Composition and Restriction

Definition: Let $f : A \to B$ and $g : B \to C$. The *composition* $g \circ f : A \to C$ is defined as $g \circ f(x) = g(f(x))$.

Note: We need that $\mathrm{Im}(f) \subseteq \mathrm{Dom}(g)$ for the composition to be defined.

Example: If $\mathrm{f} : \mathbb{Z} \to \mathbb{Z}$ is such that $\mathrm{f}(n) = 3n - 2$ and $\mathrm{g} : \mathbb{R} \to \mathbb{R}$ is such that $\mathrm{g}(m) = m/2$, then $\mathrm{g} \circ \mathrm{f} : \mathbb{Z} \to \mathbb{R}$ is $\mathrm{g} \circ \mathrm{f}(x) = (3x - 2)/2$.

Definition: Let $f : A \to B$ and $S \subset A$. The *restriction* of f to S is the function $f_{|S} : S \to B$ such that $f_{|S}(x) = f(x), \forall x \in S$.

Central Concepts of Automata Theory: Alphabets

Definition: An *alphabet* is a finite, non-empty set of symbols, usually denoted by Σ.

The number of symbols in Σ is denoted as $|\Sigma|$.

Notation: We will use $a, b, c, \ldots$ to denote symbols.

Note: Alphabets will represent the observable events of the automata.

Example: Some alphabets:

- on/off-switch: $\Sigma = \{\text{Push}\}$;
- simple vending machine: $\Sigma = \{5\ kr, \text{choc}\}$;
- complex vending machine: $\Sigma = \{5\ kr, 10\ kr, \text{choc}, \text{big choc}\}$;
- parity counter: $\Sigma = \{p_0, p_1\}$.

Strings or Words

Definition: *Strings/Words* are finite sequence of symbols from some alphabet.

Notation: We will use $w, x, y, z, \ldots$ to denote words.

Note: Words will represent the *behaviour* of an automaton.

Example: Some behaviours:

- on/off-switch: Push Push Push Push;
- simple vending machine: 5 kr choc 5 kr choc 5 kr choc;
- parity counter: $p_0 p_1$ or $p_0 p_0 p_0 p_1 p_1 p_0$.

Note: Some words do NOT represent *behaviour* though ...

Example: simple vending machine: choc choc choc.

Inductive Definition of Σ^*

Definition: Σ^* is the set of *all words* for a given alphabet Σ.

This can be described inductively in at least 2 different ways:

1. Base case: $\epsilon \in \Sigma^*$;
   Inductive step: if $a \in \Sigma$ and $x \in \Sigma^*$ then $ax \in \Sigma^*$.
   (We will usually work with this definition.)

2. Base case: $\epsilon \in \Sigma^*$;
   Inductive step: if $a \in \Sigma$ and $x \in \Sigma^*$ then $xa \in \Sigma^*$.

We can (recursively) *define* functions over Σ^* and (inductively) *prove* properties about those functions.

(More on induction next lecture.)

Concatenation

Definition: Given the strings x and y, the *concatenation* xy is defined as:

$$\begin{aligned} \epsilon y &= y \\ (ax')y &= a(x'y) \end{aligned}$$

Example: Observe that in general $xy \neq yx$.

If $x = 010$ and $y = 11$ then $xy = 01011$ and $yx = 11010$.

Lemma: *If Σ has more than one symbol then concatenation is not commutative.*

Prefix and Suffix

Definition: Given x and y words over a certain alphabet Σ:

- x is a *prefix* of y iff there exists z such that $y = xz$;
- x is a *suffix* of y iff there exists z such that $y = zx$.

Note: $\forall\ x.\ \epsilon$ is both a prefix and a suffix of x.

Note: $\forall\ x.\ x$ is both a prefix and a suffix of x.

Length and Reverse

Definition: The *length* function $|\_| : \Sigma^* \to \mathbb{N}$ is defined as:

$$\begin{aligned} |\epsilon| &= 0 \\ |ax| &= 1 + |x| \end{aligned}$$

Example: $|01010| = 5$.

Definition: The *reverse* function $\mathrm{rev}(\_) : \Sigma^* \to \Sigma^*$ as:

$$\begin{aligned} \mathrm{rev}(\epsilon) &= \epsilon \\ \mathrm{rev}(ax) &= \mathrm{rev}(x)a \end{aligned}$$

Intuitively, $\mathrm{rev}(a_1 \ldots a_n) = a_n \ldots a_1$.

Power

Of a string: We define x^n as follows:

$$x^0 = \epsilon$$
$$x^{n+1} = xx^n$$

Example: $(010)^3 = 010010010$.

Of an alphabet: We define Σ^n, the set of words over Σ with length n, as follows:

$$\Sigma^0 = \{\epsilon\}$$
$$\Sigma^{n+1} = \{ax \mid a \in \Sigma,\ x \in \Sigma^n\}$$

Example: $\{0,1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$.

Notation: $\Sigma^* = \Sigma^0 \bigcup \Sigma^1 \bigcup \Sigma^2 \ldots$ and
$\Sigma^+ = \Sigma^1 \bigcup \Sigma^2 \bigcup \Sigma^3 \ldots$

Some Properties

The following properties can be proved by induction:
(More on induction next lecture.)

Lemma: *Concatenation is associative:* $\forall x, y, z.\ x(yz) = (xy)z$.
(We shall simply write xyz.)

Lemma: $\forall x, y.\ |xy| = |x| + |y|$.

Lemma: $\forall x.\ x\epsilon = \epsilon x = x$.

Lemma: $\forall n. \forall x.\ |x^n| = n * |x|$.

Lemma: $\forall n. \forall \Sigma.\ |\Sigma^n| = |\Sigma|^n$.

Lemma: $\forall x. \mathrm{rev}(\mathrm{rev}(x)) = x$.

Lemma: $\forall x, y.\ \mathrm{rev}(xy) = \mathrm{rev}(y)\mathrm{rev}(x)$.

Languages

Definition: Given an alphabet Σ, a *language* $\mathcal{L}$ is a subset of Σ^*, that is, $\mathcal{L} \subseteq \Sigma^*$.

Note: If $\mathcal{L} \subseteq \Sigma^*$ and $\Sigma \subseteq \Delta$ then $\mathcal{L} \subseteq \Delta^*$.

Note: A language can be either finite or infinite.

Example: Some languages:

- Swedish, English, Spanish, French, . . . ;
- Any programming language;
- $\emptyset$, $\{\epsilon\}$ and Σ^* are languages over any Σ;
- The set of prime Natural numbers $\{1, 3, 5, 7, 11, \ldots\}$.

Some Operations on Languages

Definition: Given $\mathcal{L}$, $\mathcal{L}_1$ and $\mathcal{L}_2$ languages, we define the following languages:

Union, Intersection, ... : As for any set.

Concatenation: $\mathcal{L}_1\mathcal{L}_2 = \{x_1x_2 \mid x_1 \in \mathcal{L}_1,\ x_2 \in \mathcal{L}_2\}$.

Closure: $\mathcal{L}^* = \bigcup_{n\in\mathbb{N}} \mathcal{L}^n$ where $\mathcal{L}^0 = \{\epsilon\}$, $\mathcal{L}^{n+1} = \mathcal{L}^n\mathcal{L}$.

Note: $\emptyset^* = \{\epsilon\}$ and
$\mathcal{L}^* = \mathcal{L}^0 \cup \mathcal{L}^1 \cup \mathcal{L}^2 \cup \ldots = \{\epsilon\} \cup \{x_1 \ldots x_n \mid n > 0, x_i \in \mathcal{L}\}$

Notation: $\mathcal{L}^+ = \mathcal{L}^1 \cup \mathcal{L}^2 \cup \mathcal{L}^3 \cup \ldots$

Example: Let $\mathcal{L} = \{aa, b\}$, then
$\mathcal{L}^0 = \{\epsilon\}$, $\mathcal{L}^1 = \mathcal{L}$, $\mathcal{L}^2 = \mathcal{L}\mathcal{L} = \{aaaa, aab, baa, bb\}$, $\mathcal{L}^3 = \mathcal{L}^2\mathcal{L}$, ...
$\mathcal{L}^* = \{\epsilon, aa, b, aaaa, aab, baa, bb, \ldots\}$.

How to Prove the Equality of Languages?

Given the languages $\mathcal{L}$ and $\mathcal{M}$, how can we prove that $\mathcal{L} = \mathcal{M}$?

A few possibilities:

- Languages are sets so we prove that $\mathcal{L} \subseteq \mathcal{M}$ and $\mathcal{M} \subseteq \mathcal{L}$;
- Transitivity of equality: $\mathcal{L} = \mathcal{L}_1 = \ldots = \mathcal{L}_m = \mathcal{M}$;
- We can reason about the elements in the language:

  Example: $\{a(ba)^n \mid n \geqslant 0\} = \{(ab)^n a \mid n \geqslant 0\}$ can be proved by induction on n. (More on induction next lecture.)

Algebraic Laws for Languages

All laws presented in slide 14 are valid.

In addition, we have all these laws on concatenation:

Associativity:	$\mathcal{L}(\mathcal{M}\mathcal{N}) = (\mathcal{L}\mathcal{M})\mathcal{N}$
Concatenation is *not commutative:*	$\mathcal{L}\mathcal{M} \neq \mathcal{M}\mathcal{L}$
Distributivity:	$\mathcal{L}(\mathcal{M} \cup \mathcal{N}) = \mathcal{L}\mathcal{M} \cup \mathcal{L}\mathcal{N} \quad (\mathcal{M} \cup \mathcal{N})\mathcal{L} = \mathcal{M}\mathcal{L} \cup \mathcal{N}\mathcal{L}$
Identity:	$\mathcal{L}\{\epsilon\} = \{\epsilon\}\mathcal{L} = \mathcal{L}$
Annihilator:	$\mathcal{L}\emptyset = \emptyset\mathcal{L} = \emptyset$
Other Rules:	$\emptyset^* = \{\epsilon\}^* = \{\epsilon\}$
	$\mathcal{L}^+ = \mathcal{L}\mathcal{L}^* = \mathcal{L}^*\mathcal{L}$
	$(\mathcal{L}^*)^* = \mathcal{L}^*$

Algebraic Laws for Languages (Cont.)

Note: While

$$\mathcal{L}(\mathcal{M} \cap \mathcal{N}) \subseteq \mathcal{L}\mathcal{M} \cap \mathcal{L}\mathcal{N} \quad \text{and} \quad (\mathcal{M} \cap \mathcal{N})\mathcal{L} \subseteq \mathcal{M}\mathcal{L} \cap \mathcal{N}\mathcal{L}$$

both hold, in general

$$\mathcal{L}\mathcal{M} \cap \mathcal{L}\mathcal{N} \subseteq \mathcal{L}(\mathcal{M} \cap \mathcal{N}) \quad \text{and} \quad \mathcal{M}\mathcal{L} \cap \mathcal{N}\mathcal{L} \subseteq (\mathcal{M} \cap \mathcal{N})\mathcal{L}$$

don't.

Example: Consider the case where

$$\mathcal{L} = \{\epsilon, a\}, \quad \mathcal{M} = \{a\}, \quad \mathcal{N} = \{aa\}$$

Then $\mathcal{L}\mathcal{M} \cap \mathcal{L}\mathcal{N} = \{aa\}$ but $\mathcal{L}(\mathcal{M} \cap \mathcal{N}) = \mathcal{L}\emptyset = \emptyset$.

Functions between Languages

Definition: A *function* $f : \Sigma^* \to \Delta^*$ *between 2 languages* should satisfy

$$f(\epsilon) = \epsilon$$
$$f(xy) = f(x)f(y)$$

Intuitively, $f(a_1 \ldots a_n) = f(a_1) \ldots f(a_n)$.

Note: $f(a) \in \Delta^*$ if $a \in \Sigma$.

Overview of Next Lecture (HB3)

Sections 1.2–1.4 in the book and MORE:

- Formal Proofs;
- Inductively defined sets;
- Proofs by (structural) induction.

DO NOT MISS THIS LECTURE!!!