CS 455/555: Mathematical preliminaries

Stefan D. Bruda

Fall 2020

- Sets:
  - Operations: intersection, union, difference, Cartesian product
  - Big $\bigcup$, powerset (2^A)
  - Partition ($\pi \subseteq 2^A$, $\emptyset \notin \pi$, $\forall i \neq j : \pi_i \cap \pi_j = \emptyset$, $\bigcup_{\pi_i \in \pi} \pi_i = A$)
  - Equality
  - De Morgan rules

- Sets:
  - Operations: intersection, union, difference, Cartesian product
  - Big $\bigcup$, powerset (2^A)
  - Partition ($\pi \subseteq 2^A$, $\emptyset \notin \pi$, $\forall i \neq j : \pi_i \cap \pi_j = \emptyset$, $\bigcup_{\pi_i \in \pi} \pi_i = A$)
  - Equality
  - De Morgan rules
- Relations:
  - An *n*-ary relation over a set A: $R \subseteq A^n$
  - Binary relations $R \subseteq A \times A \Rightarrow$ graph representation
    1. reflexive: $\forall a \in A : (a, a) \in R$
    2. symmetric: $\forall a, b \in A : (a, b) \in R \Rightarrow (b, a) \in R$
    3. antisymmetric: $\forall a, b \in A : (a, b) \in R \Rightarrow (b, a) \notin R$
    4. transitive: $\forall a, b, c \in A : (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$
  - 1+4: preorder
  - 1+4+2: equivalence $\Rightarrow$partition in equivalence classes $[a] = \{b : (a, b) \in R\}$
  - 1+4+3: partial order (then total order)

- Functions: $f : A \to B$; special relations; one-to-one, onto, bijection
  - Natural isomorphism = "natural" bijection (e.g. between $A \times B \times C$ and $A \times (B \times C)$, between A and $\{\{a\} : a \in A\}$)

Functions and Cardinality

- Functions: $f : A \to B$; special relations; one-to-one, onto, bijection
  - Natural isomorphism = "natural" bijection (e.g. between $A \times B \times C$ and $A \times (B \times C)$, between A and $\{\{a\} : a \in A\}$)
- Cardinality: Binary relation (equivalence!) $\mathcal{E}$ over the set of all sets
  - $(A, B) \in \mathcal{E}$ also denoted by $|A| = |B| \Rightarrow A$ and B are equinumerous = there exists a bijection $e : A \to B$
  - Interesting kind of sets
    - finite: $(A, \{1, 2, \ldots, n\}) \in \mathcal{E}$ for some $n \in \mathbb{N}$; also written $|A| = n$
    - (infinitely) countable: $|A| = |\mathbb{N}|$ (count the elements)
    - uncountable
  - Is $\mathbb{N} \times \mathbb{N}$ countable?

- Induction: If
  1. $0 \in A$, and
  2. $\forall n : \{0, 1, \ldots, n\} \subseteq A \Rightarrow n + 1 \in A$

  then $A = \mathbb{N}$

- Induction: If
  1. $0 \in A$, and
  2. $\forall n : \{0, 1, \ldots, n\} \subseteq A \Rightarrow n + 1 \in A$

  then $A = \mathbb{N}$
- Pigeonhole principle: If $|A| > |B|$ then there is no one-to-one function $f : A \to B$
  - Useful example: If there is a path between vertices a and b of a graph with n vertices then there is a path between a and b of length at most n

- Induction: If
  1. $0 \in A$, and
  2. $\forall n : \{0, 1, \ldots, n\} \subseteq A \Rightarrow n + 1 \in A$

  then $A = \mathbb{N}$
- Pigeonhole principle: If $|A| > |B|$ then there is no one-to-one function $f : A \rightarrow B$
  - Useful example: If there is a path between vertices a and b of a graph with n vertices then there is a path between a and b of length at most n
- Diagonalization: Given some relation $R \subseteq A \times A$, let

$$R_a = \{b : b \in A \wedge (a, b) \in R\} \qquad D = \{a : a \in A \wedge (a, a) \notin R\}$$

  Then $D \neq R_a$ for any $a \in A$
  - Useful in proofs by contradiction
  - Interesting examples: $2^{\mathbb{N}}$ is uncountable; $[0, 1]$ is uncountable

- $R \subseteq D^{n+1}$ for some $n > 0$, $B \subseteq D$
- B is closed under R if $b_{n+1} \in B$ whenever $b_1, b_2, \ldots, b_n \in B$ and $(b_1, b_2, \ldots, b_n, b_{n+1}) \in R$
- Closure property: "B is closed under R_1, R_2, ..., R_n"
- Let $\mathcal{P}$ be a closure property (under R_1, R_2, ..., R_n) and $A \subseteq D$. Then there exists a minimal B such that $A \subseteq B$ and $\mathcal{P}$ holds for B
  - B is the closure of A under R_1, R_2, ..., R_n
  - Useful example: The reflexive and transitive closure of R is the closure of R under reflexivity and transitivity

ALPHABETS AND STRINGS

- The math of strings of symbols (such as strings of bits)
- Alphabet Σ: a finite set of symbols
- Strings (not sets!) over an alphabet
- The set of all strings over Σ: Σ^*
- Empty string: ε (also λ, in the text e)
- Operations: length ($|w|$), concatenation ($\cdot$ or juxtaposition), substring, suffix, prefix
- Length over a set A: $|w|_A$ is the length of the string w from which all the symbols not in A have been erased
  - Abuse of notation: $|w|_a$ is a shorthand for $|w|_{\{a\}}$
- Exponentiation: $w^0 = \varepsilon$; $w^{i+1} = w^i w$
- Reversal: $w = \varepsilon \Rightarrow w^{\mathbb{R}} = \varepsilon$; for $a \in \Sigma$: $w = ua \Rightarrow w^{\mathbb{R}} = au^{\mathbb{R}}$

- Language: set of strings
- Can be finite, infinite, countable, etc
- Σ^* is a language (countable?)
- Operations: union, intersection, difference, complement ($\overline{A} = \Sigma^* \setminus A$)
- Concatenation: $L_1 L_2 = \{w_1 w_2 : w_1 \in L_1 \wedge w_2 \in L_2\}$
- Kleene star (or closure—under what?):

$$L^* = \{w_1 w_2 \cdots w_n : n \geq 0 \wedge \forall 1 \leq i \leq n : w_i \in L\}$$

- Are there languages that cannot be represented?
- We generally work with mathematical descriptions
- Generators are useful for describing languages
- Generally once the language is described we find convenient to work with a regognition device (is it the case that $w \in L$?) instead

Regular Expressions and Regular Languages

- We start with very simple languages and then we combine them using a set of usual set operations
  - The set of regular languages is then the closure of $\{\{a\} : a \in \Sigma\} \cup \{\emptyset\}$ under concatenation, union, and Kleene star
- Simpler to work with an inductive definition: Regular expressions and their associated languages are defined as follows
  - $\emptyset$ is a regular expression; $\mathcal{L}(\emptyset) = \emptyset$
  - a is a regular expression for all $a \in \Sigma$; $\mathcal{L}(a) = \{a\}$
  - If α and β are regular expressions then so are $\alpha\beta$, $\alpha \cup \beta$, and α^*;
    $\mathcal{L}(\alpha\beta) = \mathcal{L}(\alpha)\mathcal{L}(\beta)$ $\mathcal{L}(\alpha \cup \beta) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta)$ $\mathcal{L}(\alpha^*) = \mathcal{L}(\alpha)^*$
  - Nothing else is a regular expression
- Regular expressions are language generators
- The set REG of regular languages contain exactly all the languages generated by regular expressions