2

May 6, 2016

0.0.1 Languages

A, B, C typical names for languages

Σ^* language of *all* strings over alphabet Σ

$\emptyset$ empty language

$A = \{\omega \in \Sigma^* : |\omega| \text{ is even}\}$

Q: For an alphabet Σ (let us take $\Sigma = \{0, 1\}$). Is the language Σ^* countable or uncountable?

A: Countable.

0.0.2 Lexicographic ordering

The *lexicographic ordering* of Σ^* is the ordering of strings as follows: - order by increasing length - dictionary ordering among strings of the same length

For $\Sigma = \{0, 1\}$, we have: $\epsilon, 0, 1, 00, 01, 10, 11, 000, ...$

Now define a 1-to-1 and onto function $f : \mathbb{N} \to \Sigma^*$ in the natural way:

$0 \to \epsilon, 1 \to 0, 2 \to 1, 3 \to 00, ...$

Proposition: for any alphabet Σ, the language Σ^* is countable.

Q: now consider the set of all languages over $\Sigma = \{0, 1\}$. Is $P(\Sigma^*)$ countable or uncountable?

A: uncountable.

Proof: we know that there exists a bijective function $f : \mathbb{N} \to \Sigma^*$.

We want a bijective function $g : P(\mathbb{N}) \to P(\Sigma^*)$.

$S \subseteq \mathbb{N}$. Let $g(S) = \{f(n) : n \in S\}$.

But that contradicts Cantor's theorem as if there is a bijective function $h : \mathbb{N} \to P(\Sigma^*)$, then $g^{-1} \circ h$ is also bijective.

0.0.3 Deterministic Finite Automata (DFA)

Things to keep in mind as we discuss the definition. 1. Definition is based on sets (and functions, etc). 2. It is a simple and not very powerful mode - but we're starting simple

Def: a DFA is a 5-tuple: $m = (Q, \Sigma, \delta, q_0, F)$ where: 1. Q = finite and nonempty set (whose elements we call *states*) 2. Σ = alphabet 3. δ is a transition function of form $\delta : Q \times \Sigma \to Q$ 4. $q_0 \in Q$ (*start* state) 5. $F \subseteq Q$ (set of *accept* states)

State diagrams: the circle-y pictures

Def: suppose $m = (Q, \Sigma, \delta, q_0, F)$ is a DFA and $\omega \in \Sigma^*$ is a string. The DFA m *accepts* ω if: 1. $\omega = \epsilon$ and $q_0 \in F$, or 2. $\omega = \sigma_1 ... \sigma_n$ where $\sigma_i \in \Sigma$, and there exists states $r_0, ... r_n \in Q$ such that $r_0 = q_0$, $r_n \in F$, and $r_{k+1} = \delta(r_k, \sigma_{k+1})$ for $k = 0, ... n-1$

Remark: m rejects ω if m does not accept ω.

Another way to define acceptance/rejection is to define a function

$$\delta^* : Q \times \Sigma^* \to Q$$

$\delta^*(q, \omega)$ is the state we land on if we start at q and read string ω. It is defined as:

$\delta^*(q, \omega) = q$

$\delta^*(q, \omega\sigma) = \delta(\delta^*(q, \omega), \sigma)$

$\forall q \in Q, \omega \in \Sigma^*, \sigma \in \Sigma$

The language *recognized* by a DFA m is $L(m) = \{\omega \in \Sigma^* : m \text{ accepts } \omega\}$.

$A = \{\omega \in \Sigma^* : |\omega| \text{ is prime}\}$

Def: Let Σ be an alphabet and let $A \subseteq \Sigma^*$. The language A is *regular* if there exists a DFA m such that $A = L(m)$.

Q: Let $\Sigma = \{0, 1\}$. How many regular languages over the alphabet Σ are there?

A: Countably many

Proof: For any $n \geq 1$, there are at most finitely many DFAs with n states (up to renaming the states) $\leq n^{2n} n 2^n$. n possible sinks for each of the $2n$ edges, including into itself. n different starting states, and 2^n different sets of accepting states.

Now create a sequence of lists:

$1 \to$ all DFAs with 1 state,

$2 \to$ all DFAs with 2 states...

The set of regular languages (over any alphabet) is countable $\implies$ there are some languages that are not regular.