Recursive Languages

Lecture 31
Section 11.1

Robb T. Koether

Hampden-Sydney College

Wed, Nov 9, 2016

Outline

Recursive Languages

Definition (Recursive Language)

A language is recursive if there is a Turing machine M that accepts it and that halts on every input. In other words, for every word $w \in \Sigma^*$, M either halts with acceptance, if $w \in L$, or M halts with rejection, if $w \notin L$.

Recursive Languages

- The following languages are recursive.

Recursive Languages

- The following languages are recursive.
  - All regular languages.

Recursive Languages

- The following languages are recursive.
  - All regular languages.
  - All context-free langauges.

Recursive Languages

- The following languages are recursive.
  - All regular languages.
  - All context-free langauges.
  - $\{a^n b^n c^n \mid n \geq 0\}$

Recursive Languages

- The following languages are recursive.
  - All regular languages.
  - All context-free langauges.
  - $\{a^n b^n c^n \mid n \geq 0\}$
  - Many others.

Recursive Languages

- The following languages are recursive.
  - All regular languages.
  - All context-free langauges.
  - $\{a^n b^n c^n \mid n \geq 0\}$
  - Many others.
- Are there languages that are not recursive?

Outline

Recursively Enumerable Languages

Definition (Recursively Enumerable Language)

A language is recursively enumerable if there is a Turing machine that accepts it.

Recursively Enumerable Languages

Definition (Recursively Enumerable Language)

A language is recursively enumerable if there is a Turing machine that accepts it.

- Such a Turing machine may or may not halt in a reject state for words not in the language. It may loop.

Recursively Enumerable Languages

Definition (Recursively Enumerable Language)

A language is recursively enumerable if there is a Turing machine that accepts it.

- Such a Turing machine may or may not halt in a reject state for words not in the language. It may loop.
- If it does always halt, then the language is actually recursive, not just recursively enumerable.

Recursively Enumerable Languages

- The following languages are recursively enumerable.

Recursively Enumerable Languages

- The following languages are recursively enumerable.
  - All recursive languages.

Recursively Enumerable Languages

- The following languages are recursively enumerable.
  - All recursive languages.
  - Many others?

Recursively Enumerable Languages

- The following languages are recursively enumerable.
  - All recursive languages.
  - Many others?
- Are there languages that are not recursively enumerable?

Outline

Countable and Uncountable Sets

Theorem

Let Σ be a finite, nonempty set. Then Σ^* is a countably infinite set.

Countable and Uncountable Sets

Proof.

- Let $\Sigma = \{a_1, \ldots, a_n\}$ for some $n \geq 1$.
- Then we can create an enumeration of Σ^* if we order its member first by length and then, within those groups, order them by their indexes:

$$\underbrace{\lambda}_{0}, \underbrace{a_1, \ldots, a_n}_{1}, \underbrace{a_1a_1, a_1a_2, a_1a_3, \ldots, a_na_n}_{2}, a_1a_1a_1, \ldots$$

- We have seen this ordering before.
- This enumeration demonstrates that the set is countable.

□

Countable and Uncountable Sets

Theorem

Let S be an infinite set. Then $\mathcal{P}(S)$ is a uncountable.

Countable and Uncountable Sets

Proof.

- Let $S = \{a_1, a_2, a_3, \ldots\}$
- Any infinite string of 0's and 1's can be interpreted as representing a subset of S.
  - A 1 in position i means that a_i is in the subset.
  - A 0 in position i means that a_i is not in the subset.
- For example, $0011010\ldots$ represents $\{a_3, a_4, a_6, \ldots\}$.

□

Countable and Uncountable Sets

Proof.

- Now suppose that $\mathcal{P}(S)$ is countable.
- Then its members (the subsets of S) can be listed $S_1, S_2, S_3, \ldots$
- Form a two-way infinite array and consider the diagonal.
- For example,

	a_1	a_2	a_3	a_4	a_5	$\cdots$
S_1	0	0	1	1	0	$\cdots$
S_2	0	1	0	0	1	$\cdots$
S_3	1	1	0	1	0	$\cdots$
S_4	0	0	1	1	1	$\cdots$
S_5	1	1	0	1	1	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

□

Countable and Uncountable Sets

Proof.

- Now suppose that $\mathcal{P}(S)$ is countable.
- Then its members (the subsets of S) can be listed $S_1, S_2, S_3, \ldots$
- Form a two-way infinite array and consider the diagonal.
- For example,

	a_1	a_2	a_3	a_4	a_5	$\cdots$
S_1	0	0	1	1	0	$\cdots$
S_2	0	1	0	0	1	$\cdots$
S_3	1	1	0	1	0	$\cdots$
S_4	0	0	1	1	1	$\cdots$
S_5	1	1	0	1	1	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

□

Countable and Uncountable Sets

Proof.

- Form a binary string that is the exact opposite of the diagonal elements.
- In the example, that string would be $10100\ldots$, representing $\{a_1, a_3, \ldots\}$.
- That set cannot not be in the listing $S_1, S_2, S_3, \ldots$, and that is a contradiction.
- Therefore, $\mathcal{P}(S)$ is uncountable.

□

Outline

Non-Recursively Enumerable Sets

Theorem

There exists a language that is not recursively enumerable. That is, there is a language L such that for every Turing machine M, $L \neq L(M)$.

Proof.

- Assume that $\Sigma \neq \varnothing$ (obviously).

□

Non-Recursively Enumerable Sets

Proof.

- Assume that $\Sigma \neq \varnothing$ (obviously).
- Then Σ^* is an infinite set.

$\square$

Non-Recursively Enumerable Sets

Proof.

- Assume that $\Sigma \neq \varnothing$ (obviously).
- Then Σ^* is an infinite set.
- Each language over Σ is a subset of Σ^*, so (by the previous theorem) there are uncountably many different languages.

□

Non-Recursively Enumerable Sets

Proof.

- On the other hand, there are only countably many Turing machines.

□

Non-Recursively Enumerable Sets

Proof.

- On the other hand, there are only countably many Turing machines.
- Each Turing machine can be represented as a finite binary string, as we saw when designing the universal Turing machine.

□

Non-Recursively Enumerable Sets

Proof.

- On the other hand, there are only countably many Turing machines.
- Each Turing machine can be represented as a finite binary string, as we saw when designing the universal Turing machine.
- The set of all strings over $\{0, 1\}$ (not just those that describe Turing machines) is countably infinite.

□

Non-Recursively Enumerable Sets

Proof.

- Therefore, there can be no onto mapping from the set of Turing machines to the set of all languages.

□

Non-Recursively Enumerable Sets

Proof.

- Therefore, there can be no onto mapping from the set of Turing machines to the set of all languages.
- In particular, the mapping $M \to L(M)$ is not onto.

$\square$

Non-Recursively Enumerable Sets

Proof.

- Therefore, there can be no onto mapping from the set of Turing machines to the set of all languages.
- In particular, the mapping $M \to L(M)$ is not onto.
- So, for some language L, there is no Turing machine M such that $L(M) = L$.

□

Outline

Assignment

Homework

- Section 11.1 Exercises 2, 3, 5, 8, 10, 12, 13 (if and only if).