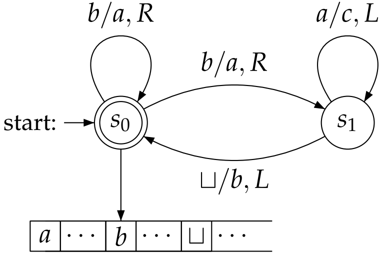
CS 208: Automata Theory and Logic

Part II, Lecture 2: Decidability

S Akshay



.

Department of Computer Science and Engineering,
Indian Institute of Technology Bombay.

Decidable languages

- A TM accepts language L if it has an accepting run on each word in L.
- A TM decides language L if it accepts L and halts on all inputs.

Decidable and Turing recognizable languages

- A language L is decidable (recursive) if there exists a Turing machine M which decides L (i.e., M halts on all inputs and M accepts L).

- A language L is Turing recognizable (recursively enumerable) if there exists a Turing machine M which accepts L.

Algorithms and Decidability

Algorithms $\Longleftrightarrow$ Decidable (i.e, TM decides it)

- A decision problem P is said to be decidable (i.e., have an algorithm) if the language L of all *yes* instances to P is decidable.
- A decision problem P is said to be semi-decidable (i.e., have a semi-algorithm) if the language L of all *yes* instances to P is r.e.
- A decision problem P is said to be undecidable if the language L of all *yes* instances to P is not decidable.

Examples of Decidable languages and problems

- (Acceptance problem for DFA) Given a DFA does it accept a given word?
- (Emptiness problem for DFA) Given a DFA does it accept any word?
- (Equivalence problem for DFA) Given two DFAs, do they accept the same language?

Examples of Decidable languages and problems

- (Acceptance problem for DFA) Given a DFA does it accept a given word?
  - $L^{A}_{DFA} = \{< B, w >| \ A$ is a DFA that accepts input word $w\}$
- (Emptiness problem for DFA) Given a DFA does it accept any word?
  - $L^{\emptyset}_{DFA} = \{< A >| \ A$ is a DFA, $L(A) = \emptyset\}$
- (Equivalence problem for DFA) Given two DFAs, do they accept the same language?
  - $L^{EQ}_{DFA} = \{< A, B >| \ A, B$ are DFAs, $L(A) = L(B)\}$

Examples of Decidable languages and problems

- (Acceptance problem for DFA) Given a DFA does it accept a given word?
  - $L^{A}_{DFA} = \{< B, w >\mid A \text{ is a DFA that accepts input word } w\}$
- (Emptiness problem for DFA) Given a DFA does it accept any word?
  - $L^{\emptyset}_{DFA} = \{< A >\mid A \text{ is a DFA}, L(A) = \emptyset\}$
- (Equivalence problem for DFA) Given two DFAs, do they accept the same language?
  - $L^{EQ}_{DFA} = \{< A, B >\mid A, B \text{ are DFAs}, L(A) = L(B)\}$
- What about NFAs, regular expressions

Relationship among languages

$$\text{Regular} \subsetneq \text{Decidable} \underset{?}{\subseteq} \text{Turing recognizable} \underset{?}{\subseteq} \text{All languages}$$

Relationship among languages

$$\text{Regular} \subsetneq \text{Decidable} \underset{?}{\subseteq} \text{Turing recognizable} \underset{?}{\subseteq} \text{All languages}$$

$$\text{DFA/NFA} < \text{Algorithms/Halting TM} \underset{?}{\leq} \text{Semi-algorithms/TM}$$

Relationship among languages

$$\text{Regular} \subsetneq \text{Decidable} \underset{?}{\subseteq} \text{Turing recognizable} \underset{?}{\subseteq} \text{All languages}$$

$$\text{DFA/NFA} < \text{Algorithms/Halting TM} \underset{?}{\leq} \text{Semi-algorithms/TM}$$

Languages outside R.E.

Thm: There exist languages that are not R.E

Proof: Recall Cantor's argument from First Lecture.

- No. of R.E languages is countable. Why?
- Set S of all words over a finite alphabet Σ is countably infinite.
- Set of all languages over Σ is the set of subsets of S and is therefore uncountable.
- By Cantor's argument, for some such language, there must be no accepting TM.

Diagonalization: go via binary strings over $\{0, 1\}$ which are uncountable.

The acceptance problem for Turing Machines

Given a TM, does it accept a given input word?

$L_{TM}^{A} = \{< M, w >| \ M \text{ is a TM and } M \text{ accepts } w\}$

- L_{TM}^{A} is Turing recognizable: consider TM U which on input $< M, w >$ simulates M on w and accepts if M accepts and rejects if M rejects.

Theorem

L_{TM}^{A} is undecidable.

Proof of undecidability

Suppose $L^A_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$ was decidable.

1. Let H be the deciding TM: on input $\langle M, w\rangle$,

$$H(\langle M, w\rangle) = \begin{cases} \mathit{accept} & \text{if } M \text{ accepts } w \\ \mathit{reject} & \text{if } M \text{ does not accept } w \end{cases}$$

2. Construct TM D which on input $\langle M\rangle$, runs H on input $\langle M, \langle M\rangle\rangle$ and outputs opposite of H.

$$D(\langle M\rangle) = \begin{cases} \mathit{accept} & \text{if } M \text{ does not accept } \langle M\rangle \\ \mathit{reject} & \text{if } M \text{ accepts } \langle M\rangle \end{cases}$$

3. Finally, run D with its own description $\langle D\rangle$ as input!

$$D(\langle D\rangle) = \begin{cases} \mathit{accept} & \text{if } D \text{ does not accept } \langle D\rangle \\ \mathit{reject} & \text{if } D \text{ accepts } \langle D\rangle \end{cases}$$

Proof of undecidability

Diagonalization in the above argument

Enumerate Turing machines in the y-axis and their encodings in the x-axis.

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\ldots$	$\langle D \rangle$	$\ldots$
M_1	$\overline{\textit{accept}}$	*reject*	*accept*	$\ldots$	*accept*	$\ldots$
M_2	*accept*	$\overline{\textit{accept}}$	*accept*	$\ldots$	*accept*	$\ldots$
M_3	*reject*	*reject*	$\overline{\textit{reject}}$	$\ldots$	*reject*	$\ldots$
$\vdots$			$\vdots$		$\vdots$	
$D = M_i$	*reject*	*reject*	*accept*	$\ldots$	$\overline{(??)}$	$\ldots$
$\vdots$			$\vdots$		$\vdots$	

More properties of decidable and r.e. languages

Regular $\subsetneq$ Decidable $\subsetneq$ R.E $\subsetneq$ All languages

What about closure under complementation?

More properties of decidable and r.e. languages

Regular $\subsetneq$ Decidable $\subsetneq$ R.E $\subsetneq$ All languages

What about closure under complementation?

Theorem

If L is decidable, so is $\overline{L}$.

More properties of decidable and r.e. languages

Regular $\subsetneq$ Decidable $\subsetneq$ R.E $\subsetneq$ All languages

What about closure under complementation?

Theorem

If L is decidable, so is $\overline{L}$.

Theorem

L is decidable iff L is R.E and $\overline{L}$ is also R.E.

More properties of decidable and r.e. languages

Regular $\subsetneq$ Decidable $\subsetneq$ R.E $\subsetneq$ All languages

What about closure under complementation?

Theorem

If L is decidable, so is $\overline{L}$.

Theorem

L is decidable iff L is R.E and $\overline{L}$ is also R.E.

So, what about $\overline{L^A_{TM}}$?