Reductions and Rice's theorems

Meenakshi D'Souza

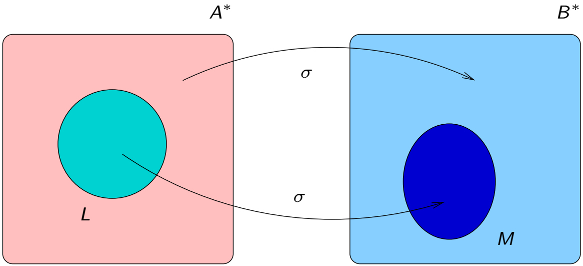
International Institute of Information Technology
Bangalore.

T2 2022-23

Reductions

Let $L \subseteq A^*$ and $M \subseteq B^*$ be two languages. We say L reduces to M and write $L \leq M$ iff there exists a computable map $\sigma : A^* \to B^*$ such that

$$w \in L \text{ iff } \sigma(w) \in M.$$



Reductions

- The function σ need not be one-to-one or onto.
- It must, however, be total and effectively computable. i.e. σ must be computable by a total TM that, on any input x halts with $\sigma(x)$ written on its tape.
- The relation $\leq$ of reducibility between languages is transitive: if $A \leq B$ and $B \leq C$, then $A \leq C$.
  - If σ reduces A to B and τ reduces B to C, then $\tau \circ \sigma$, the composition of σ and τ, is computable and reduces A to C.

Reductions and recursive/re-ness

Theorem

If $A \leq B$ then:

1. *If B is r.e. then so is A.*
2. *If B is recursive then so is A.*

Or to put it differently:

Theorem

If $A \leq B$ then:

1. *If A is not r.e. then neither is B.*
2. *If A is not recursive then neither is B.*

Proof of (1)

1. Suppose $A \leq B$ via σ and B is r.e. Let M be a TM such that $B = L(M)$. Build a machine N for A as follows: on input x, first compute $\sigma(x)$, then run M on input $\sigma(x)$, accepting if M halts.

   Then

$$\begin{array}{lcll} N \text{ accepts } x & \Leftrightarrow & M \text{ accepts } \sigma(x) & \text{(definition of } N\text{)} \\ & \Leftrightarrow & \sigma(x) \in B & \text{(definition of } M\text{)} \\ & \Leftrightarrow & x \in A & \text{(definition of } \sigma\text{)} \end{array}$$

Proof of (2)

1. Recall: A set is A is recursive iff both A and complement of A are r.e.
2. Suppose $A \leq B$ via σ and B is recursive. Note that $A^C \leq B^C$ via the same σ. If B is recursive, then both B and B^C are r.e. Then, by part (1) of the theorem, we know that both A and A^C are r.e., thus A is recursive.

Examples of reductions

Let L be the language $\{M \mid M \text{ accepts } \epsilon\}$. Then

$$\mathrm{HP} \leq L.$$

- Describe a computable map σ which witnesses the reduction.

Hence, since HP is undecidable (i.e. not recursive) so is L.

Examples of reductions

Let L be the language $\{M \mid L(M) \text{ is finite}\}$. Then

$$\text{HP} \leq L.$$

- We describe a computable map σ such that

  $$M\#x \in \text{HP} \Leftrightarrow \sigma(M\#x) \in L$$

  which witnesses the reduction.
- In other words, from M and x, we want to construct a TM M'' such that

  $$M \text{ halts on } x \Leftrightarrow L(M'') \text{ is finite.}$$

Example of reductions

Given $M\#x$, construct a machine M'' that on input y,

- saves y on a separate track
- writes x on the tape
- simulates M on x for $|y|$ steps (it erases one symbol of y for each step of M on x that it simulates)
- accepts if M has *not* halted within that time, otherwise rejects.

$$\begin{aligned} M \text{ halts on } x \quad &\Rightarrow \quad L(M'') = \{y \mid |y| < \text{ running time of } M \text{ on } x\} \\ &\Rightarrow \quad L(M'') \text{ is finite.} \end{aligned}$$

$$\begin{aligned} M \text{ does not halt on } x \quad &\Rightarrow \quad L(M'') = \Sigma^* \\ &\Rightarrow \quad L(M'') \text{ is infinite.} \end{aligned}$$

Hence, since HP is undecidable (i.e. not recursive) so is L.

Examples of reductions

Similarly we can show that:

- $\mathrm{HP} \leq \{M \mid M \text{ accepts a CFL}\}$.
- $\mathrm{HP} \leq \{M \mid M \text{ accepts a recursive language }\}$.

Rice's theorem

Theorem (Rice)

Any non-trivial property of r.e. languages is undecidable.

Rice's theorem

Theorem (Rice)

Any non-trivial property of r.e. languages is undecidable.

Theorem (Rice)

Any non-monotone property of r.e. languages is not even recursively enumerable.

Properties of r.e. languages

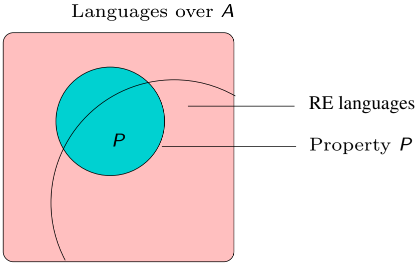
Property P of r.e. languages over an alphabet A is a subset of r.e. languages over A. In other words, a property of the r.e. sets is a map $P : \{ \text{ r.e. subsets of } \Sigma^*\} \to \{\top, \bot\}$, where $\top$ $(\bot)$ represents truth (falsity).

For example, the property of emptiness is represented by the map

$$P(A) = \begin{cases} \top & \text{if } A = \emptyset \\ \bot & \text{if } A \neq \emptyset \end{cases}$$

A property P defines a langauge

$$L_P = \{M \mid L(M) \in P\}.$$



Properties of r.e sets: Examples

- Properties of r.e. sets:
  - $L(M)$ is finite.
  - $L(M)$ is regular/CFL.
  - M accepts 10011, i.e. $10011 \in L(M)$.
  - $L(M) = \Sigma^*$.
- Properties of TMs that are *not* properties of r.e. sets:
  - M has atleast 481 states.
  - M halts on all inputs.
  - M rejects 110011.

Non-trivial properties of r.e. sets

- Non-trivial properties are those that are neither universally true nor universally false.
- For a property to be non-trivial, there must be at least one r.e. set that satisfies the property and at least one that does not.

Proof of part(1) of Rice's theorem

Let P be a non-trivial property of the r.e. sets. Assume without loss of generality that $P(\emptyset) = \perp$. Since P is non-trivial, there must exist an r.e set A such that $P(A) = \top$. Let K be the TM accepting A.
We reduce HP to the set $\{M \mid P(L(M)) = \top\}$, thereby showing that the latter is undecidable.

Proof of part (1) of Rice's theorem

Given $M\#x$, construct a machine $M' = \sigma(M\#x)$ that on input y

- saves y on a separate track
- writes x on its tape
- runs M on input x
- if M halts on x, M' runs K on y and accepts if K accepts.

Now,

$$\begin{array}{lcccc} M \text{ halts on } x & \Rightarrow & L(M') = A & \Rightarrow & P(L(M')) = P(A) = \top \\ M \text{ does not halt on } x & \Rightarrow & L(M') = \emptyset & \Rightarrow & P(L(M')) = P(\emptyset) = \bot \end{array}$$

This constitutes a reduction from HP to the set $\{M \mid P(L(M)) = \top\}$. Since HP is r.e., the latter set is r.e. too.

Part (2) of Rice's theorem

- A property $P : \{\text{ r.e. sets }\} \to \{\top, \bot\}$ of the r.e. sets is called monotone if for all r.e. sets A and B, if $A \subseteq B$, then $P(A) \leq P(B)$.
- Note: Here $\leq$ means less than or equal to in the order $\bot \leq \top$.
- P is monotone if whenever a set has the property, then all supersets of that set have it as well.
- For example, the properties "$L(M)$ is infinite" and "$L(M) = \Sigma^*$" are monotone but "$L(M)$ is finite" and "$L(M) = \emptyset$" are not.

Rice's theorem

Theorem (Rice)

Any non-trivial property of r.e. languages is undecidable.

Rice's theorem

Theorem (Rice)

Any non-trivial property of r.e. languages is undecidable.

Theorem (Rice)

Any non-monotone property of r.e. languages is not even recursively enumerable.