Reductions and Rice's theorems

Deepak D'Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

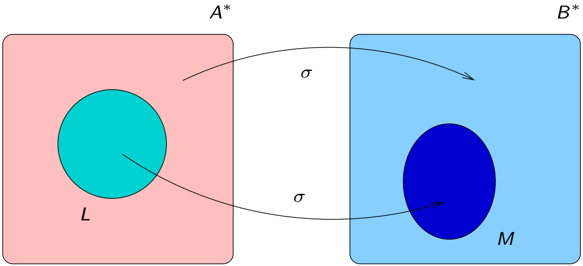
22 November 2016

Outline

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

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



Examples of reductions

- Let L be the language $\{n \mid n \text{ is even }\}$ (with say n encoded in binary). Let L' be the language $\{l\#m\#r \mid l \mod m = r\}$. Then $L \leq L'$ via the computable map $n \mapsto n\#2\#0$.
- Does L' reduce to L?
- 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.

Reductions and recursive/re-ness

Theorem

If $L \leq M$ then:

1. *If M is r.e. then so is L.*
2. *If M is recursive then so is L.*

Or to put it differently:

Theorem

If $L \leq M$ then:

1. *If L is not r.e. then neither is M.*
2. *If L is not recursive then neither is M.*

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 $\{M \mid M$ accepts a regular language$\}$. Then

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

- Describe a computable map σ which witnesses the reduction.
- Hence, since $\neg\mathrm{HP}$ is undecidable (i.e. not recursive) so is L.
- In fact, since $\neg\mathrm{HP}$ is not r.e., we can say that L is not r.e..

Rice's theorem

Theorem (Rice)

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

Rice's theorem

Theorem (Rice)

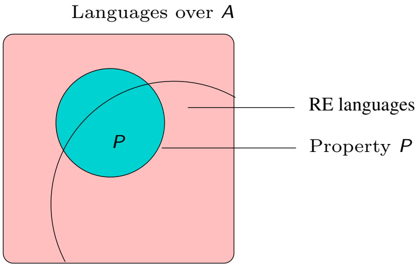
Any non-trivial property of r.e. languages is undecidable.

Theorem (Rice)

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

Properties of languages

A property P of languages over an alphabet A is a subset of languages over A.



Non-trivial and montone properties

- A property P is a non-trivial property of r.e. languages, if there is at least one r.e. language L satisfying P, and another L' not satisfying P.

Non-trivial and montone properties

- A property P is a non-trivial property of r.e. languages, if there is at least one r.e. language L satisfying P, and another L' not satisfying P.
  - E.g. "is empty" is non-trivial

Non-trivial and montone properties

- A property P is a non-trivial property of r.e. languages, if there is at least one r.e. language L satisfying P, and another L' not satisfying P.
    - E.g. "is empty" is non-trivial
    - "is not accepted by a TM" is trivial.
- A property P of languages is monotone (w.r.t r.e. languages) if for all r.e. sets A and B, whenever $A \subseteq B$ and $P(A)$, we have $P(B)$.
- In other words, P is monotone if whenever a set has the property P, all its supersets have it as well.

Non-trivial and montone properties

- A property P is a non-trivial property of r.e. languages, if there is at least one r.e. language L satisfying P, and another L' not satisfying P.
  - E.g. "is empty" is non-trivial
  - "is not accepted by a TM" is trivial.
- A property P of languages is monotone (w.r.t r.e. languages) if for all r.e. sets A and B, whenever $A \subseteq B$ and $P(A)$, we have $P(B)$.
- In other words, P is monotone if whenever a set has the property P, all its supersets have it as well.
  - "is infinite" is monotone,

Non-trivial and montone properties

- A property P is a non-trivial property of r.e. languages, if there is at least one r.e. language L satisfying P, and another L' not satisfying P.
  - E.g. "is empty" is non-trivial
  - "is not accepted by a TM" is trivial.
- A property P of languages is monotone (w.r.t r.e. languages) if for all r.e. sets A and B, whenever $A \subseteq B$ and $P(A)$, we have $P(B)$.
- In other words, P is monotone if whenever a set has the property P, all its supersets have it as well.
  - "is infinite" is monotone,
  - "$L(M)$ is finite" is not monotone.

Rice's theorems

For a property P, we define

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

Theorem (Rice 1953)

Any non-trivial property of r.e. languages is undecidable. That is, if P is a non-trivial property of r.e. languages, then the language L_P is not recursive.

Theorem (Rice 1956)

Any non-monotone property of r.e. languages is not even recursively enumerable. That is, if P is a non-monotone property of r.e. languages, then the language L_P is not even recursively enumerable.

Proof of Rice's Theorem 1

- Let P be a non-trivial property of r.e. languages. Then there are TM's K and T such that $L(K)$ satisfies P and $L(T)$ does not satisfy P.
- We show that $L_P = \{M \mid L(M) \text{ satisfies } P\}$ is not recursive.
- Case 1: If $\emptyset$ does not satisfy P. We reduce HP to L_P.
- 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 as M on input x
  - if M halts on x, M' runs as K on y and accepts iff K accepts.

$$L(M') = \begin{cases} L(K) & \text{if } M \text{ halts on } x \\ \emptyset & \text{if } M \text{ does not halt on } x. \end{cases}$$

Proof of Rice's Theorem 1

- Case 2: If $\emptyset$ satisfies P. We reduce $\neg\mathrm{HP}$ to L_P.
- 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 as M on input x
  - if M halts on x, M' runs as T on y and accepts iff T accepts.

$$L(M') = \begin{cases} \emptyset & \text{if } M \text{ does not halt on } x \\ L(T) & \text{if } M \text{ halts on } x. \end{cases}$$

Proof of Rice's Theorem 2

- Let P be a non-monotone property of r.e. sets.
- Then there are TM's K and T such that $L(K) \subseteq L(T)$ and $L(K)$ satisifes P but $L(T)$ does not.
- We show $\neg\mathrm{HP} \leq L_P$.
- Given $M\#x$ output the description of M' that
  - Given input y on Tape 1.
  - Copies y on Tape 2, writes x on Tape 3
  - Run (in an interleaved fashion) as M on x, K on y, and T on y.
  - accept iff either
    - K accepts y, or,
    - M halts on x and T accepts y.

Proof of Rice's Theorem 2

Notice that:

$$L(M') = \begin{cases} L(K) & \text{if } M \text{ does not halt on } x \\ L(T) & \text{if } M \text{ halts on } x. \end{cases}$$

Some applications

From Rice's Theorem 1:

- "Accepts ϵ" is undecidable.
- "Accepts an infinite language" is undecidable.

$$\{M \mid M \text{ accepts an infinite language}\}.$$

From Rice's Theorem 2:

- "Accepts the empty language" is "highly" undecidable (non-r.e.).
- "Accepts a finite language" is highly undecidable (non-r.e.).

$$\{M \mid M \text{ accepts a finite language}\}.$$

- "Accepts a regular language" is highly undecidable (non-r.e.).