CSCC63 Winter 2022 Tutorial 1

Turing Machines & Decidability
Languages on Turing Machines
Dovetailing
Reductions

Richard Hong
richard.hong@mail.utoronto.ca

Office Hours: Mondays 5-6pm

A language is Recognizable if:
if a turing machine M accepts all elements in L,
and for the elements not in L, M can either loops or reject

A language L is Co-Recognizable, if:
the complement is recognizable.

A language L is Decidable if:
L is recognizable and co-recognizable

Suppose L and L' is decidable,

$L \cap L'$ is decidable, true or false

L is decidable => exists M decides L
L' is decidable => exists M' decides L'

M'' will decide the intersection:

M'' on input <x>:
  run x on M
  run x on M'
  if both accept, accept
  reject

if x in the intersection, M'' accepts x
when x is not in the intersection, M'' rejects x

Suppose L and L' is decidable,

$L \cup L'$ is decidable, true or false

L is decidable => M that decides L
L' is decidable => M' that decides L'

build a TM that runs the input on both M and M',
accept it either accepts, reject otherwise

L = { <M> | M accepts some string ending in 101 }
Is this recognizable.

Let x_i be an enumeration over {0, 1}* that ends in 101

```
R on input <M>:
    for i = 0 to inf:
      for k = 0 to i:
        [run M on x_k for i steps]
        if M accepts, we accept
```

if M is in L, then M will eventually accept one of x_k in i steps.

Suppose L and L' are recognizable languages,
L union L' is recognizable, true or false?

L is recognizable => exists M that recognizes L
L' is recognizable => exists M' that recognizes L'

```
R on input <x>:
  for i = 0 ... inf:
    run M on x for i steps
    run M' on x for i steps
    if either accept, accept
    if both reject, reject
```

suppose that x is in L union L', this means x will be recognized by M or M' then eventually M or M' will accept, thus our TM will accept.

suppose that x is not in L union L', if M and M' both reject x, our TM will also reject x, if either or both M and M' loops, our TM will loop.

Mapping Reductions

let L and L' be languages,
L <= L' if there is a fuction f,
such that x in L iff f(x) in L'

if L <= L' and L' is decidable => L is decidable
if L <= L' and L is undecidable => L' is undecidable

if L <= L' and L' is recognizable => L is recognizable
if L <= L' and L is unrecognizable => unrecognizable

Proof: if L is a recognizable language and L <= L-complement, then L is decidable

$$A \le_m B \iff \overline{A} \le_m \overline{B}$$

L-complement <= L-complement-complment
L-complement <= L, L is recognizable
L-complement is recognizable

recognizable AND corec => decidable

Let

$$L = \begin{cases} \{x \in \{0,1\}^*: x \text{ ends in } 0\}, & \text{if it rained in Paris on June 3, 1628} \\ \{x \in \{0,1\}^*: x \text{ ends in } 1\}, & \text{otherwise} \end{cases}$$

Is L a regular language?

(0 + 1)* 0

(0 + 1)* 1

L = { <M> | M rejects/loops on all strings that end in 101 }

HALT = { <M, w> | M halts on w }
HALT-complement = { <M, w> | M loops on w }
HALT-complement is not recognizable, we'll try to reduce HALT-complement to L

```
P on input <M, w>
  define M' on input <x>:
    run M on w
    accept x
return <M'>
```

<M, w> in HALT-C <=> M' in L

if <M, w> is in HALT-C then M' is in L
then M loops on w, M' will loop on all strings => M' never accepts => M' rejects/loops on all strings ending in 101

if <M, w> in HALT then M' is not in L
if M halts on w, M' will accept all strings => M' never reject or loop on any string => M' is not in L

HALT-complement is not recognizable, neither is L