Lec 15. Universal TM, diagonal method

Eunjung Kim

Hardwired TM to programmable TM!

- TM is defined by its transition function.
- This means that one TM can compute (recognize or decide) a single function (language).
- One TM, useful for a single purpose only.
  $\rightsquigarrow$ hardwired as produced in the factory.
- But computer as we know is an all-round player with programs.
  $\rightsquigarrow$ stored-program computer, universal.
- Universal TM, the mathematical model that embodies this historic transition.

Hardwired TM to programmable TM!

Key insight

TM not only ‘receives’ an input string, but TM itself can be an input string (once appropriately encoded as a string).

Hardwired TM to programmable TM!

Key insight

TM not only 'receives' an input string, but TM itself can be an input string (once appropriately encoded as a string).

- Let's build a super TM U which reads an arbitrary TM M and an input w to M, and does what M would do on the input w.

Hardwired TM to programmable TM!

Key insight

TM not only 'receives' an input string, but TM itself can be an input string (once appropriately encoded as a string).

- Let's build a super TM U which reads an arbitrary TM M and an input w to M, and does what M would do on the input w.
- If U can simulate any other TM, with U we can do any computation that any TM M can do by loading (reading) M and an input to M; instead of using all sorts of TM's, we use a single TM U - a universal TM.

Hardwired TM to programmable TM!

Key insight

TM not only ‘receives’ an input string, but TM itself can be an input string (once appropriately encoded as a string).

- Let’s build a super TM U which reads an arbitrary TM M and an input w to M, and does what M would do on the input w.
- If U can simulate any other TM, with U we can do any computation that any TM M can do by loading (reading) M and an input to M; instead of using all sorts of TM’s, we use a single TM U - a universal TM.
- Turing proved that a universal TM exists. A couple of legendary scientists and mathematicians including Turing himself realized this concept in the 40’s, the earliest versions of modern-day computers.

Encoding a Turing machine

Encoding of TM

1. Consider TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$.
2. Codifying each component of M.
   - $Q = \{q_1, \ldots, q_s\}$
   - q_1 is interpreted as the start state, q_2 accept state, q_3 reject state.
   - $\Gamma = \{a_1, \ldots, a_t\}$.
   - Left header move is associated with 1, Right header move with 2.
3. A transition $\delta(q_h, a_i) = (q_j, a_k, L)$ is represented as a 5-tuple of numbers; $(h, i, j, k, 1)$
4. 5-tuple expression of a transition as a $\{0, 1\}$-string: $0^h10^i10^j10^k10$
5. TM is expressed as a $\{0, 1\}$ string by
   - encoding each transition using the above scheme
   - concatenation all transitions, each transition separated by 11 (a pair of 1's).

Encoding TM: example

TM $M = (\{q_1, q_2, q_3\}, \{0, 1\}, \{0, 1, B\}, \delta, q_1, q_{accept} = q_2, q_{reject} = q_3)$.

$\delta(q_1, 1) = (q_3, 0, R)$	0100100010100
$\delta(q_3, 0) = (q_1, 1, R)$	0001010100100
$\delta(q_3, 1) = (q_2, 0, R)$	00010010010100
$\delta(q_3, B) = (q_3, 1, L)$	0001000100010010

0100100010100110001010100100110001001001010011000100010001001 0

Universal Turing machine

(Rather informal) definition

Let τ be an encoding scheme of TM and an input string.
A Turing machine U is called a universal Turing machine with encoding scheme τ if it accepts a string s if and only

1. $s = \tau(M) \circ \tau(w)$ for some TM M and a string w of alphabet of M, and
2. M accepts w.

Universal Turing machine

Gödel showed that there exists a universal Turing machine U.

U has 3 tapes.

1. Input tape: the encoding of M and the encoding of an input w to M (separated by 111) is loaded here. Never altered.
2. Simulation tape: whatever happens in the (single) tape of M happens M is simulated (replicated) here.
3. State tape: the state of M during the execution on w is written here.

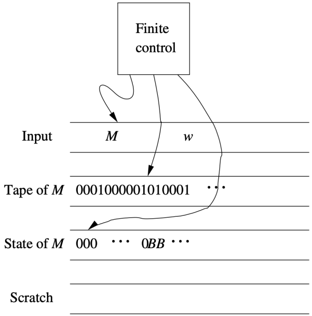

Figure 1.61, Sipser 2012.

All languages TM-recognizable?

No. A fundamental consequence of uncountability of $\mathbb{R}$, and that TM has a finite description.

Outline

Consider the alphabet $\{0, 1\}$.

1. $\{0, 1\}^*$ have the same size as $\mathbb{N}$.
2. the collection of all languages over $\{0, 1\}$ have the same size as $2^{\mathbb{N}}$.
3. $2^{\mathbb{N}}$ is uncountable while $\mathbb{N}$ is countable.
4. the collection of all Turing machines have the same size as $\mathbb{N}$
5. at least one language over $\{0, 1\}$ does not have TM recognizing it.

Countable versus uncountable

The size of a set

- A function φ from A to B is a bijection if it is one-to-one (injection) and onto (surjection).
- We say that two sets A and B have the same size if there is a bijection from A to B.
- A set is countable if it is finite or has a bijection to $\mathbb{N}$.
- A set is uncountable if it is not countable.

Countable sets

Having a bijection from $\mathbb{N}$ to a set A is equivalent to listing all elements of A (the list can be infinite).

- $2\mathbb{N}$
- $\mathbb{Z}$
- $\{0, 1\}^*$
- Σ^* for any finite set Σ
- the set of all rational numbers

Countable sets: rational numbers

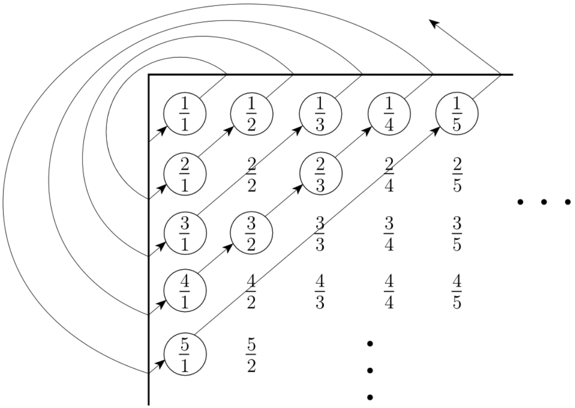

Figure 4.16, Sipser 2012.

Uncountable sets

$\mathbb{R}$ and $2^{\mathbb{N}}$ are uncountable

1. Suppose the contrary; let φ be a bijection from $\mathbb{N}$ to $2^{\mathbb{N}}$ (or to $\mathbb{R}$).
2. Goal: construct an element $X \in 2^{\mathbb{N}}$ (or $x \in \mathbb{R}$) which is not listed by φ $\rightsquigarrow$ contradition.
3. Constructing such an element is possible via diagonal argument.

Diagonal argument: φ lists all real numbers in $[0, 1]$

- Rows are indexed by $1, 2, \ldots$, i.e. $\mathbb{N}$
- i-th row corresponds to the real number $\varphi(i)$, with j-th entry being the j-th digit after the decimal separator.
- Diagonalization step: construct a new real number which is not listed by φ by perturbing all the diagonal entries.

Diagonal argument for uncountability of $[0, 1]$

0.8	1	3	4	2	0	8 ⋯
0.0	1	1	2	1	9	0 ⋯
0.2	0	3	1	4	1	3 ⋯
0.7	0	3	4	4	1	3 ⋯
0.1	0	2	7	4	9	3 ⋯
0.3	1	0	3	6	0	1 ⋯
0.2	4	3	1	4	7	7 ⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋱

$\rightsquigarrow$ consider a real number $x = 0.\bar{8}\bar{1}\bar{3}\bar{4}\bar{4}\bar{0}\bar{7} \cdots = 0.7243186 \cdots$
The perturbation on each digit can be arbitrary (just avoid using 0 and 9).

x is not listed by φ!

Diagonal argument for uncountability of $2^{\mathbb{N}}$

Diagonal argument: suppose φ lists all elements in $2^{\mathbb{N}}$.

- Rows and columns are indexed by $1, 2, \ldots$, i.e. $\mathbb{N}$
- i-th row corresponds to the set $\varphi(i)$ of $2^{\mathbb{N}}$, with j-th entry being 1 if and only if j is in the set.
- Diagonalization step: construct a new set which is not listed by φ by flipping all the diagonal entries.

0	0	1	1	1	0	1	$\cdots$
0	1	1	1	1	1	0	$\cdots$
1	0	1	1	0	1	0	$\cdots$
1	0	1	0	1	1	0	$\cdots$
0	0	1	1	1	0	1	$\cdots$
0	1	0	1	1	0	1	$\cdots$
1	1	1	1	0	0	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	

Consider the set

$$X = \bar{0}\bar{1}\bar{1}\bar{0}\bar{1}\bar{0}\bar{0} \cdots = 1001011 \cdots$$

$\rightsquigarrow$ X is not listed by φ!

Countable or uncountable?

- The collection of all languages over $\{0, 1\}$?
- $\{\tau(M) \subseteq \{0, 1\}^* : M \text{ is a Turing machine}\}$?
- The collection of all languages over $\{0, 1\}$ recognizable by some Turing machine?

Language unrecognizable by TM

- The collection of all languages over $\{0, 1\}$? Uncountable.
- $\{\tau(M) \subseteq \{0, 1\}^* : M \text{ is a Turing machine}\}$? Countable.
- The collection of all languages over $\{0, 1\}$ recognizable by some Turing machine? Countable.

Language unrecognizable by TM

- The collection of all languages over $\{0, 1\}$? Uncountable.
- $\{\tau(M) \subseteq \{0, 1\}^* : M \text{ is a Turing machine}\}$? Countable.
- The collection of all languages over $\{0, 1\}$ recognizable by some Turing machine? Countable.

Unrecognizable

There is a language which cannot be recognized by any Turing machine.