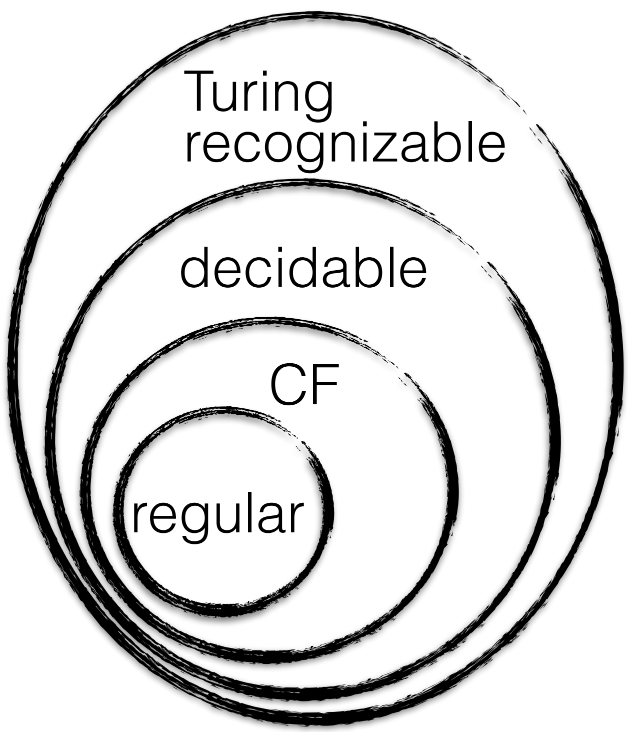
Decidability and Undecidability

2/17/2016

Pete Manolios

Theory of Computation

Models of Computation

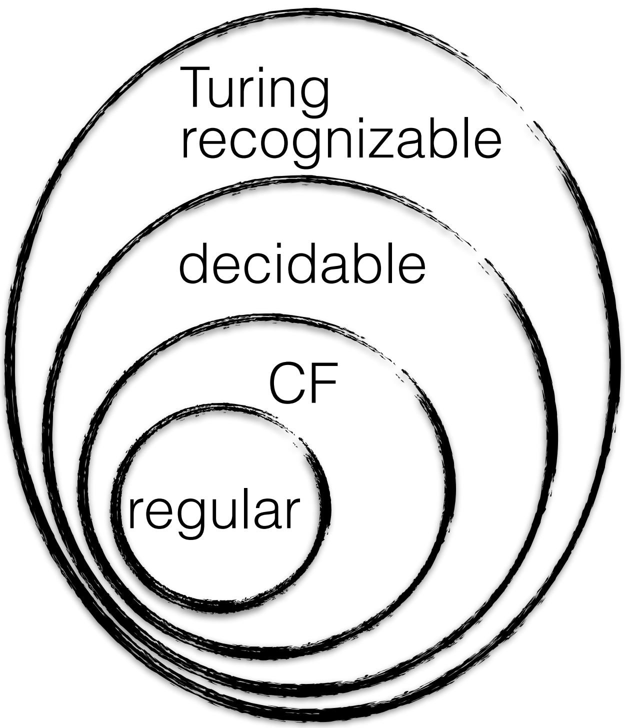


- Regular: finite state machine
- CF: + stack
- Turing machine: + infinite tape
- Decidable (recursive): yes/no
- Recognizable (r.e.): yes

Decidability

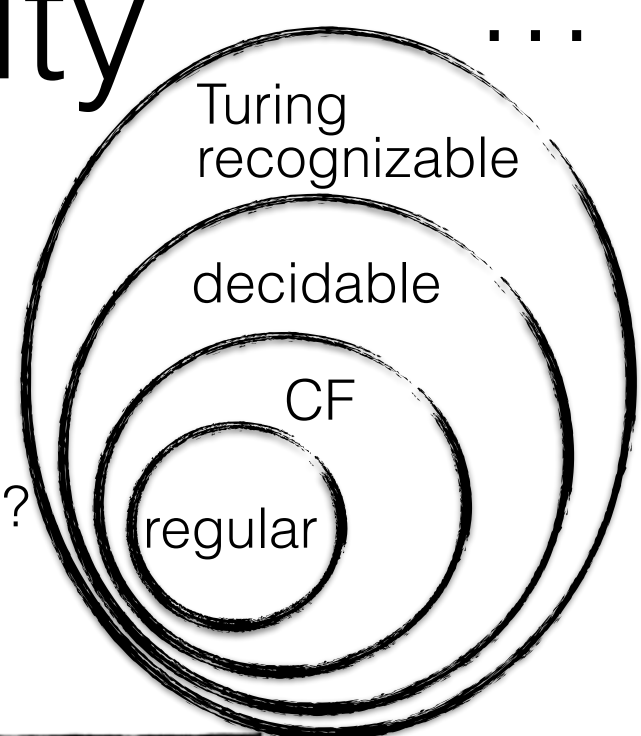
- We will use Church-Turing thesis
- What that means is we'll describe and think about algorithms just like you did in algorithms class
- Because using TMs is really tedious and painful
- Because we "know" that TMs ≡ pseudo code

Decidability

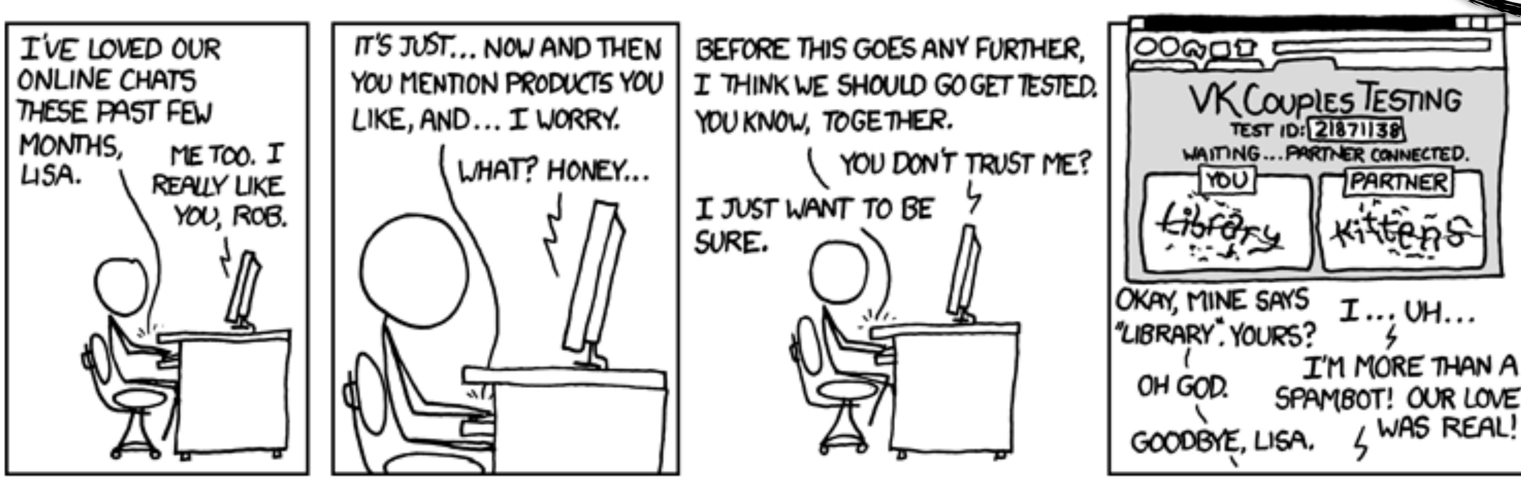


- D is a DFA that accepts w
- N is an NFA that accepts w
- D is a DFA that accepts a non-empty language
- A, B are DFAs and L(A) = L(B)
- C is a CFL that accepts w
- C is a CFL that accepts a non-empty language
- But, A, B are CFLs and L(A) = L(B) **not** decidable
- When the model of comp increases in power
- Your ability to analyze it decreases

Undecidability



- Limits of what can be done with a computer
- Of broad intellectual, philosophical interest
  - Can humans solve problems TMs can't?
  - Turing test: can machines behave like humans?
  - Can machines have consciousness?



http://imgs.xkcd.com/comics/suspicion.png

Counting Infinities

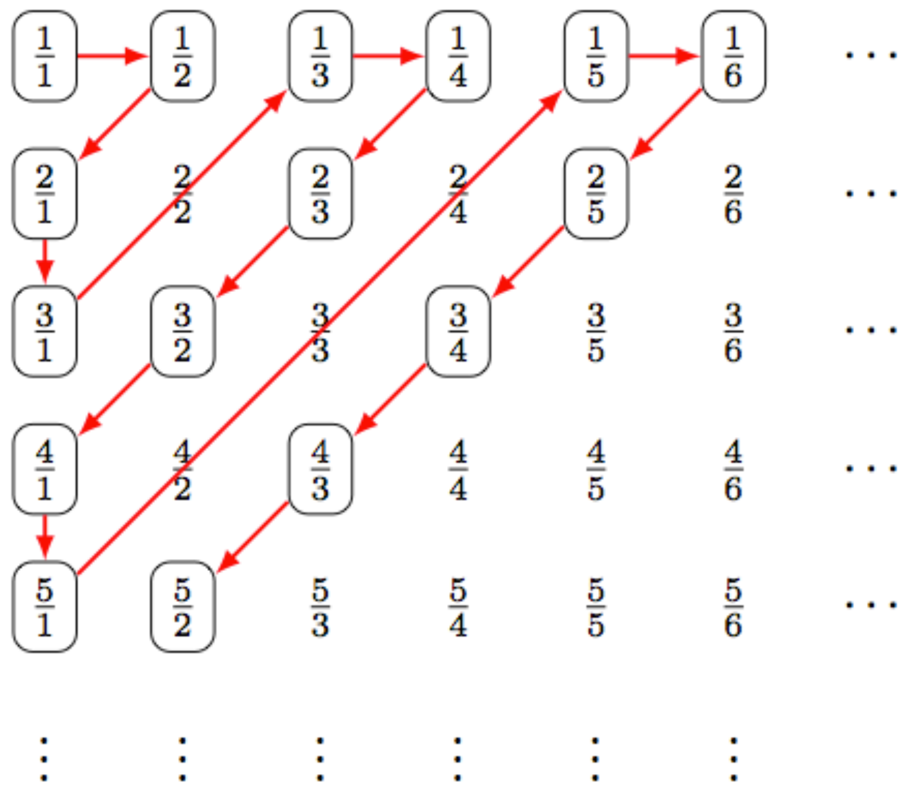
- f: A → B is *injective* or *one-to-one* if it $a \neq b \Rightarrow f(a) \neq f(b)$
- f: A → B is *surjective* or *onto* if it $\cup_{a \in A} \{f(a)\} = B$
- f is *bijective* or a *correspondence* if it is both injective and surjective
- If f: A → B is bijective then each element of A maps to a unique element of B and conversely
- Given A,B if ∃ a bijection f: A → B then |A| = |B|: they have the same size
- This makes intuitive sense for finite sets, but has non-intuitive consequences for infinite sets
- $|\{a, b, c, d\}| = |\{1, 21, 3, 2\}| = |\{d, a, f, b, d, a\}| = 4$
- $|\mathbb{N}| =? |\mathbb{N} \setminus \{0, 1, 2\}|$
- $|\mathbb{N}| = |\{n \in \mathbb{N}: n \text{ is even}\}| = |\mathbb{Z}| = |\mathbb{Q}| = \omega$
- If $|A| \leq \omega$ it is *countable.* ω is the first infinite *ordinal number*.

$\mathbb{N}$	: 0, 1, 2, 3, 4, …
$\mathbb{N} \setminus \{0, 1, 2\}$	: 3, 4, 5, 6, 7, …
$\mathbb{Z}$	: 0, 1, -1, 2, -2, …

Theorem: $|\mathbb{Q}| = \omega$

Have to enumerate the rationals. Here's how:

Note: no duplicates



https://divisbyzero.files.wordpress.com/2013/04/screen-shot-2013-04-16-at-9-23-14-pm.png

Theorem: A countable union of countable sets is countable!

Theorem: $|\mathbb{R}| > \omega$

- $\mathbb{R}$ is *uncountable*: infinite *and* no bijection between $\mathbb{R}$ and ω
- Clearly $|\mathbb{R}| \geq \omega$. We show $|\mathbb{R}| \neq \omega$.
- The proof is by contradiction.
- Suppose that there is a bijection, say:
- We derive a contradiction by showing that it can't include every real number
- Select r_i to differ from digit i of f(i)
- Don't use 0, 9 (because 0.9999… = 1.0000…)
- We showed $|\mathbb{R}| > \omega$ and that $|A| > \omega$ for A is any non-empty interval of reals
- This technique is called *diagonalization* and is due to Cantor (1873)

$\mathbb{N}$	$\leftrightarrow$	*reals in* (0,1)
1	$\leftrightarrow$	.835987...
2	$\leftrightarrow$	.250000...
3	$\leftrightarrow$	.559423...
4	$\leftrightarrow$	.500000...
5	$\leftrightarrow$	.728532...
6	$\leftrightarrow$	.845312...
⋮		⋮
n	$\leftrightarrow$	$.r_1 r_2 r_3 r_4 r_5 \dots r_n \dots$
⋮		⋮

http://tispaquin.blogspot.com/2010/02/what-is-universe-expanding-into.html

Existence of the Undecidable

- There exist languages that are not Turing-recognizable (R.E.)
- So they are also not Turing-decidable (R.) either
- *And* it turns out that *most* languages are not Turing-recognizable!
- Observe: If $|\Sigma| \leq \omega$ then the set of all strings, Σ^*, is countable
- Observe: The set of all TMs is countable (each is described by a finite string of symbols over a finite alphabet)
- Observe: $B = \{0,1\}^{\omega}$ is uncountable (binary representation of reals in [0..1])
- Observe: There is a bijection between $\mathscr{L}$, the set of languages, and B. Use the characteristic function: given $L \in \mathscr{L}$, $f(L) = \langle s_1 \in L, s_2 \in L, s_3 \in L, \ldots \rangle$
- So, $|\mathscr{L}| > \omega$ and most languages are not Turing-recognizable

A_{TM} is R.E.

- A_{TM} = {<M,w>: M is a TM that accepts w}
- Theorem: A_{TM} is R.E. (Turing recognizable)
- Proof: Consider TM U: On input <M, w> it runs M on w. If M halts and accepts w, accept. If M halts and rejects w, reject.
- Note: U is a universal Turing machine

A_{TM} is Undecidable

- Theorem: A_{TM} is Undecidable. (A_{TM} = {<M,w>: M is a TM that accepts w})
- Proof: Suppose there exists a TM H that decides A_{TM}. Then, for any input <M,w>, H accepts if M accepts w and rejects otherwise.
- Consider a TM D that takes an input <M>, the description of M, and takes the following steps.
    - Run H on <M,<M>>
    - If H accepts, reject
    - If H rejects, accept
- Since H is a decider, D is also a decider.
- Consider D's output on <D>. If D accepts, then this implies that according to H, D rejects <D>. If D rejects, then this implies that according to H, D accepts <D>. But this is a contradiction.

Diagonalization?

- Another way to see this is that we have essentially proved that the language {<M> : M accepts <M>} is undecidable. How did we do this?
- Number the machines M_1, M_2, ... Suppose the above language is decidable by a TM E.
- Define D to be a machine that on input <M>, accepts if E rejects <M>, and rejects if E accepts <M>.
- This is precisely flipping the diagonal entries of the matrix in which the columns list the machines M_1, M_2, ..., and the rows list the inputs $<M_1>$, $<M_2>$, ...
- If D is on this list, then we obtain a contradiction.

L and ¬L are RE then L is R.

- ¬L is the complement of L: $\Sigma^* \setminus L$
- Theorem: If L and ¬L are Turing-recognizable, then L is decidable.
- Proof: Let M_1 and M_2 be TMs that recognize L and ¬L. Given a string w, exactly one of the following happens
  - M_1 accepts w or M_2 accepts w
- TM M for deciding L simulates M_1 and M_2 in parallel, running one step of each on w.
- Within a finite number of steps, one of them will halt and accept.
- If M_1 accepts, then M accepts. If M_2 accepts, then M rejects.

$\neg A_{TM}$ is not RE

- Corollary: $\neg A_{TM}$ is unrecognizable (not RE)
- What is $\neg A_{TM}$?
- {<M, w>: M is not a TM or M does not accept w}
- Proof: A_{TM} is not decidable, so by previous theorem either A_{TM} or $\neg A_{TM}$ is not RE, but A_{TM} is RE, so $\neg A_{TM}$ is not.

Halting Problem

- $HALT_{TM}$ = {<M, w>: M halts on w}
- Theorem: $HALT_{TM}$ is undecidable.
- Proof: We show that if $HALT_{TM}$ is decidable, then so is A_{TM}.
- Preview of reduction: We reduce from A_{TM} to $HALT_{TM}$ ($A_{TM} \leq HALT_{TM}$).
- Suppose H is the decider for $HALT_{TM}$. Then the decider A for A_{TM} is as follows. On input <M, w>, A calls H on input <M, w>. If H accepts, then A runs M on w and accepts if M accepts w, rejecting otherwise. If H rejects, then A rejects.
- Consider <M,w> in A_{TM}. Since M accepts w, M halts on w. So H accepts <M, w>. Since M accepts and halts on w, A's call of M on w terminates in an accept state.
- Consider <M,w> not in A_{TM}. There are two cases. The first is when M halts on w and rejects w. So H accepts <M, w>. A's call of M on w terminates in a reject state. The second case is when M does not halt on w. So H rejects <M, w>, and so does A.