COSC 341
Theory of Computing
Lecture 14
Recursivity, Universal Turing Machines

Stephen Cranefield
stephen.cranefield@otago.ac.nz

Lecture slides (mostly) by Michael Albert

Keywords: Recursivity, Universal Turing machines

What do we call languages that TM's accept?

A language L is recursively enumerable (RE) if there is a TM M that accepts it.

- That means that if $w \in L$ then M halts and accepts on input w.
- If $v \notin L$, then running M on v might halt and reject, might crash, or might not halt at all.
- That's a bit unsatisfactory – so long as M keeps running, we're uncertain about the status of the input (is it in L or not?)

What do we call languages that TM's really accept?

A language L is recursive if there is a TM M that *halts on all inputs* and accepts only (and exactly) the words in L.

- That's better – M is now a decision procedure for L.
- For that reason, we sometimes say that L is a decidable language, or just that L is decidable.
- On any input, w, we will (eventually) know whether or not w belongs to L.

An observation

Theorem

If both L and the complement, $\overline{L} = \Sigma^* \setminus L$, of L are recursively enumerable, then L is recursive.

- Use a three-tape machine with tapes called: input, in-L, and out-L.
- Start with the real input, w, on the input tape.
- Copy it to the other two tapes.
- Now run the TM that will halt if w is in L on the in-L tape, and the TM that will halt if w is not in L on the out-L tape.
- As soon as one of them halts, halt and accept if it was the in-L machine, and reject if it was the out-L machine.

Language enumerators

What if we want to want to list the elements of a language. What does that involve?

- A machine M enumerates L if it writes (all and only) the elements of L on an output tape (eventually).
- We need to be careful here – it must be the case that the write head on the output tape always moves to the right.
- And some punctuation symbol separates the elements of L from one another.
- Otherwise, we can't be sure at any point the we've actually "seen" some element (it might be erased or extended).
- Repetitions are allowed, but don't need to be.

Connections

Theorem

A language is recursively enumerable if and only if it is enumerated by some Turing machine.

- Given an enumerator, to build an acceptor, just add a module that takes an input word w, runs the enumerator, and checks each time the enumerator produces a word whether or not it's equal to w (and accepts and halts if it is).
- Given an acceptor (which we assume never crashes) build an enumerator by implementing “for each k from 0 to infinity, run the acceptor for (up to) k steps on each possible input word of length $\leq k$, adding any accepted words to the output tape”.

Enumeration by length

A machine M enumerates L by length if it enumerates L and in the output all the words of length 0 precede those of length 1 which precede those of length 2 . . .

Theorem

A language L can be enumerated by length if and only if it is recursive.

- If L is recursive, take a machine M that accepts it and halts on all inputs. Run it on each string of length 0, then each string of length 1 and so on. Each time a string is accepted, add it to an output tape. The resulting machine enumerates L by length.
- Conversely, suppose that L can be enumerated by length. If L is finite then it is regular and so certainly recursive, and there is nothing to prove. Otherwise, take the enumerator by length and a desired input word w. Wait until either w or some longer word appears, and accept or reject accordingly.

Odd consequence of the previous slide

If a language is recursively enumerable but not recursive, then:

- it can be enumerated
- but it cannot be enumerated by length.

So these languages are not very intuitive to us.

Universal Turing machines (UTMs)

- It's possible to define a representation, $R(M)$, of any Turing machine M where $R(M)$ is a string (see example in Notes 12).
- It's possible to construct a universal Turing machine, U, that works as follows on input of the form $R(M)\#w$
  - w is copied to some working tape
  - Thereafter, observing only that tape, it looks like M is running on w
  - If M halts on w, then so does U on $R(M)\#w$. Also, U accepts if M does.
  - See Notes 12 for (a little) more information.
- On other inputs, we don't care what U does (except it should never halt and accept).
- A UTM can simulate all TMs.
- A UTM is an abstract model of a programmable computer.

The language HALT

- The halting language, HALT, consists of all (and only) the strings $R(M)\#w$ where M is a Turing machine and M halts on w.
- This is exactly the language accepted by U
- Therefore, it is recursively enumerable.
- Is it recursive (i.e. decidable)?