Decidable languages

Atri Rudra

May 26

Announcements

- Handouts
  - Sample final
  - List of topics for the finals
  - H/W #8
    - Remember your lowest H/W grade will be dropped
- Turn in your H/W #7
- Pick up graded H/W #6 at end of class

Please remember…

- I want to show you the "cool" stuff
  - There are problems that are "unsolvable"

Today's puzzle

- Show that the following language is decidable
- $\{\langle G \rangle \mid G \text{ is a CFG and } 1^* \subseteq L(G)\}$

Last lecture

- Three things that a TM can do on an input
  - Halt and accept 
  - Halt and reject 
  - Loop 
- A TM is a decider if it halts on all inputs

A couple of classes of languages

- L is Turing-recognizable
  - Exists a TM that accepts exactly the strings in L
- L is decidable
  - Exists a decider that accepts exactly the strings in L

A_{DFA} is decidable

- A_{DFA} = { ⟨B,w⟩ | B is a DFA and B accepts w }
- M =" On input ⟨B,w⟩
  - Check if B is indeed a DFA
  - Simulate B on w
  - If the simulation leads B to a final state then accept else reject."

This is a decider as the simulation always terminates

Questions ?

You have a choice

The choices are…

Will spend more time on decidable languages next week

- Red pill
  - Go through some more decidable languages quickly
  - Spend most time on diagonalization
- Blue pill
  - Spend more time on some decidable languages
  - Do as much of diagonalization as possible

Both topics are in the homework and no class on Monday

If you chose the red pill

- E_{DFA} = { ⟨A⟩ | A is a DFA and L(A)=∅ }
- E_{DFA} is decidable.
  - Construct a decider
- T = " On input ⟨A⟩, where A is a DFA
  - Mark start state of A
  - Repeat until no new state gets marked
    - p → q If p is marked and q is not, mark q
  - If no final state is marked accept else reject."

Checking if a final state is reachable from the start state

Another example

- EQ_{DFA}={ ⟨A,B⟩| A and B are DFAs and L(A)=L(B) }
- EQ_{DFA} is decidable
- F = "On input ⟨A,B⟩, where A and B are DFAs
  - Minimize A and B to A' and B'
  - Check if A' and B' are isomorphic
    - If they are then accept
    - Else reject."

Alternate proof in Sipser: uses E_{DFA} as a sub-routine