Name: _____

Exercise C1, 3/15/05

Part 1: Chomsky Hierarchy for Your Problems

The Chomsky hierarchy can help you identify the minimal mechanisms (grammars/automata) that are appropriate for your decidable problem. For example, if you know that parenthesis matching can be specified by a CFG and processed by a PDA, you would use these instead of more powerful and thus more costly unrestricted grammar and TM. At the same time, you know that you cannot use a regular expression or a FSA to do the job. We will be studying the two classes "context-free" and "regular" more in detail in the coming units. In this exercise, you will revisit your own problems, where you may want to review what you learned in other courses (most notably "compilers") and develop some intuition behind the hierarchy.

Task: Revisit the problems (including the subproblems) you discussed in Exercise B6/C0. Wherever possible, classify the problems based on the extended Chomsky hierarchy (i.e., TM-recognizable, decidable, context-free, or regular).

Part 2: Grammar and TM Variants

As we discussed in Module B, all nontrivial properties of a TM are undecidable. That is, we cannot have a general mechanism to decide on them. On the other hand, simpler mechanisms are in general much more manageable. So, in many cases, it makes sense to consider and use simpler mechanisms where appropriate. In this part, you *speculate* the power of some downgraded versions of grammars and TMs. If you have questions, contact the instructor.

Task: For each of the following TM/grammar variants, try to find out what class of languages (from the extended Chomsky hierarchy) can be accepted/recognized/generated.

A. CFGs without empty productions (note that CFGs can have rules of the form $A \rightarrow \varepsilon$, where ε is the empty string)
B. CFGs with rules such that $|RHS| \leq 2$ (e.g., $A \rightarrow B\ C$ is acceptable, but $A \rightarrow B\ C\ D$ is not)
C. TMs with 2 stacks along with an additional input tape (i.e., like PDAs except that they are equipped with 2 independent stacks)
D. TMs with a queue along with an additional input tape (i.e., like PDAs except that the tape is used as a queue, not a stack)

Hints

- For the first two problems, try to transform an arbitrary CFG to the limited versions. If you can do that, the limited versions can still accept all CFLs. If not, they may be less powerful.
- For Problem C, see whether you can access an arbitrary element in one stack using the second stack (if so, the 2 stacks can simulate a single infinite tape). For Problem D, see whether the queue can simulate a stack or an infinite tape.
- You are not expected to use references. However, if you do, cite them.

Survey: Time spent between classes: _____

Unit C1 selected slides for the in-class Group Exercise 1

Grammar Hierarchy

- Type 0: Unrestricted grammar
- Type 1: Context-Sensitive Grammar (CSG)
  - $|\text{LHS}| \leq |\text{RHS}|$ [i.e., no shrinking]
- Type 2: Context-Free Grammar (CFG)
  - $|\text{LHS}| = 1$ [i.e., ignore the context of rewriting]
- Type 3: Regular Grammar ~ Regular Expression
  - RHS has at most one nonterminal at the left/right edge.

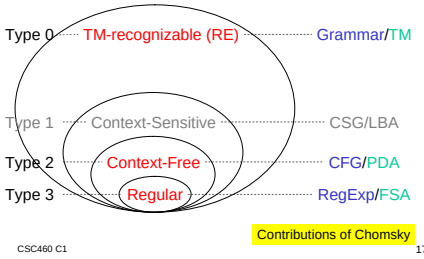
Automata Hierarchy

- Type 0: TM
- Type 1: Linear Bounded Automaton (LBA)
  - Tape space limited to the input size
- Type 2: Pushdown Automaton (PDA)
  - Tape as a stack (with a separate input tape)
- Type 3: Finite-State Automaton (FSA)
  - Head movement only to the right

Language Hierarchy

- Type n languages are accepted/generated by Type n grammars and recognized by Type n automata.
- Type 0: RE (TM-recognizable)
- Type 1: Context-Sensitive Languages (CSLs)
- Type 2: Context-Free Languages (CFLs)
- Type 3: Regular Languages

Chomsky Hierarchy



// End