Problem Set 5

CS373 - Spring 2012

Due: Thursday April 26 at 2:00 PM in class (151 Everitt Lab)

Please follow the homework format guidelines posted on the class web page:

`http://www.cs.uiuc.edu/class/sp12/cs373/`

1. Decidable, Recognizable, or Neither?
   [**Category**: Proof, **Points**: 40]

   Prove whether the following languages are Turing decidable, recognizable (but undecidable), or unrecognizable. Be sure to explain any TM construction that you provide. You can use the examples in Sipser sections 4.1, 4.2, 5.1, and 5.3 in your reductions.

   (a) $A = \{\langle G\rangle \mid G \text{ is a CFG and } L(G) \neq \Sigma^*\}$ (10 Points)

   (b) $B = \{\langle G, H\rangle \mid G \text{ and } H \text{ are CFGs and } L(G) = L(H)\}$ (10 Points)

   (c) $C = \{\langle M, n\rangle \mid n \in \mathbb{N} \text{ and } M \text{ is a TM that does not accept any string in less than } n \text{ steps}\}$ (10 Points)

   (d) $D = \{\langle M\rangle \mid M \text{ is a TM that halts on some input string}\}$ (10 Points)

   (*Remember*: To prove that a language is recognizable but not decidable, you need to prove both that it is undecidable and that it is recognizable.)

2. False Reduction Proof
   [**Category**: Analysis, **Points**: 10]

   Consider the language

   $$\text{REGULAR} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) \text{ is regular}\}$$

   Explain why the following reduction is incorrect and provide a corrected version.

   Claim: REGULAR is not Turing recognizable.
   Proof: We provide a reduction from $\overline{A_{TM}}$ to REGULAR. Assume there is a TM M_R that recognizes REGULAR. Then the following TM recognizes $\overline{A_{TM}}$:

```
$M_{CATM}$
  On input $\langle M, w\rangle$:
  Construct a TM $N$:

  $N$
      On input $x$:
      if $x$ is of the form $0^n1^n$ then accept $x$
      Simulate $M$ on input $w$
      if $M$ accepts $w$ then accept $x$
      else reject $x$

  Simulate $M_R$ on input $\langle N\rangle$
  if $M_R$ accepts then accept $\langle M, w\rangle$
  else reject $\langle M, w\rangle$
```

3. Read Two and Append Process
[**Category**: Comprehension, **Points**: 12]

Consider a new type of model called the Read Two and Append Process (RTAP) that computes by continually modifying the input string. An RTAP *reads and removes* the first two symbols of the current string, and then *appends* some string to the end depending on the symbols read. The RTAP accepts whenever the length of the current string is less than 2.

Formally, an RTAP is a triple $R = (\Sigma, \Gamma, f)$ where Σ is the input alphabet, Γ is the legal symbol alphabet ($\Sigma \subseteq \Gamma$), and $f\colon \Gamma \times \Gamma \to \Gamma^*$ is the function that maps prefix symbols to new suffix strings. R operates as follows:

(1) The current string $w \in \Gamma^*$ is initialized to the input.

(2) If $|w| < 2$, accept the input.

(3) If $w = w_1 w_2 w_3 w_4 ... w_n$ for $w_i \in \Gamma$, w becomes updated to $w_3 w_4 ... w_n f(w_1, w_2)$.

(4) Repeat from Step (2).

The language of R is the set of all strings that accept as input to R. Strings that do not accept will "run forever".

Prove that RTAPs can accept all Turing recognizable languages over $\Sigma = \{0, 1\}$ that contain all strings with length less than 2.

4. Unary Unrecognizability
[**Category**: Proof, **Points**: 10]

Prove that there is an unrecognizable unary language. (*Hint*: Think about conversions between alphabets.)

5. Infinite Unrecognizibility
[**Category**: Proof, **Points**: 12]

Prove that the following language is unrecognizable by a mapping reduction from $\overline{A_{TM}}$:

$$N = \{\langle M \rangle \mid M \text{ is a TM and } M \text{ accepts an infinite number of strings}\}$$

6. Covering
[**Category**: Algorithm, **Points**: 16]

Let S be a nonempty set and let $S_1, S_2, \ldots, S_m$ be a finite sequence of nonempty subsets of S such that $S = \bigcup_{1 \le i \le m} S_i$. Define a *cover* of S to be a set of natural numbers $C \subseteq \{1, 2, \ldots, m\}$ such that $S = \bigcup_{i \in C} S_i$. That is, covers are collections of the S_i that contain all of the elements of S. Notice that by definition, the set of all S_i ($C = \{1, 2, \ldots, m\}$) is a cover of S.

(a) Consider the following problem: "Given a nonempty set S and subsets $S_1, S_2, \ldots, S_m$, is there a cover of S of size 2?" Describe an algorithm that gives the answer given S and $S_1, S_2, \ldots, S_m$. (4 Points)

(b) Argue that the previous problem is in P. (4 Points)

(c) Consider the following problem: "Given a nonempty set S, subsets $S_1, S_2, \ldots, S_m$, and a number $k \in \mathbb{N}$, is there a cover of S of size k or less?" Describe an algorithm that gives the answer given S, the subsets $S_1, S_2, \ldots, S_m$, and k. (4 Points)

(d) Argue that the previous problem is in NP. (4 Points)