3004
Homework #1
Due 21 November 2014 (Friday)

Mark Herbster
m.herbster@cs.ucl.ac.uk

This piece of coursework is worth 2.5% of the marks for the course, and is GRADED. The dead-line for handing it in is 12.00 on Friday 21 November. It should be handed in to either J. Giwa or P. Fenoy at the reception desk. Please make sure you include your full name, your course and year etc on the work you submit.

To encourage collaboration this coursework may be done in groups of four only a single solution set need be submitted from a group of four; however, the front page must clearly indicate the 1-4 participating students. Each participating student must sign a plagiarism statement. *Presentation is important and will be taken into account when marking*.

Preliminaries: Let $\Sigma = \{0, 1\}$, unless stated otherwise for a given problem. Let $LA(M)$ denote the language a TM M semidecides. Let code() be a total recursive 1-1 function that encodes either turing machines or input strings, into strings from $\{0, 1\}^*$. Denote the set of all positive integers by $\mathbb{N}^+ = \{1, 2, 3, \ldots\}$

1 Warmup [20 Points]

1. What is Σ^*?
2. What is the cardinality of Σ^*? Justify your answer.
3. What is a language? Give an example.
4. What is $P(\Sigma^*)$ also known as S_Σ?
5. Is the union and intersection of two r.e. languages also r.e.? Why?
6. **[10 Points]:** Define $f(x, y) := x \dot{-} y$; where

$$x \dot{-} y := \begin{cases} x - y & x \geq y \\ 0 & x < y \end{cases} .$$

   For the following two parts give programs to compute $f(x, y)$ and *provide detailed comments* about your programs. Assume that x and y are nonnegative integers.

   (a) Design a TM.
   (b) Design a URM.

2 Intermediate [60 Points]

1. Are there any non-recursive finite languages?

   Which of the two possibilities holds,

   (a) Yes.
   (b) No.

   Prove (or sketch in detail) which justifies your answer.

2. What is the cardinality of the set of all **finite** languages over the alphabet $\Sigma = \{0, 1\}$?

   (a) This set is countable
   (b) This set is not countable.

   Give an argument which justifies your answer.

3. The "not halting" language is defined to be,

$$L_{\text{nhp}} = \{\text{code}(M)\text{code}(I) : \text{Turing machine } M \text{ does not halt on input } I\}$$

   Prove that the language L_{nhp} is not r.e.

4. Given the decision problem,

   "Given a TM M does the language that it semidecides contain all strings of length less than 5"

   Consider the corresponding language L, which is

$$L = \{\text{code}(M) : \text{LA}(M) \supseteq \{\epsilon, 0, 1, 00, \ldots, 1111\}\}$$

   Which of the two possibilities holds?

   (a) L is recursive.
   (b) L is recursively enumerable but not recursive.

   Give a proof which justifies your answer.

5. **[Problem originally posed by Sean Holden]:** Provide a discussion, of the following question. In particular, provide a specific answer to the question, and explain why you have reached your conclusion.

   We are interested in a particular, informally stated decision problem, and we have formalised it by introducing an alphabet Σ and an encoding scheme, such that we obtain a corresponding language L. The encoding scheme possesses all the required properties. Also, we have demonstrated that L is not recursive, and so we conclude that the problem is not solvable. Is it possible, by using a different encoding scheme (and possibly a different alphabet) to make the problem solvable?

   Give a proof or an example that justifies your answer.

6. Consider a grammar $\mathbb{G}$ consisting of a set of rules of the following type $x_1x_2\ldots x_m \rightarrow y_1y_2\ldots y_n$, and a start string $w_1w_2\ldots w_p$, here x_i, y_i and w_i are characters from an alphabet Σ which includes a distinguished character ϵ which represents the null character and only appears on the right hand side of a rule. For example, consider the following grammar $\mathbb{G}$ with four rules and start string s

$$\mathbb{G} = (\{s \rightarrow ab, db \rightarrow \epsilon, a \rightarrow dd, dd \rightarrow b\}; s)$$

   generates the strings $L(\mathbb{G}) = \{ab, bb, ddb, d\}$, the strings generated by a grammar are those which may be generated by one or more (recursive) applications of the rules (thus unless generated by the application of a rule the start string s is *not* in $L(\mathbb{G})$. A given grammar may generate an infinite number of strings for example

$$\mathbb{G}(\{a \rightarrow aaa\}; a)$$

   generates the set of the strings $L(\mathbb{G}) = \{aaa, aaaaa, aaaaaa\ldots\}$.

   (a) Give a set a of rules and a start string which will generate all the strings which contain an equal number of a's and b's (and only those strings) i.e.,

$$\{\epsilon, ab, ba, aabb, abab, abba, baab, baba, bbaa, aaabbb, \ldots\}$$

   (b) Prove (sketch in detail) that the question, "Given a grammar $\mathbb{G}$, is $L(\mathbb{G}) = \emptyset$?" is *solvable* in general.
   (c) Prove (sketch in detail) that the question, "Given a grammar $\mathbb{G}$ and a string x, is $x \in L(\mathbb{G})$?" is *unsolvable* in general.

3 Advanced [20 Points]

1. "Given a TM M does the language that it semidecides contain at least one string?"

   Consider a corresponding language L, which is

   $$L = \{\text{code}(M) : \text{LA}(M) \text{ contains one or more strings}\}$$

   Indicate one of three possibilities,

   (a) L is recursive.
   (b) L is recursively enumerable but not recursive.
   (c) L is not recursively enumerable.

   Give a proof which justifies your answer.

2. Given the decision problem,

   "Given a TM M does it halt for every input I within 100 steps"

   Consider a corresponding language L, which is

   $$L = \{\text{code}(M) : \text{for every input } I \text{ TM } M \text{ halts within 100 steps when run on } I\}$$

   Choose one of three possibilities (circle one)

   (a) L is recursive.
   (b) L is recursively enumerable but not recursive.
   (c) L is not recursively enumerable.

   Give a proof which justifies your answer.

3. Given the decision problem,

   "Given a TM M does there exist an input I such that M halts with within length(I) steps"

   Consider a corresponding language L, which is

   $$L = \{\text{code}(M) : \text{there exists an input } I \text{ such that TM } M \text{ halts within length}(I) \text{ steps}\}$$

   Indicate one of three possibilities,

   (a) L is recursive.
   (b) L is recursively enumerable but not recursive.
   (c) L is not recursively enumerable.

   Give a proof which justifies your answer.

4. Given the decision problem,

   "Given a TM M does it halt on all inputs"

   Consider a corresponding language L, which is

$$L = \{\text{code}(M) : LA(M) = \{0, 1\}^*\}$$

   Indicate one of three possibilities,

   (a) L is recursive.
   (b) L is recursively enumerable but not recursive.
   (c) L is not recursively enumerable.

   Give a proof which justifies your answer.