Instructions

- Classroom Problems C10.1–C10.3 will be discussed and solved onsite at the tutorial sessions in lecture week 10. No credit is given for these problems.
- Homework Problems H10.1–H10.3 you should solve on your own, and be available to present your solutions at one of the tutorial sessions in lecture week 11. In order to get course credit, you need to indicate your solved problems on the signup sheet circulated at the beginning of the session.
- Supplementary Problems S10.1–S10.2 provide further illustration and extension of the course material, but will usually not be covered at the tutorials. You are however invited to work on these problems too, and discuss them with the course staff. Sample solutions are provided on MyCourses.

Classroom Problems

C10.1 Prove that the following problem is undecidable:

> Given a Turing machine M, an input string x and a tape symbol s, does M ever write the symbol s on its tape when it is run with the input x?

C10.2 Prove that the following problems are undecidable:

(a) Given a Turing machine M with input alphabet Σ, does it accept all input strings, i.e. is it the case that $\mathcal{L}(M) = \Sigma^*$?

(b) Given a Turing machine M with input alphabet Σ, does it reject all input strings, i.e. is it the case that $\mathcal{L}(M) = \emptyset$?

C10.3 You are being offered the following programming assignment:

> *Intel Ice 9 code optimisation*
> A large company producing embedded systems software would like to have a code optimiser that, given as input any machine language program for the new Intel Ice 9 processor, will output the smallest machine language program that is functionally equivalent to the given one (i.e. has the same input-output behaviour).

Your (justified) comments?

Homework Problems

H10.1 Division by zero is a common error in computer programs. Naturally, it would be very good to make sure that our software is free of such errors before it is published. Thus, let us consider the following decision problem DIV-BY-ZERO:

> Given a computer program P (in your favourite programming language) that can use an unbounded amount of memory. Is there any input that causes the program to perform division by zero?

Explain why this problem is undecidable. (Assuming that an unbounded amount of memory is really accessible, i.e. pointers are not restricted to some fixed bitlength etc.)

Hint: Show that if the problem were decidable, then so would be the Halting Problem for Turing machines. That is, describe how one can transform a pair $\langle M, x \rangle$, where M is a Turing machine and x is an input string, into a computer program P such that M halts on x if and only if P performs division-by-zero on some input. You don't have to give full details of the transformation, a high-level description is enough.

H10.2 Prove that the following problem is undecidable:

> Given a Turing machine M, does it accept the empty input string ε?

Hint: Consider the proof of Lemma 10.5 or the examples following Corollary 10.9 on the lecture slides.

H10.3 Consider the following languages. For each language: (i) give a string that belongs to the language, (ii) give a string that does not belong to the language, (iii) determine whether the language is decidable or not. Justify your answers.

(a) $\{x \in \{0, 1, ..., 9\}^* \mid x$ is (a decimal representation of) a prime number$\}$

(b) $\{\langle l, m \rangle \mid l$ is a list of rational numbers whose average is less than the rational number $m\}$

(c) $\{p \mid p$ is a polynomial on n variables that has integer zeros (roots) for some values of its variables in the domain $[-1000000, 1000000]^n\}$

(d) $\{\langle G, x \rangle \mid G$ is a context-free grammar that generates string $x\}$

(e) $\{\langle G, s, t \rangle \mid G$ is a finite undirected graph that has a path from vertex s to vertex $t\}$

(f) $\{\langle G, k \rangle \mid G$ is a finite undirected graph that contains a clique of k or more vertices$\}$[1]

[1] `http://en.wikipedia.org/wiki/Clique_(graph_theory)`

(g) $\{P \mid P$ is a Python program that does not contain any division operators$\}$

(h) $\{P \mid P$ is a Python program whose execution halts eventually when it is run without any input and has unbounded amount of memory available$\}$

Note: For items (b)–(h), where no specific alphabet or representation of the language elements as strings is given, you may choose any reasonable encoding. For instance, Python programs can be viewed as strings over the Unicode alphabet. Similarly, polynomials can be expressed in the ASCII alphabet so that for instance the polynomial $7x_1x_2^3 + 1.5x_1^7$ is expressed as the string `7*x1*x2^3+1.5*x1^7`.

Supplementary Problems

S10.1 Prove the following connections between computable functions and languages:

(a) A language $A \subseteq \Sigma^*$ is decidable, if and only if its characteristic function

$$\chi_A : \Sigma^* \to \{0,1\}, \qquad \chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

is a computable function.

(b) A language $A \subseteq \Sigma^*$ is semi-decidable, if and only if either $A = \emptyset$ or there exists a computable function $g : \{0,1\}^* \to \Sigma^*$ such that

$$A = \{g(x) \mid x \in \{0,1\}^*\}.$$

Note: This is the characterisation underlying the historic term "recursively enumerable language": a language A is r.e. if and only if its elements can be "enumerated" by a recursive (= computable) function g.

S10.2

(a) Prove that any decision problem that has only finitely many possible inputs is decidable.

(b) Prove that the problem "Does the decimal expansion of π contain 1000 consecutive zeros?" is decidable. What does this result tell you about (i) the decimal expansion of π, (ii) the notions of decidability and undecidability?