
Computing Theory
COSC 1107/1105
Assignment 2: Computability

Assessment Type	Individual assignment. Submit online via Canvas → Assignments → Assignment 2. Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements/relevant discussion forums.
Due Date	Tuesday 19th October 2021, 11:59pm (note new date)
Marks	100 worth 25% of your final assessment

1 Overview

This assignment requires you to demonstrate your knowledge of some key issues in computability, and of non-determinism. There is also a part which deals with the Platypus game.

2 Assessment details

1. Passwords (8 marks)

The Dwarves of the Lonely Mountain have a lot of time on their hands, and are very worried about information about the size of their horde of gold getting around various parts of Middle-Earth, as they fear invasion by all and sundry if any hint of their true wealth were to be revealed. Accordingly King Durin XIII decrees that all records are to be written only in encrypted Khuzdul, in which there are 32 distinct characters. The encryption process is based on a Khuzdul keyword of n characters in length. Durin's advisors inform him that an evil wizard with an army of orc-slaves at his disposal (or a few of the lesser intelligent hobbits :-)) could exhaustively search through all possible keywords at a rate of 100,000 keywords per day. Durin decrees that the royal Khuzdul keyword must be secure "until the end of the age".

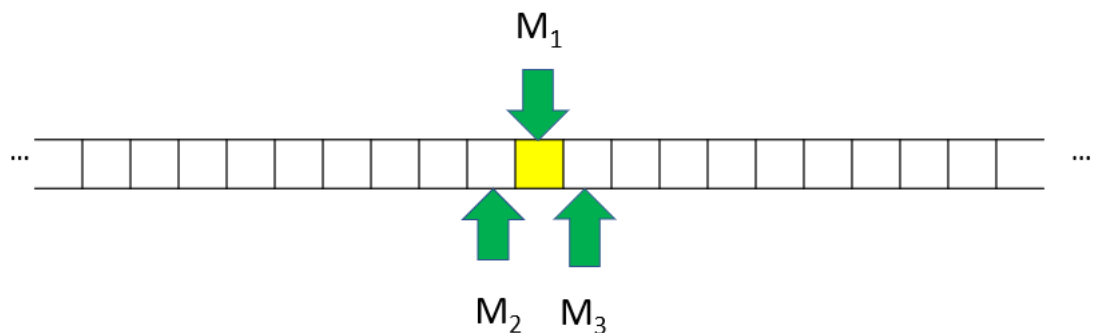
- (a) Determine an appropriate interpretation of Durin's statement "until the end of the age". In other words, define what you think this means and hence what the maximum such time would be. (2 marks)
- (b) Given your previous answer, how long should the royal keyword be? Explain your answer. (2 marks)
- (c) Sillibo, a passing hobbit, happens to point out to Durin that not all of the 32 Khuzdul characters are equally likely to be used, as there are 18 that are only used very rarely. Hence an outsider might concentrate on the other 14 and make significant progress. Assuming Sillibo is correct, calculate how long it would take an evil wizard to discover a Khuzdul keyword of length given in

your previous answer, assuming that at most 14 different Khuzdul characters are used in the keyword. (2 marks)

- (d) Sillibo also tells Durin, who is very impressed with the hobbit's knowledge, that the royal keyword should not include a Khuzdul translation of the name "Arkenpebble" (a famous jewel revered by all dwarves), as this is very well known to all in Middle-Earth, and hence will certainly be known to any evil wizard. Given the translation of Arkenpebble into Khuzdul uses 6 characters, evaluate Sillibo's claim. In other words, is the royal keyword still as secure as Durin would like if it does contain the translation of Arkenpebble and the evil wizard correctly guesses this. (2 marks)

2. Computability (14 marks)

The generalised 3-player Platypus game is defined as follows. Let M_1 , M_2 , and M_3 be Turing machines, which share the same tape. The tape is initially blank. The initial configuration of the three machines is as shown below.



As in the Platypus game, each machine takes turns to move (but there is no scoring involved).

- (a) Show that the halting problem for the 3-player generalised Platypus game is undecidable. You may use any reduction you like. (6 marks)
- (b) Suppose the 3-player generalised Platypus game is played on a Turing machine with a finite tape (making the halting problem decidable), and that this problem has been shown to be NP-complete. Given your above reduction from some problem A to the 3-player generalised Platypus game, can this reduction be used to conclude that A is NP-complete? Why or why not? Explain your answer. (2 marks)
- (c) Consider the two Turing machines M_1 and M_2 below.

M_1 : Whatever the input is, M_1 overwrites each character in the input, resulting in a totally blank tape. M_1 then terminates.

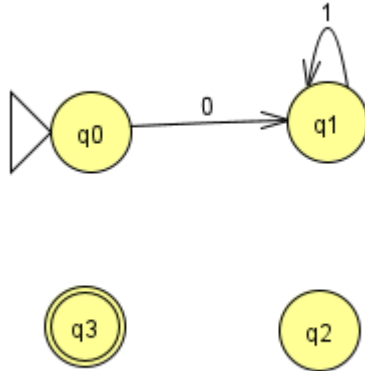
M_2 : Given an input string w , M_2 outputs another Turing machine M_w which will take a blank tape as input, print w on the tape and then terminate.

Explain how these two Turing machines are used in at least three reductions of the Halting problem to other undecidable problems. (6 marks)

3. Nondeterminism

(8 marks)

Consider the incomplete NFA M_0 below, whose alphabet is $\{0, 1\}$.



Use M_0 to create three more NFAs M_1 , M_2 and M_3 according to the constraints below. Explain in one or two English sentences how you constructed each NFA.

- Each of M_1 , M_2 and M_3 must contain at least 10 transitions (potentially but not necessarily including λ -transitions) and must be an NFA but not a DFA. Specifically there must be at least one combination of state and input (either 0 or 1) for which there are at least two possible states. Put another way, removing all λ -transitions must not result in a DFA.
 - Use JFLAP to transform M_1 , M_2 and M_3 into equivalent DFAs.
 - The sizes of the DFA resulting from the determinising algorithm must be as below. Note that the JFLAP implementation of this often omits an “error” state, i.e. it may be necessary to add an extra state to the result from JFLAP in order to account for this. The size constraints below assume a fully deterministic DFA; one way to check for this is that if the DFA has k states, there must be exactly $2k$ transitions (one for each of 0 and 1 in each state).
- (a) The size of the DFA corresponding to M_1 is 2. (3 marks)
- (b) The size of the DFA corresponding to M_2 is 5 (2 marks)
- (c) The size of the DFA corresponding to M_3 is at least 9 (this may be harder than you think!) (3 marks)

4. Pumping Lemma

(20 marks)

- (a) There are three errors in the statement of the Pumping Lemma for regular languages below. Find all three, and state how to correct them. Explain each of your corrections. (4 marks)

Let L be a regular language. Then $\exists n \geq 1$ such that for some $w \in L$ such that $|w| \geq n$, $\forall x, y, z$ such that $w = xyz$ where

- i. $|xy| < n + 1$
- ii. $y \neq \lambda$
- iii. $xy^iz \in L$ for all $i \geq 1$

- (b) One fine day, as soon as lockdown was over, Elladan and Elrohir talk an afternoon walk in the gardens of Lothlorien. While contemplating the meaning of existence underneath the massive ancient trees, they come across ten pieces of parchment, which appear to have been hastily ripped up during the recent War of the Ring, on which are written some strange runes that they cannot read. Desperate to find out what these mean, they send the fragments to Imladris for analysis, where the wisest still in Middle Earth would be found. After several anxious weeks, a passing eagle delivers a package to them. Inside they find a letter from Elrond himself, explaining that despite their best efforts, they were unable to fully translate the runes, and so there are some places where there are several possible translations of particular phrases. Elladan and Elrohir therefore have two tasks. The first one is to arrange the fragments in the most likely order. Then they have to select from each of the possible translations so that the overall message makes sense.

The fragments are below, with the parts where there are alternative translations labelled with capital letters, like this:

(Translated text) A (Translated text)

A1: Alternative 1

A2: Alternative 2

A3: Alternative 3

1: Let w be A

A1: $0^n 2^n 1210^n 2^n$. A2: $2^n 1212^n$. A3: $0^n 1210^n$. A4: $1^n 1211^n$.

2: and so $xy^i z$ is B

B1: $2^n 0^{n+j} 1212^n 0^n$ B2: $2^{n+j} 1212^n$ B3: $0^{n+j} 1210^n$ B4: $1^{n+j} 1211^n$

3: Then $\exists n \geq 1$ such that C and D,

C1: for all $w \in L$ and $|w| \geq n$ C2: for some $w \in L$ and $|w| \geq n$ C3: for all $w \in L$ and $|w| \leq n$ C4: for some $w \in L$ and $|w| \leq n$

D1: $\exists x, y, z$ such that $w = xyz, |xy| \leq n$ D2: $\forall x, y, z$ such that $w = xyz, |xy| \leq n$ D3: $\exists x, y, z$ such that $w = xyz, |xy| \geq n$ D4: $\forall x, y, z$ such that $w = xyz, |xy| \geq n$

4: Finrod's Tale: A proof that the language $L = E$ is F.

E1: $\{w121|w \in \{0, 1, 2\}^*\}$ E2: $\{121w|w \in \{0, 1, 2\}^*\}$ E3: $\{w000w|w \in \{0, 1, 2\}^*\}$ E4: $\{w121w|w \in \{0, 1, 2\}^*\}$

F1: regular F2: context-free F3: not regular F4: empty

5: As the Pumping Lemma requires $xy^i z \in L$, this is a G

G1: problem G2: paradox G3: contradiction G4: tautology

6: and so we have shown that our assumption is false, i.e. that L is H.

H1: regular H2: context-free H3: not regular H4: empty

7: Assume L is I.

I1: regular I2: context-free I3: not regular I4: empty

8: Now consider J

J1: $i = 0$ J2: $i = 1$ J3: $i = 2$ J4: $i = 3$

9: so we have $w \in L$, K and $|xy| \leq n$, and so we have L for some $1 \leq j \leq n$.

K1: $|w| \leq n$ K2: $|w| < n$ K3: $|w| \geq n$ K4: $|w| > n$

L1: $y = 0^j$ L2: $y = 1^j$ L3: $y = (02)^j$ L4: $y = (12)^j$

10: $y \neq \lambda$ and M

M1: $xy^iz \notin L$ for all $i \geq 0$ M2: $xy^iz \notin L$ for some $i \geq 0$ M3: $xy^iz \in L$ for all $i \geq 0$ M4: $xy^iz \in L$ for some $i \geq 0$

You do not need to write the proof out in full. Just indicate the fragment order and your choice for each of the letters in a sequence like the one below.

9K2 8J3 4E4 ...

(6 marks)

- (c) Let L be any regular language and let M be a DFA for it with n states. Explain how you can use the Pumping Lemma to show that L is infinite iff there is a string $w \in L$ such that $n \leq |w| \leq 2n - 1$. (4 marks)
- (d) Let L be any regular language over $\{a, b, c\}$. Show how the Pumping Lemma can be used to demonstrate that in order to determine whether or not L is empty, we need only test at most $(3^n - 1)/2$ strings. (2 marks)
- (e) The Pumping Lemma for context-free languages is below.

Let L be a context-free language. Then there is an $n \geq 1$ such that for any string $w \in L$ with $|w| \geq n$ there exists strings x, y, z, u, v such that $w = xyzuv$ and

- i. $|yzu| \leq n$
- ii. $|y| + |u| > 0$
- iii. $xy^izu^iv \in L$ for all $i \geq 0$

Use this to show that the language $L = \{a^{i^2+i} \mid i \geq 0\}$ is not context-free by filling in the gaps below. (4 marks)

Proof: Assume _____. So the Pumping Lemma applies, and so for any string $w \in L$ with $|w| \geq n$ there exist strings x, y, z, u, v such that $w = xyzuv$ and

- i. $|yzu| \leq n$
- ii. $|y| + |u| > 0$
- iii. $xy^izu^iv \in L$ for all $i \geq 0$

Let _____. So $w \in L$ and $|w| \geq n$, and $w = xyzuv$, $|yzu| \leq n$, $|y| + |u| > 0$ and $xy^izu^iv \in L$ for all $i \geq 0$. Now as _____, this means $|yzu| = |y| + |z| + |u| \leq n$ and so $|y| + |u| \leq n$.

Let $i = 0$ and consider $|xy^0zu^0v| = |xzu| = ______ - |y| - |u| = n^2 + n - (|y| + |u|) \geq n^2 + n - n > ______ = (n - 1)^2 + (n - 1)$. So $n^2 + n = ______ > |xy^0zu^0v| > (n - 1)^2 + (n - 1)$, and so $xy^0zu^0v \notin L$. This is a contradiction, and so L is not context-free.

5. Intractable problems

(10 marks)

Intractable problems are decidable problems, but for which the best known solution is exponential (or worse). Describe two intractable problems and their practical

application. You should write one introductory paragraph on intractable problems, and two further paragraphs, one on each problem, and a reason that you selected each one. Some suggestions will be given in class and on Canvas.

Please note that I am not at all interested in what you can find on Google or Wikipedia or anything like that. What I really want to see is some evidence of you thinking about intractable problems and what effect these have, such as the relationship between the vertex cover problem and sensor networks.

Perhaps an interesting way to address this is to consider what differences it would make if your intractable problem could be solved efficiently. For example, if we could solve say the Travelling Salesperson problem in $O(n^n)$ time, what would be possible that is impossible now?

Another possibility is to consider how intractability can be a useful thing, such as keeping information secure in cryptosystems or in related applications such as blockchain.

Whatever problems you choose, please avoid the temptation to 'cut, copy and edit'; as soon as you do that, you have done something wrong. I would far prefer to read your own words and your own perspective on these problems.

6. Universality (24 marks)

In a nutshell, you are expected to revise and extend your work on this topic in Assignment 1.

In Assignment 1, you investigated one of the following three topics, or came up with your own related topic or creative story.

- [Two-dimensional Turing machines](#)
- [Small universal Turing machines](#)
- [Notable universal Turing machines](#)

For this assignment, **you are expected to either continue your investigation from Assignment 1 on the same topic in more depth, or to make a different choice.** In other words, you can either continue with your choice from Assignment 1, or make a different one now. Whatever your decision, **you are expected to write about 1800-2000 words (9 or 10 paragraphs) overall.** This should include a revised version of your Assignment 1 submission, so that if you continue with the same choice as in Assignment 1, this is will be an extended form of that work. If you make a different choice, that is fine, but you should include your (potentially revised) Assignment 1 submission as part of this submission. So you have two different choices for the two assignments, you are expected to write about the same length on each; if you have the same choice for each, you should write about 1800-2000 words overall. Either way, **the submission for Assignment 2 will involve around 1000 words over and above what you submitted for Assignment 1.**

As in Assignment 1, you may also propose an alternative topic, or write a **creative story** involving a Turing machine of some kind. You can do this even if you did not choose either of these in Assignment 1. **However, for any alternative topic or creative story, please seek approval from the lecturer before commencing**

work on it. This is to make sure that what you are doing is appropriate for this assignment — I would hate to see an outcome where you do a lot of work, only for it not to count because it does not address the intended content.

One alternative topic of particular interest is **quantum computing**; if anyone is interested in pursuing this, you are strongly encouraged to do so (but as above, please do consult me first). Another possibility is **zero-knowledge proofs**, but again please consult me before doing this.

A further point is that ***you can present your information in other ways that a formal report if you wish.*** Some suggestions are below. Please keep in mind that you still need to discuss the technical content; the point is to find a way that assists you with this, rather than being a blockage for you.

- Pick a side in the debate about the 2007 universal TM competition
- Langton’s Ant vs Paterson’s worm
- ‘Cellular automata are better than Turing machines’
- Write a children’s story, movie scene, poem, ...
- 2D TMs as a game, map, drawing a picture, annotating photos, ...
- Implementation of some aspects (be careful of rabbit holes!)
- Experiment with Java implementation of 2D Universal TM
- Langton’s ant with ‘boundaries’ (see an example [here](#))

You will be marked according to the rubric below.

Points	Description	Details
18-24	Exemplary	You have explored your chosen topic well. Your report is clear and well-written, and has the appropriate length. This is interesting and informative.
12-17	Accomplished	You have explored your chosen topic reasonably well. Your report is generally good, but can be improved with some more attention to detail, particularly concerning the amount of technical detail.
6-11	Developing	Your report needs some further work, either to increase the level of content or to improve the choice of material and its presentation.
0-5	Beginning	Your report does not show evidence of sufficient investigation, and is lacking in detail.

7. The Platypus game. (16 marks)

We have previously talked about running as large a tournament as possible with the Platypus game. The way we will do this in this assignment is for each of you to run a tournament of 2,500 machines. From these, you will report your top 10 machines (see below for details). These 10 will then be part of a knock-out tournament to determine the overall winner.

Before answering the questions below, do the following.

- Get your allocation of 2,500 machines. These will in a OneDrive directory that will be shared with you. Once you have the link, look for a file in the folder with your student number, i.e. if your student number is 7654321, look for the file 7654321.pl. Store this somewhere that suits you with a name like `machines.pl` for convenience.
- Open a new SWI-Prolog shell and consult both `platypus.pl` and `machines.pl` (or whatever you called it). Then run the command below. This will classify your list of machines into three categories, and will generate one file each category named `none.pl`, `reachable.pl` and `unreachable.pl`, as per the description below. This will be in the directory specified by the `results_directory` predicate in `platypus.pl`; you can change this to something more convenient for you if you like.

?- classify_machines.

None The machine has no platypus state in the fourth row of columns 1-6
Reachable It is possible reach the platypus state from the kangaroo state
Unreachable It is not possible reach the platypus state from the kangaroo state

The point of this analysis is to work out whether it is possible for the machine to terminate the game or not. An example of each class of machine is below. In the first case, the machine cannot terminate the game, as it is not possible for it to go from the start state (**kangaroo**) to the **platypus** state as there is no transition that will move the machine into the **platypus** state.

Y	G	Y	G	Y	G	Y
K	K	E	E	W	W	P
g	y	y	g	y	y	g
Emu	Emu	Wombat	Kangaroo	Wombat	Emu	Kangaroo
w	gg	gg	w	w	w	gg

The second and third cases occur when there is indeed such a **platypus** state. When it is possible to move from the **kangaroo** state to the **platypus** state (depending on the cells on the tape of course), then the machine is classified as reachable, as in the machine below. This is because it is possible to move from the **kangaroo** state to the **emu** state (column 1 or 2), from the **emu** state to the **wombat** state (column 3) and from the **wombat** state to the **platypus** state (column 6).

Y	G	Y	G	Y	G	Y
K	K	E	E	W	W	P
g	y	y	g	y	y	g
Emu	Emu	Wombat	Kangaroo	Wombat	Platypus	Kangaroo
w	gg	gg	w	w	w	gg

It is also possible that even if such a **platypus** state exists, it is not possible to move into that state from the **kangaroo** state. In this case, the machine is classified as unreachable, as in the machine below. In this machine we can get from the **kangaroo** state to the **wombat** state and vice-versa, but we can never get to the **emu** state or **platypus** state from either of these.

Y	G	Y	G	Y	G	Y
K	K	E	E	W	W	P
g	y	y	g	y	y	g
Wombat	Wombat	Platypus	Kangaroo	Wombat	Kangaroo	Kangaroo
w	gg	gg	w	w	w	gg

Intuitively, machines classified as reachable are in some sense “genuine” with the others being “imposters”. Accordingly, having separate tournaments for each seems only fair.

- Run a tournament for each of the following classes of machines
 - All 2,500 machines
 - Only machines classified as reachable
 - Only machines classified none or unreachable (ie all those not classified as reachable).

To do this, you will first have to prepare a file containing only the machines as above (the above `classify_machines` command will more or less do this for you). You will also need to use the command below, which will run the tournament with all of the options below.

?- `my_tournament([competition])`.

Variation **Description**

<i>Tree</i>	5 points for whenever either tree is reached
<i>Green</i>	2 points rather than 1 for changing green to yellow
<i>Animal</i>	1 point every time a change of animal occurs
<i>Tiebreaker</i>	A random starting configuration is chosen with game length is 200

- (a) For each of your tournaments, report the overall time taken, the top 10 machines (by 'football' ranking), the overall number of wins and draws, and the number of winless machines. How many machines were classified as none, reachable and unreachable respectively? Report your results in a table as below. (2 marks)

Class	Number	Percentage
Reachable		
Unreachable		
None		

- (b) Re-run your tournament of all machines, but this time you are to include 10 extra machines of your own choice. These should be different from any machines in your list already, but otherwise you are free to choose them however you like.

You can use `platypus.pl` from Canvas ([link here](#)) to assist with this if you wish. This will check whether your 10 added machines are 'legal', and whether or not they were already part of your allocation. To do this, consult `platypus.pl` and `machines.pl` as above, and a third file `extra.pl`. Then run the command

?- `check_new`.

This will then output whether your added machines are legal, and whether they are already part of your allocation or not (and if they are, which ones are already in their allocation).

You should report where each of these 10 machines finish in the 'football' ranking, and your reasons for choosing each of them. (2 marks)

- (c) Were you surprised how your chosen 10 machines performed? What can you conclude from this about high performing Platypus machines? If you had to choose one machine to represent you in a tournament, what machine would it be? Briefly explain your decision. (4 marks)
- (d) In the previous assignment, you calculated the largest Platypus tournament you can play on your machine in 4 hours, ie $4 \times 60 \times 60 = 14,400$ seconds. This is of course for the 'standard' 2-player game. 3-player and 4-player tournaments will of course take longer. Calculate the largest 3- and 4-player tournaments you can play on your machine in 4 hours. You may assume that a 3- or 4-player match takes the same time as a 2-player one. You may also find the following table useful (see the notes on 3- and 4-player tournaments for how these are derived). (2 marks)

Players	Matches required
2	$n(n+1)/2$
3	$n(n+1)(n+2)/6$
4	$n(n+1)(n+2)(n+3)/24$

- (e) If the Platypus tournament were to be run again, what alterations would you recommend? Some ideas are below; you can add others as you see fit. You should have at least three suggestions. (6 marks)
- Once students are allocated their machines, they play a tournament amongst these to find the best 10. These 10 then take on the best 10 from other students. This could include the possibility of students choosing their 10 machines from their allocation or from the set of machines which are not allocated to any student.
 - The initial tape and scoring processes are changed from tournament to tournament. These are announced in advance, and allow choices to be made for which machines will be used.
 - Non-player machines can be added. These are not competitors, but may change the tape in ways that influence the game. Such machines would not be allowed to terminate the game (presumably by allowing them transitions for a green cell with a platypus).
 - When a player has a platypus on a green cell, the game does not halt, but is "rebooted", i.e. the tape reverts to its initial state, the player which 'halted' the game gets a bonus or a penalty, and the machines involved are changed in some way. This change could be swapping rows 3, 4 and 5 of column 1 and 2 (kangaroo), and the same for columns 3 and 4 (emu) and 5 and 6 (wombat). There could be a maximum of say 5 reboots per game.
 - (insert your idea here!)

3 Submission

You should submit a PDF file, and all `tournament.csv` files from your tournaments. Do not use a zip file. Just add attach all relevant files to the submission on Canvas. **No**

other file formats will be accepted.

In addition, for Questions 4b and 4e, you **must** fill in your answers on the Canvas Quiz here. As noted in announcements, this will assist with marking assignments more quickly.

4 Marking guidelines

Your assessment will be marked according to the criteria below.

- Completeness and accuracy of your answers to the first five questions.
- Completion of your section of the Platypus game tournament.
- Quality of your comments on the Platypus game tournament.