Due: Wed 10 April, 23:59

Submission is through give and should consist of:

- 1. A single pdf file with typeset answers, maximum size 4Mb. Prose should be typed, not handwritten. Use of LATEX is strongly encouraged, but not required.
- 2. A file partner.txt with a single line on the format. z<digits> . This must be the zID of your group partner. For individual submissions, write your own zID.

Submit your work using the web interface linked on the course website, or by running the following on a CSE machine

give cs2111 assn2 assn2.pdf partner.txt

This assignment is to be done in pairs. Individual submissions are discouraged, but will be accepted. Only one member of the pair should submit. Work out who will submit ahead of time—duplicate submissions are extremely annoying to sort out.

Late submission is allowed up to 5 days (120 hours) after the deadline. A late penalty of 5% per day will be deducted from your total mark.

Discussion of assignment material with others is permitted, but you may not exchange or view each others' (partial) solutions. The work submitted *must* be your own, in line with the University's plagiarism policy. Note in particular that attempting to pass off AI writing (e.g. ChatGPT output) as your own still counts as plagiarism.

Background In the lectures, we defined the set of traces (aka runs) for unlabelled transition systems. For example, if we ignore the labels on this transition system:



the set of traces starting from q_0 , which we will write $Tr(q_0)$, is the following:

*{q*₀*, q*₀*q*₁*, q*₀*q*₁*q*₂*, q*₀*q*₁*q*₃*, q*₀*q*₁*q*₃*q*₄*}*

But the set of *maximal* traces, which we will write $MTr(q_0)$, is

 $\{q_0q_1q_2, q_0q_1q_3q_4\}$

For labelled transition systems, we are often less interested in what states we visit, and more interested in the labels of the transitions we take. To this end, we define the set of *action traces* starting from a state *s*, written ATr(s), to be the label sequences we can use to label the transitions of Tr(s). So in our example, we would have $ATr(q_0) = \{\lambda, a, ab, ac, ad, ada\}$.

More formally, we define ATr(s) as follows:

$$\mathsf{ATr}(s) = \{ \sigma : \exists \tau \in \mathsf{Tr}(s), |\sigma| = |\tau| - 1 \land \forall i < |\sigma|, \tau_i \xrightarrow{\sigma_i} \tau_{i+1} \}$$

where we use σ , τ to range over (possibly infinite) sequences, and where $|\sigma| = \infty$ if σ is infinite. Similarly, the *maximal action traces* MATr(*s*) are the label sequences we can use to label the transitions of MTr(s). So in the above example we have MATr(q_0) = {ab, ac, ada}.

Problem 1

In the above example, there happened to be more action traces than traces, but that is not always the case. In the following, s_0 refers to the initial state of whatever transition system you come up with.

Give examples of transition systems where the following is true:

(b) $ Tr(s_0) > ATr(s_0) $ (4 matrix)(c) $ MATr(s_0) = ATr(s_0) $ (4 matrix)(d) There are infinitely many states, but $ATr(s_0)$ is finite.(4 matrix)(e) $ATr(s_0)$ is infinite, but there are only finitely many states.(4 matrix)	(a)	$ATr(s_0) $ (4 ma	ırks)
(c) $ MATr(s_0) = ATr(s_0) $ (4 matrix)(d) There are infinitely many states, but $ATr(s_0)$ is finite.(4 matrix)(e) $ATr(s_0)$ is infinite, but there are only finitely many states.(4 matrix)	(b)	ATr(s ₀) (4 ma	ırks)
(d) There are infinitely many states, but $ATr(s_0)$ is finite.(4 matrix)(e) $ATr(s_0)$ is infinite, but there are only finitely many states.(4 matrix)	(c)	$= ATr(s_0) \tag{4 ma}$	ırks)
(e) $ATr(s_0)$ is infinite, but there are only finitely many states. (4 ma	(d)	nfinitely many states, but $ATr(s_0)$ is finite. (4 ma	ırks)
	(e)	nfinite, but there are only finitely many states. (4 ma	ırks)

Problem 2

(40 marks)

Consider the following two labelled transitions M_0 and M_1 , representing coffee machines:



The label \$1 represents the machine receiving a coin, *c* represents the machine serving coffee in response to the coffee button being pressed, and similarly *t* represents the machine serving tea in response to the tea button being pressed.

(a) What is $MATr(q_0)$? What is $MATr(r_0)$?

(6 marks)

Now we want some coffee. Here's a transition system representing a coffee buyer *B*:



We use the same alphabet to represent actions of the customer: \$1 represents inserting a coin, c represents pressing the coffee button and obtaining coffee, and so t would represent pressing the tea button and obtaining tea. ¹

Now that we have models of both the coffee buyer and the coffee machines, we can model how they interact. For that, we need to construct their *synchronous product*.

¹One might argue that there is a difference between being the presser of a button, and being the button that's pressed. For the purposes of this assignment, this distinction won't matter, so we use the same alphabet to denote both.

Definition 1 (Synchronous product) *The* synchronous product $C_0 || C_1$ *of two labelled transition systems* $C_0 = (S, \rightarrow_S, s_0)$ and $C_1 = (T, \rightarrow_T, t_0)$ *is defined as:*

$$C_0 \parallel C_1 = (S \times T, \rightarrow, (s_0, t_0))$$

where \rightarrow is defined as follows for all s, t, α , s', t':

 $(s,t) \xrightarrow{\alpha} (s',t')$ iff $s \xrightarrow{\alpha}_{S} s'$ and $t \xrightarrow{\alpha}_{T} t'$.

In words, the synchronous product allows only the transitions that are possible in both transition systems. If only one side is ready to proceed with an action, it can't proceed. Intuitively, the \$1 action in the transition system $M_0 \parallel B$ represents a successful interaction between machine and buyer: the buyer inserts a coin, and the machine accepts it.

- (b) Draw the state diagrams of $M_0 \parallel B$ and $M_1 \parallel B^2$ Or if you prefer, explicitly enumerate all of their transitions. Feel free to leave out unreachable nodes from the diagram. (6 marks)
- (c) In the transition system $M_0 \parallel B$, what is MATr((q₀, b₀))? In the transition system $M_1 \parallel B$, what is MATr((r₀, b₀))? (6 marks)
- (d) Compare your answers to Questions (a) and (c). Is there a difference between M_0 and M_1 that is evident from one of the answers, but not the other? Why do you think that is? (10 marks)
- (e) Define a safety property and a liveness property that, taken together, imply that our coffee drinker *B* gets appropriate service from the machine. (6 marks)
- (f) Of the systems $M_0 \parallel B$ and $M_1 \parallel B$, which ones satisfy your safety property? Which ones satisfy your liveness property? (6 marks)

Problem 3

Read Section 4.1.4 ("Order of Commands") of RFC 5321³, which specifies how a session between a mail client and a mail server may proceed according to the SMTP protocol. A session is a structured interaction in which one or more parties exchange messages; in the case of SMTP, the purpose of a typical session is for the client to transmit e-mail to the server. The client tells the server what it wants to do by sending a

Define a DFA over the input alphabet

{DATA, EHLO, EXPN, HELP, MAIL, NOOP, RCPT, RSET, VRFY, QUIT}

sequence of protocol-specific commands. For our purposes, you don't necessarily need to understand what

the commands do, so you don't have to read the rest of the RFC. But if you're curious feel free to.

that accepts exactly the admissible sequences of client-issued commands, as specified by Section 4.1.4 of the RFC. Ignore server responses and command parameters. Sometimes the server needs to make a decision about whether our command is acceptable based on the parameters; in such situations, assume the server will find anything we send acceptable.

Explain any design decisions you made in constructing the DFA, and any interpretative liberties you took. In particular, was there any point where you found the text of the RFC to be ambiguous? If so, how did you choose to interpret the ambiguous text?

(40 marks)

²State diagrams are the graph representation of a transition system used in the lectures and in this assignment description ³https://tools.ietf.org/html/rfc5321#section-4.1.4