INF210: Modelling of Computing
Oblig 1

Åsmund Aqissiaq Arild Kløvstad

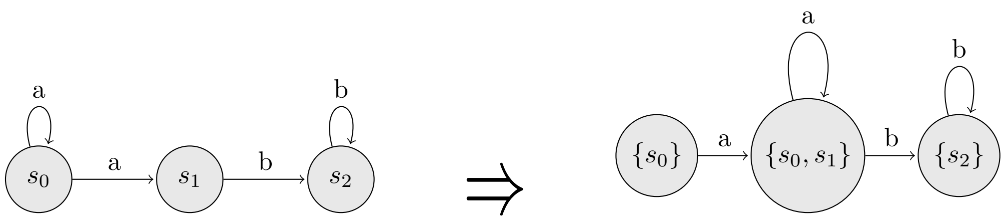
September 24, 2020

main.pdf 3.6

(2) Give the complete labelled transition relation for the 3-puzzle
We let $\{xyz4 \xrightarrow{l} xy4z\}$ denote $\{(xyz4, l, xy4z) \mid x \neq y \neq z \in \{1,2,3\}\}$ the set of transitions from states of the form $xyz4$ to states of the form $xy4z$ by the label l.

$$\begin{aligned} R = \{xyz4 \xrightarrow{l} xy4z\} &\cup \{xyz4 \xrightarrow{u} x4zy\} \\ \cup\{xy4z \xrightarrow{r} xyz4\} &\cup \{xy4z \xrightarrow{u} 4yxz\} \\ \cup\{x4yz \xrightarrow{l} 4xyz\} &\cup \{x4yz \xrightarrow{d} xzy4\} \\ \cup\{4xyz \xrightarrow{r} x4yz\} &\cup \{x4yz \xrightarrow{d} yx4z\} \end{aligned}$$

(3) Make the deterministic LTS for $(L, S, R) = (\{a, b\}, \{s_0, s_1, s_2\}, \{(s_0, a, s_0), (s_0, a, s_1), (s_1, b, s_2), (s_2, b, s_2)\})$



Nondeterministic $\Rightarrow$ deterministic LTS (ignoring states with no transitions)

in text the deterministic LTS

$(L', S', R') = (\{a, b\}, \mathcal{P}(S), \{(\{s_0, a, \{s_0, s_1\}), (\{s_0, s_1\}, a, \{s_0, s_1\}), (\{s_0\}, b, \{s_2\}), (\{s_2\}, b, \{s_2\})\})$

(5) Design an LTS with 11 states that describes the payment of at least 10 with units of 1, 2, 5 and 10. (no change).
Let $a \oplus b$ denote $min\{a + b, 10\}$ or

$$\begin{aligned} L &= \{1, 2, 5, 10\} \\ S &= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\} \\ R &= \{\{n, 1, n \oplus 1\} \cup \{n, 2, n \oplus 2\} \cup \{n, 5, n \oplus 5\} \cup \{n, 10, n \oplus 10\} \mid n \in S\} \end{aligned}$$

(6) Show that in any given LTS, if $s'', \boldsymbol{v} \vdash_R^* s', \lambda$ and $s', \boldsymbol{u} \vdash_R^* s, \lambda$, then $s'', \boldsymbol{uv} \vdash_R^* s, \lambda$.
$s'', \boldsymbol{v} \vdash_R^* s', \lambda$ means that our LTS contains the steps $s' \xrightarrow{v_0} \cdot \xrightarrow{v_1} ... \xrightarrow{v_n} s''$ where $v_0, v_1, ..., v_n$ are the symbols of $\boldsymbol{v}$. Similarly we haves $\xrightarrow{u_0} \cdot \xrightarrow{u_1} ... \xrightarrow{u_n} s'$. We can concatenate these paths to form $s \xrightarrow{u_0} \cdot \xrightarrow{u_1} ... \xrightarrow{u_n} s' \xrightarrow{v_0} \cdot \xrightarrow{v_1} ... \xrightarrow{v_n} s''$ which means $s'', \boldsymbol{uv} \vdash_R^* s, \lambda$

(8) Prove:
(1) For any state $x \in S$ and word $\boldsymbol{w} \in L^*$, there exists a unique state $Y_{x,\boldsymbol{w}} \in \mathcal{P}(S)$ such that $Y_{x,\boldsymbol{w}}, \boldsymbol{w} \vdash^*_{R_\mathcal{P}} \{x\}, \lambda$.
We prove (1) by induction on $\boldsymbol{w}$.
Step 0:

$$\boldsymbol{w} = \lambda \Rightarrow Y_{x,\boldsymbol{w}} = \{x\}$$

$\{x\}$ is a unique set so (1) holds for the empty word.
Induction step: Assume $Y_{x,\boldsymbol{w}}$ is a unique state, we look at $Y_{x,\boldsymbol{w}l}$ for some arbitrary label l. $Y_{x,\boldsymbol{w}l} = r(Y_{x,\boldsymbol{w}}, l)$ and since r is a function, this is a unique set.

main.pdf 5.6

(2) Let $(M, 1, *)$ be a finite monoid and $\varphi : \Sigma^* \to M$ a morhpishm from $(\Sigma^*, \lambda\cdot)$ to $(M, 1, *)$. Show that the image of a word is given by the images of its letters.

We want to show $\varphi(l_1 ... l_n) = \varphi(l_1) * ... * \varphi(l_n)$.

$$\begin{aligned}\varphi(l_1...l_n) &= \varphi(l_1 \cdot ... \cdot l_n) \text{ a word is the concatenation of its letters} \\ &= \varphi(l_1) * ... * \varphi(l_n) \text{ by the associativity of concatenation, and the morhpishm condition}\end{aligned}$$

Show that for every $F \subseteq M$, $L = \{\boldsymbol{w} \in \Sigma^* \mid \varphi(\boldsymbol{w}) \in F\}$ is regular.
We construct a finite state machine $M_L = (\Sigma, M, q_0 R, A)$ (using A for the set of Accepting states) with:

$$\begin{aligned} q_0 &= 1 \\ A &= F \\ R &= \{m \xrightarrow{l} m' \mid m, m' \in M, m * \varphi(l) = m'\} \end{aligned}$$

Accepting paths in this machine look like $1 \xrightarrow{l_1} 1 * \varphi(l_1) ... \xrightarrow{l_n} 1 * \varphi(l_1) * ... * \varphi(l_n) \in F$. By the morphism condition $1 * \varphi(l_1) * ... * \varphi(l_n) = \varphi(\lambda \cdot l_1 ... l_n) = \varphi(\boldsymbol{w})$ so this machine accepts L.

Show that $L' = \{a^{|\boldsymbol{w}|} \mid \boldsymbol{w} \in L\}$ is a regular language.
We can reuse the structure of M_L to define a new FSM $M_{L'} = (\Sigma, M, q_0 R', F)$ with $R' = \{m \xrightarrow{a} m' \mid m \to lm' \in R\}$. This new machine has the same structure as M_L, but every label is a. It accepts L'.

(3) Let L be a language over Σ. For $\boldsymbol{v}, \boldsymbol{w} \in \Sigma^*$ define $\boldsymbol{v} \sim_L \boldsymbol{w} := \boldsymbol{vu} \in L \iff \boldsymbol{wu} \in L$ for all $\boldsymbol{u} \in \Sigma^*$.
Show that $\sim_L$ is an equivalence relation:

- reflexive: $\boldsymbol{vu} = \boldsymbol{vu}$, so $\boldsymbol{v} \sim_L \boldsymbol{v}$
- symmetric: $\iff$ is symmetric, so $\sim_L$ is also
- transitive: let $\boldsymbol{x} \sim_L \boldsymbol{y}, \boldsymbol{y} \sim_L \boldsymbol{z}$. Then $\boldsymbol{xu} \in L \iff \boldsymbol{yu} \in L$ and $\boldsymbol{yu} \in L \iff \boldsymbol{zu} \in L$ so $\boldsymbol{xu} \in L \iff \boldsymbol{zu} \in L$

Show that all equivalence classes $[a^n], n \geq 0$ in $L = \{a^n b^n \mid n \geq 0\}$ are different.
$a^n \boldsymbol{u} \in L \iff \boldsymbol{u} = b^n$. Clearly $b^n \neq b^m$ for $n \neq m$ so $[a^n]$ is different for each n.

Show that all equivalence classes $[ba^n]$ are the same.
There is no $\boldsymbol{u}$ s.t $ba^n\boldsymbol{u} \in L$ so $[ba^n]$ is the class of words that cannot be extended to be in L for all n.

Consider the LTS $(\Sigma, \Sigma^*/\sim_L, R_L)$ with $R_L = \{([\boldsymbol{w}], l, [\boldsymbol{w}l])\}$ Show that R_L is well defined.
We want to show that $\boldsymbol{v} \sim_L \boldsymbol{w} \implies (R_L([\boldsymbol{v}], l, [\boldsymbol{v}l]) \iff R_L([\boldsymbol{w}], l, [\boldsymbol{w}l]))$.
$\boldsymbol{v} \sim_L \boldsymbol{w}$ means $\boldsymbol{vu} \in L \iff \boldsymbol{wu} \in L$ for all $\boldsymbol{u} \in \Sigma^*$. Consider $V = \{\boldsymbol{v} \in \Sigma^* \mid \boldsymbol{v} = l\boldsymbol{x}, \boldsymbol{x} \in \Sigma^*\} \subseteq \Sigma^*$ the subset of words starting with l. Since it is a subset, the equivalence implies $\boldsymbol{vu} \in L \iff \boldsymbol{wu} \in L$ for all $u \in V$, which means $\boldsymbol{v}l\boldsymbol{u} \in L \iff \boldsymbol{w}l\boldsymbol{v} \in L$ for all $u \in \Sigma^*$ – the same as $\boldsymbol{v}l \sim_L \boldsymbol{w}l$. Hence $[\boldsymbol{v}] \sim_L \boldsymbol{w} \iff [\boldsymbol{v}l] \sim_L [\boldsymbol{w}l]$ and R_L is well defined.

Show that the LTS is deterministic.
R_L is injective and well defined, so it is a function. An LTS where the transition relation is a function is deterministic.

Assume L is accepted by some FSM $(\Sigma, Q, q_0, \Upsilon, F)$.
Show that for any $q \in Q$ and words $\boldsymbol{v}, \boldsymbol{w} \in \Sigma^*$, if q is reachable from q_0 with both $\boldsymbol{v}$ and $\boldsymbol{w}$, then $\boldsymbol{v} \sim_L \boldsymbol{w}$.
q reachable from q_0 by $\boldsymbol{v}$ and $\boldsymbol{w}$ means there exists a path $q_0 \xrightarrow{l_0} ... \xrightarrow{l_n} q$ such that $l_0 \cdot ... \cdot l_n = \boldsymbol{v}$ and one where $l_0 \cdot ... \cdot l_n = \boldsymbol{w}$. Consider $\boldsymbol{u}$ such that $\boldsymbol{u} = l_0 \cdot ... \cdot l_n$ and the path $q \xrightarrow{l_0} ... \xrightarrow{l_n} q'$ is in the FSM (note that $\boldsymbol{u}$ could be the empty word, in which case $q = q'$). We can concatenate the paths to q with the path described by $\boldsymbol{u}$. Then both $\boldsymbol{vu}$ and $\boldsymbol{wu}$ are in L iff $q \in F$, so $\boldsymbol{v} \sim_L \boldsymbol{w}$.

Show that $\Sigma^*/\sim_L$ is finite.
Consider the function $\mu : [\boldsymbol{w}] \mapsto q$ reached by $\boldsymbol{w}$ from q_0. Our previous results shows that if two words both reach q, then they are in the same equivalence class, so μ is injective. We may assume every state is reachable by *some* word without affecting the language accepted by our FSM, so μ is also surjective. Hence μ is a bijection $\Sigma^*/\sim_L \rightarrow Q$ and since Q is finite, so is $\Sigma^*/\sim_L$.

Prove that the language $L' = \{a^nb^n \mid n \geq 0\}$ is irregular. (Not accepted by an FSM).
We have seen that an FSM $(\Sigma, Q, q_0, \Upsilon, F)$ which accepts L has $|Q| = |\Sigma^*/\sim_L|$. We have also seen that $[a^n]$ is distinct for each n. Hence $|\Sigma^*/\sim_{L'}|$ is not finite, and there can be no FSM accepting L'.

Book 3.1

(2) Which of the following are accepted?
aaabab and *bbbabab* are accepted.

(3) Write an expression for the language accepted by the automaton.
$\{a^nb \mid n \geq 0\} \cup \{(ba)^nb \mid n \geq 0\} = \{(a|ba)^nb \mid n \geq 0\}$

(6) Write an expression for the language accepted by the automaton.
$\{(a^n(a|b)ab)^m a^k(a|b)a \mid n, m, k \geq 0\}$

(15) Find a deterministic automaton accepting the same language as the given automaton.

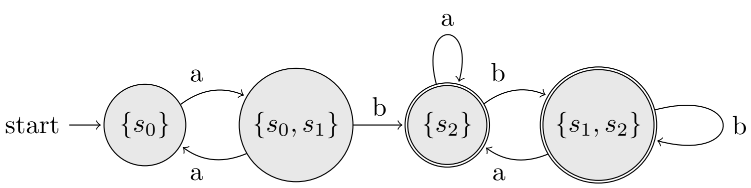


Figure 1: Deterministic automaton for (15)

(18) Find a deterministic automaton accepting the same language as the given automaton.

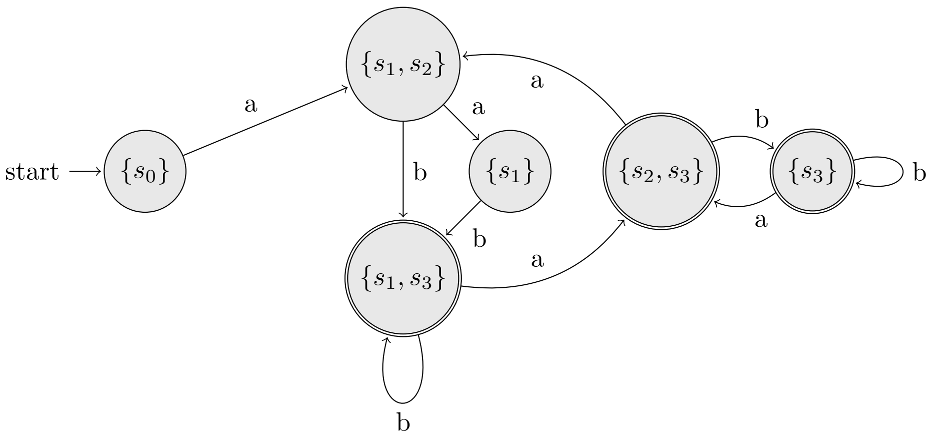


Figure 2: Deterministic automaton for (18)

Book 3.2

(2) a) Given finite automata for L_1 and L_2, construct one that accepts $L_1 \cup L_2$. Note that the complete automata have been left out for the sake of my sanity.

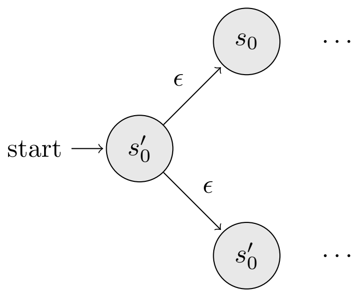


Figure 3: $L_1 \cup L_2$

(b) construct an automata for $L_1 \cdot L_2$

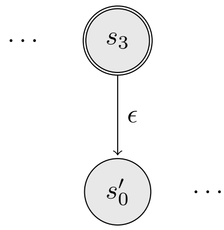


Figure 4: $L_1 \cdot L_2$

(c) construct automata for L_1^* and L_2^*

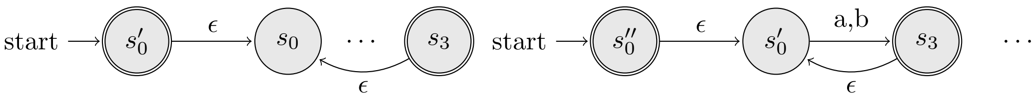


Figure 5: Automata for L_1^* (left) and L_2^* (right)

(5) Use transition graphs to construct the regular language accepted by the automaton.
We construct the following transition graph and determine that the accepted language is $\{b^n ab^m ab^k a \mid n, m, k \geq 0\}$.

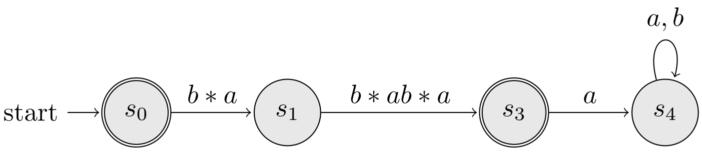


Figure 6: The transition graph for (5)

(7) Use transition graphs to construct the regular language accepted by the automaton.

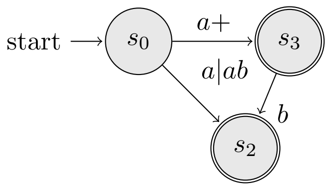


Figure 7: The transition graph for (7)

We note that the accepted language is $a + |a|ab|a + b$ (where + denotes one or more occurences of the preceding letter) which simplifies to $a + |a + b$