Exercises on Chapter 4

Beau De Clercq

October 2019

Ex. 4.1

Question

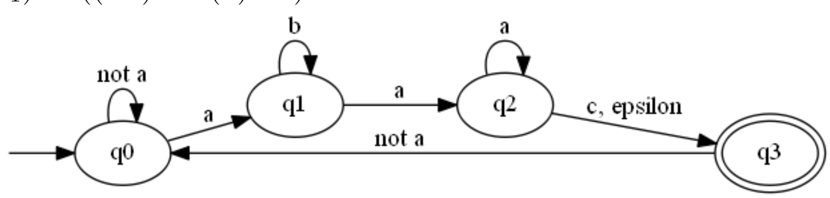
Let $AP = \{a, b, c\}$. Consider the following LT properties:

- If a becomes valid, afterward b stays valid ad infinitum or until c holds.
- Between two neighboring occurrences of a, b always holds.
- Between two neighboring occurrences of a, b occurs more often than c.
- $a \wedge \neg b$ and $b \wedge \neg a$ are valid in alternation or until c becomes valid.

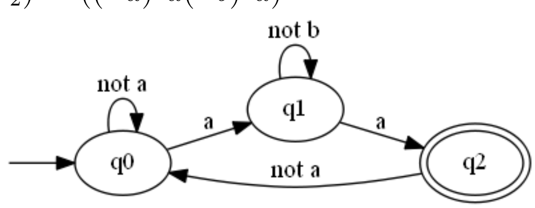
For each property $P_i (1 \leq i \leq 4)$, decide if it is a regular safety property (justify your answers) and if so, define the NFA A_i with $L(A_i) = \text{BadPref}(P_i)$.

Answer

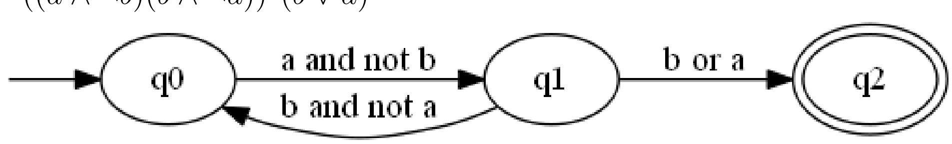
- $\text{BadPref}(P_1) = ((\neg a)^* ab^*(a)^+ c^?)^\omega$



- $\text{BadPref}(P_2) = ((\neg a)^* a(\neg b)^* a)^\omega$



- Not possible: counting is not possible.
- $\text{BadPref}(P_4) = ((a \wedge \neg b)(b \wedge \neg a))^*(b \vee a)$



Ex. 4.2

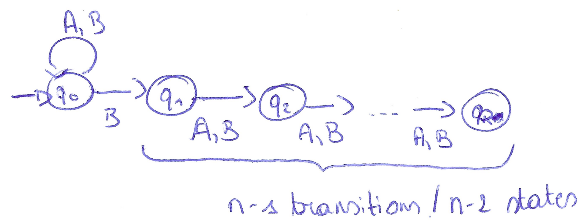
Question

Let $n \geq 1$. Consider the language $L_n \subseteq \Sigma^*$ over the alphabet $\Sigma = \{A, B\}$ that consists of all finite words where the symbol B is on position n from the right, i.e., L contains exactly the words $A_1 A_2 ... A_k \in \{A, B\}^*$where k≥n and $A_{kn+1} = B$. For instance, the word ABBAABAB is in L_3.

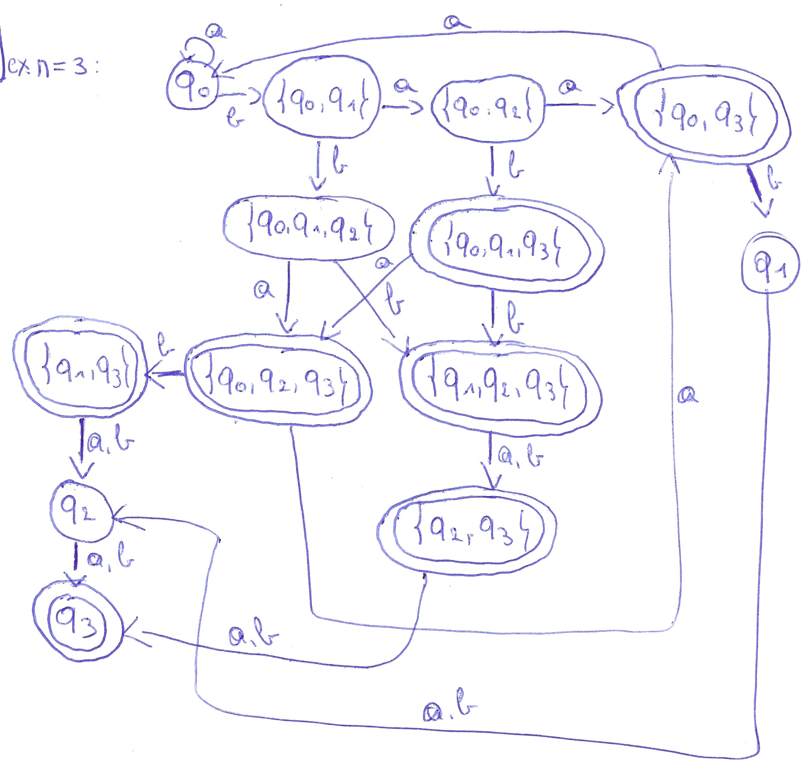
- Construct an NFA A_n with at most n+1 states such that $L(A_n) = L_n$.
- Determinize this NFA A_n using the powerset construction algorithm.

Answer

4.2 a)



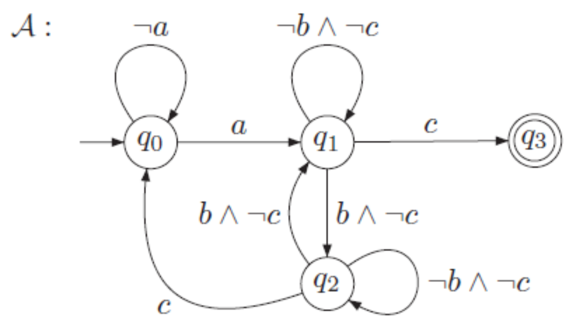
b) ex: n = 3:

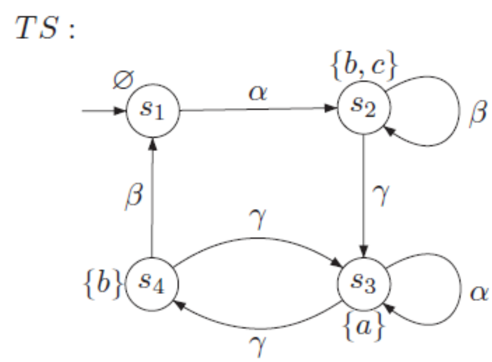


Ex. 4.5

Question

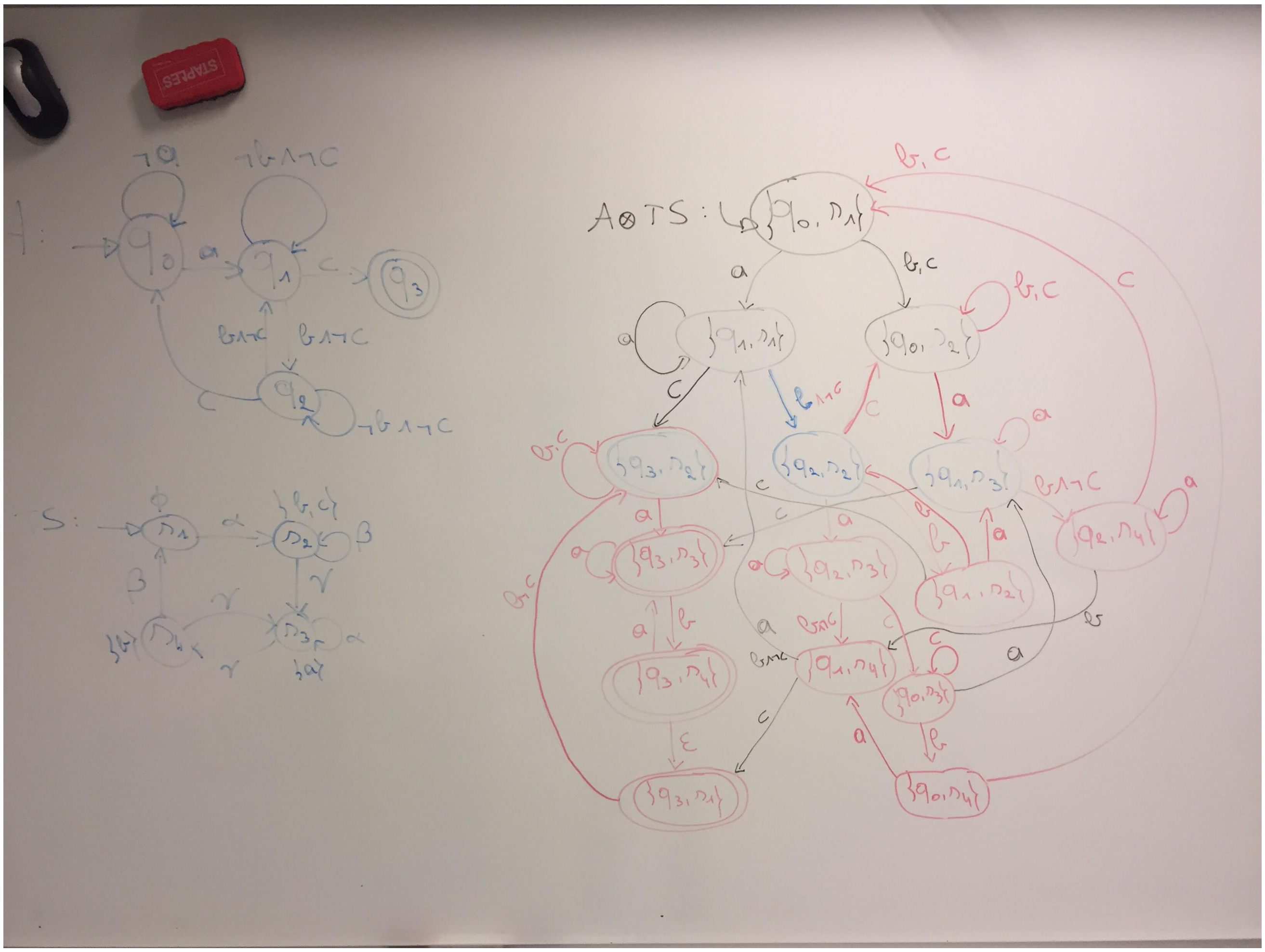
Let AP $= \{a, b, c\}$. Consider the following NFA A (over the alphabet 2^{AP}) and the following transition system TS:





Construct the product $TS \otimes A$ of the transition system and the NFA.

Answer

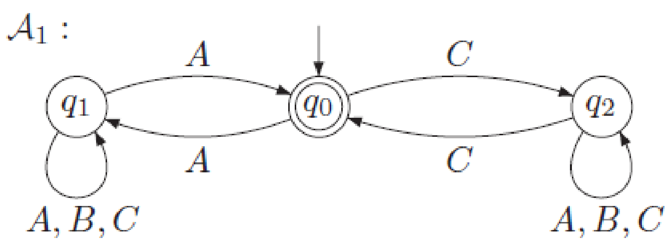


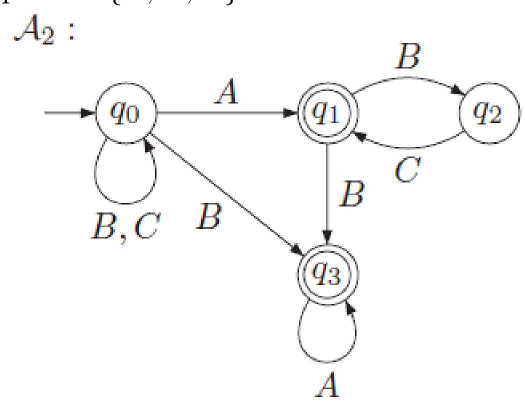
A:
S:
A⊗TS:
{q0, r1}
{q1, r1}
{q0, r2}
{q3, r2}
{q2, r2}
{q1, r3}
{q2, r4}
{q3, r3}
{q2, r3}
{q1, r2}
{q3, r4}
{q1, r4}
{q0, r3}
{q3, r1}
{q0, r4}

Ex. 4.12

Question

Consider the following NBA A_1 and A_2 over the alphabet $\{A, B, C\}$:





Find ω-regular expressions for the languages accepted by A_1 and A_2.

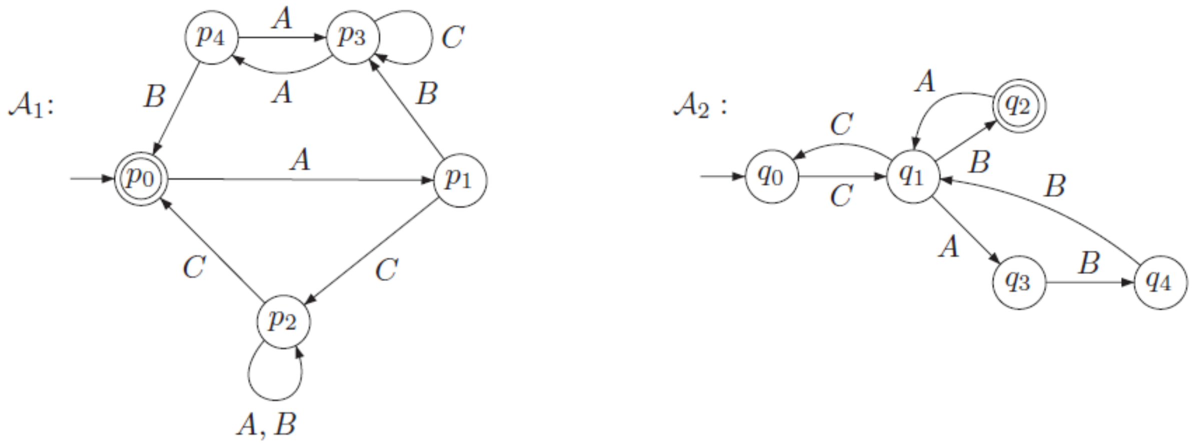
Answer

$$
\begin{aligned}
L = (\quad & ((A(A \vee B \vee C)^* A)^* \vee (C(A \vee B \vee C\vee)^* C)^*)^* \\
& ((B \vee C)^*((A((BC)^* \vee (BA^*))))) \\
)^\omega &
\end{aligned}
$$

Ex. 4.13

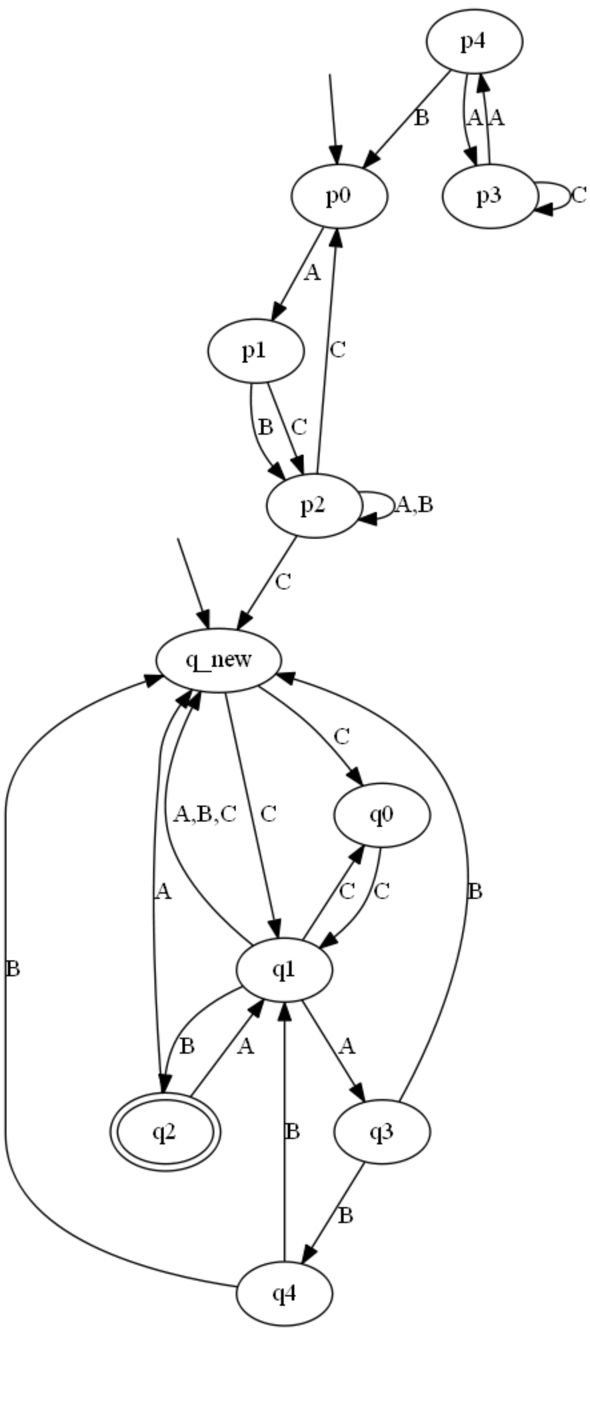
Question

Consider the NFA A_1 and A_2:



Construct an NBA for the language $L(A_1).L(A_2)^\omega$.

Answer



p4
B
A
A
p0
p3
C
A
p1
C
B
C
p2
A,B
C
q_new
C
A,B,C
C
q0
A
C
C
B
B
q1
B
A
A
q2
B
q3
B
q4

Ex. 4.14

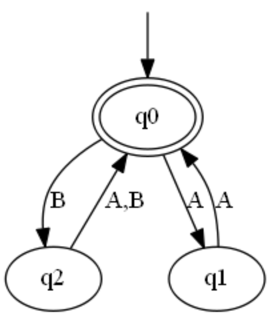
Question

Let AP $= \{a, b\}$. Give an NBA for the LT property consisting of the infinite words $A_0A_1A_2...(2^{AP})^\omega$ such that

$\overset{\infty}{\exists}\, j \geqslant 0.\, (a \in A_j \wedge b \in A_j)$ and $\exists j \geqslant 0.\, (a \in A_j \wedge b \notin A_j)$.

Provide an ω-regular expression for $L_\omega(A)$.

Answer



Ex. 4.15

Question

Let AP $= \{a, b, c\}$. Depict an NBA for the LT property consisting of the infinite words $A_0A_1A_2...(2^{AP})^\omega$ such that $\forall j \geq 0. A_{2j} \models (a \vee (b \wedge c))$.
Recall that $A \models (a \vee (b \wedge c))$ means $a \in A$ or $\{b, c\} \subseteq A$,i.e., $A \in \{\{a\}, \{b, c\}, \{a, b, c\}\}$.

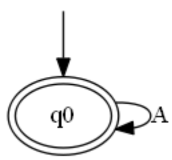
Answer

Ex. 4.16

Question

Consider NBA A_1 and A_2 depicted in Figure 4.26. Show that the powerset construction applied to A_1 and A_2 (viewed as NFA) yields the same deterministic automaton, while $L_\omega(A_1) \neq L_\omega(A_2)$.

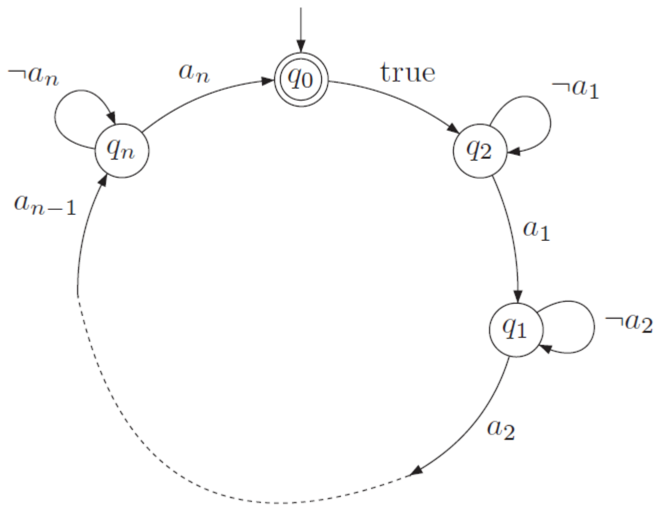
Answer

Both automatons have the same result of the powerset construction as depicted below. 

Ex. 4.17

Question

Consider the following NBA A with the alphabet $\Sigma = 2^{AP}$ where $AP = \{a1, ..., an\}$ for $n > 0$.



- Determine the accepted language $L_\omega(A)$.
- Show that there is no NBA A' with $L_\omega(A) = L_\omega(A')$ and less than n states.

Answer

- $L(A) = (true(\neg a_1)^* a_1 (\neg a_2)^* a_2 ...)^\omega$ (the language where a_i occurs infinitely often)
- The automaton to represent this language can be constructed with a minimum of n states. If it would be possible to do so in less than n states, the automaton would have to include a loop somewhere which would render some part of the automaton redundant and thereby changing the accepted language.