# Endterm Exam

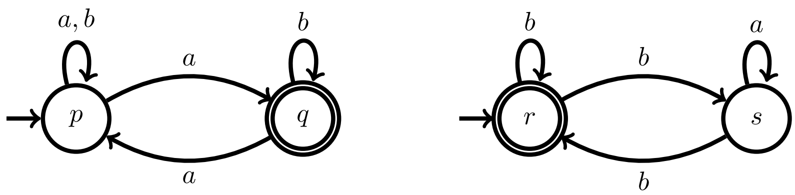
**Exercise 1**
Give two **NFA**s, each of them with at most 25 states, that accept the following languages $L_1$ and $L_2$. Note that you do not need to explicitly draw all the states, rather describe the structure of the automata precisely.

There are solutions with 12 states for $L_1$ and 11 states for $L_2$. However, full points will be awarded for any correct solution with at most 25 states.

(a) $L_1$ is the set of strings over the alphabet $\{0, 1, \ldots, 9\}$ such that the final digit has appeared before. For example, $4610216, 33, 2869322, 19401, 12343234 \in L_1$ and $25692553, 2, 14653345609, \epsilon \notin L_1$.

(b) $L_2$ is the set of strings over the alphabet $\{0, 1, \ldots, 9\}$ such that the final digit has *not* appeared before. For example, $46653, 5489890, 2 \in L_2$ and $5788, 578045, \epsilon \notin L_2$.

**Exercise 2**
Consider the following two NFAs $A_1$ and $A_2$.




(a) Use the *IntersNFA* algorithm, as described in the lectures, to construct an NFA $A$ which recognizes $\mathcal{L}(A_1) \cap \mathcal{L}(A_2)$. **Label the states of $A$ as 1, 2, 3 and 4, in the order by which the algorithm *IntersNFA* constructs them.**

(b) Use the *NFAtoRE* algorithm, as described in the lectures, to convert the NFA $A$ constructed in the previous subproblem, into a regular expression. **The solution must contain the automaton after the preprocessing step and also the automata obtained after removing each state.** Further,

- **You must remove states in ascending order**, i.e., you must first remove the state 1, then state 2 and so on.
- While writing the solution, if you come across a long regular expression, you can abbreviate it by a variable and use this abbreviation. **For example**, you can let $\sigma$ stand for the regular expression $(b^*a + a^*b)^*$ and then instead of writing $(b^*a + a^*b)^*$ throughout the solution, you can instead use $\sigma$.

**Exercise 3**
Let $L$ be a language. The operation $T(L)$ is defined as follows:

- Remove every even-length word from $L$.
- For each odd-length word, remove the middle letter.

For example, if $L = \{aab, bbaa, babab\}$, then $T(L) = \{ab, baab\}$. The even-length word $bbaa$ is deleted, the middle letter of $aab$ is removed to make $ab$, and the middle letter of $babab$ is removed to make $baab$.

For each of the following languages $L_i$, $i = 1, 2$, determine if $T(L_i)$ is regular or not. If it is regular give a regular expression representing it; if it is not regular prove this by analyzing the set of its residuals.

(a) $L_1$ is the language of the regular expression $(bab)^*$.

(b) $L_2$ is the language of the regular expression $aa^*bb^*$.

### Exercise 4

For a fixed-length language $L$ over $\Sigma = \{a, b\}$ we denote by $q_L$ the state of the master automaton representing $L$. We also denote by $M(L)$ the fragment of the master automaton that contains $q_L$ and all its residuals, that is, it contains all the states between $q_L$ and $q_\emptyset$, **including $q_L$ and $q_\emptyset$** (and no other states).

(a) Give a fixed-length language $L$ over $\Sigma = \{a, b\}$ of length 4, such that $M(L)$ has exactly 10 states. Enumerate the words of $L$ and draw $M(L)$.

(b) What is the maximal possible number of states in $M(L)$, if $L$ is a fixed-length language over $\Sigma = \{a, b\}$ of length 4? Justify your answer.

(c) What is the minimal possible number of states in $M(L)$, if $L$ is a fixed-length language over $\Sigma = \{a, b\}$ of length 4? Justify your answer.

### Exercise 5

(a) Let $\phi(x, y) = (x + y \geq 2) \wedge \exists z(y + z \leq 1)$, where $x, y, z$ range over natural numbers (including 0). Give an NFA $A_\varphi$ over $\{0, 1\}^2$ which recognizes the set of all least significant bit first (lsbf) encodings of pairs which satisfy the formula $\phi$. You can use any method to compute $A_\varphi$.

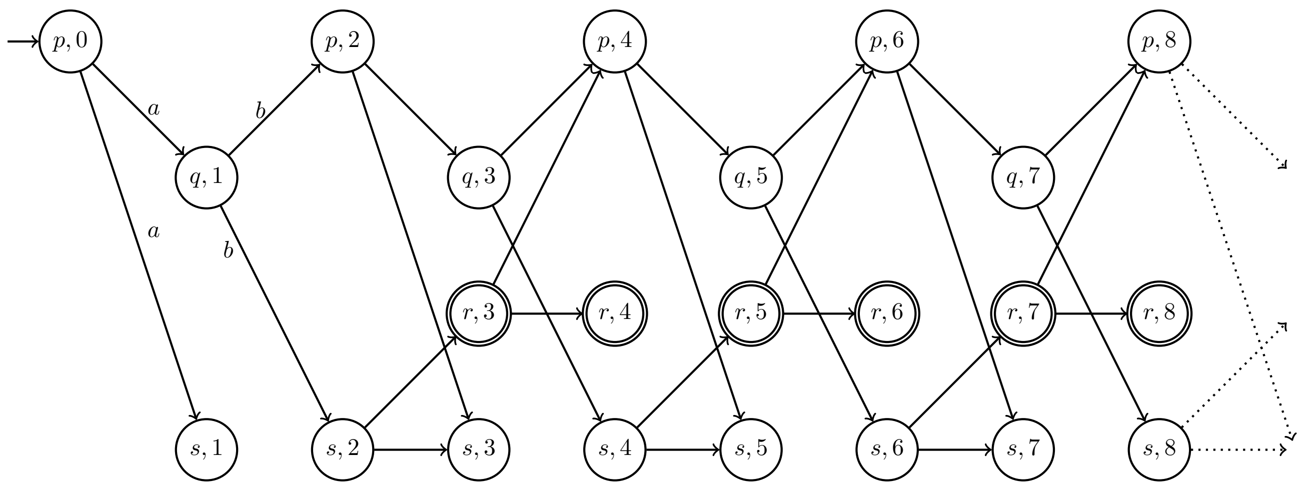
(b) Give a regular expression and an NFA recognizing the language of the following MSO formula:

$$\exists X, y.\ Q_b(y) \wedge \forall x(x \in X \Rightarrow Q_a(x)) \wedge \forall x(x = y \vee x \in X) \wedge \mathsf{evensize}(X)$$

Here $\mathsf{evensize}(X)$ is the formula which is true for a set $X$ if and only if $X$ contains an even number of elements.

### Exercise 6

The following is a fragment of dag$((ab)^\omega)$ on a Büchi automaton with four states $\{p, q, r, s\}$. We omit most of the edge labels to increase readability.



(a) Give a Büchi automaton $B$ with four states $\{p, q, r, s\}$ over $\Sigma = \{a, b\}$ such that $\text{dag}((ab)^\omega)$ on $B$ matches the graph from above.

(b) Prove that every infinite path of $\text{dag}((ab)^\omega)$ has only even stable rankings.

(c) Does $\text{dag}((ab)^\omega)$ have an odd ranking? Justify your answer.

**Exercise 7**

Recall: An LTL formula is a tautology if it is satisfied by all computations. For each of the following formulas over AP $= \{p, q, r\}$, check if it is a tautology. If it is a tautology, give a formal proof. Otherwise, give a computation that does not satisfy the formula.

(a) $\mathbf{G}(p \; \mathbf{U} \; \neg p) \Leftrightarrow \mathbf{G}(\neg p \; \mathbf{U} \; p)$

(b) $((\mathbf{G}\neg p) \; \mathbf{U} \; p) \Leftrightarrow p$

(c) $\big(((\mathbf{G}\neg p) \; \mathbf{U} \; p) \wedge \neg p\big) \Rightarrow \big((\mathbf{G}q) \vee (\neg q \; \mathbf{U} \; r)\big)$

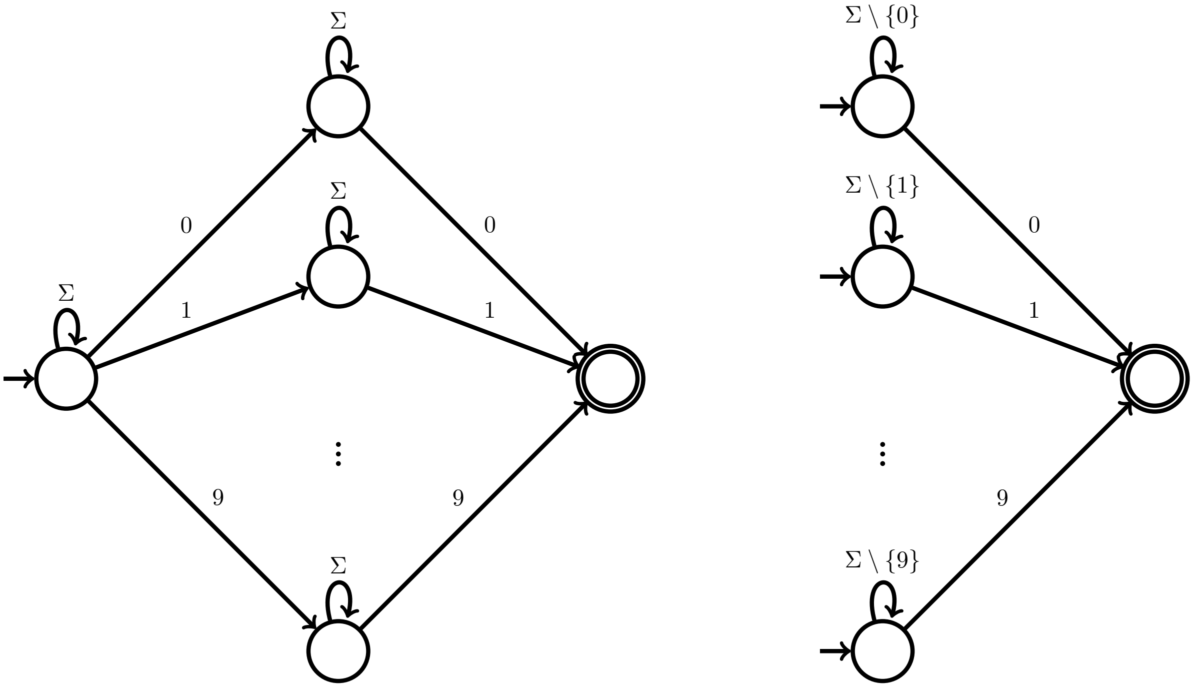
**Exercise 8**

## This is a bonus question.

Let $S = (Q, \Sigma, \delta, Q_0)$ be a semi-automaton. Let $\mathcal{F} = \{F_1, \ldots, F_k\}$ where each $F_i \subseteq Q$. Let $M_{S,\mathcal{F}}$ be the set of **infinite words** $w$ such that there is a run $\rho$ of $w$ on $S$ and a set $F_i \in \mathcal{F}$ with the property that the set of states appearing in the run $\rho$ is **exactly** $F_i$. Prove that $M_{S,\mathcal{F}}$ is $\omega$-regular.

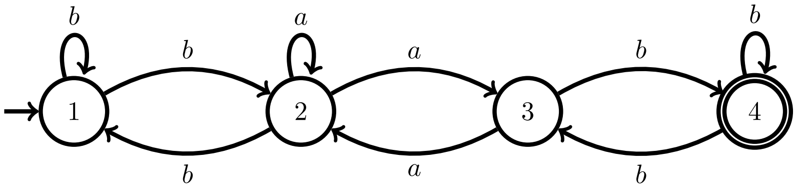
**Solution 1**

The solutions for (a) and (b) are given below in the left hand side and the right hand side respectively.

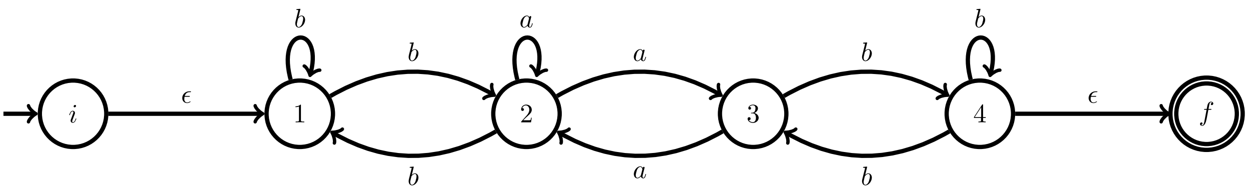



**Solution 2**

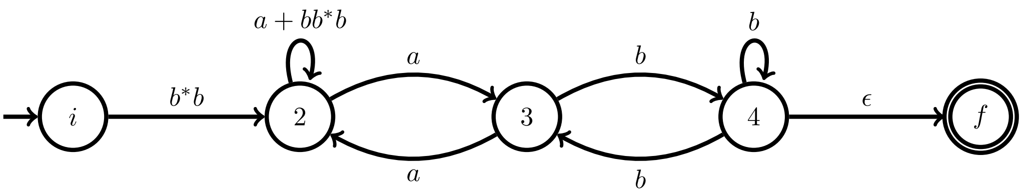
(a) The NFA obtained is as follows:



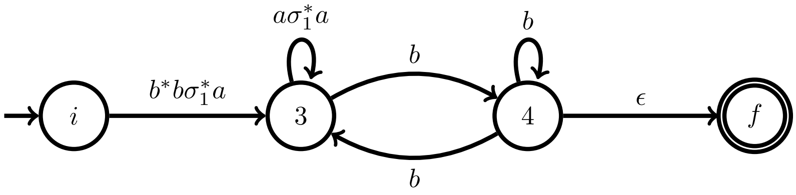

(b) After the preprocessing step, we have the following automaton:



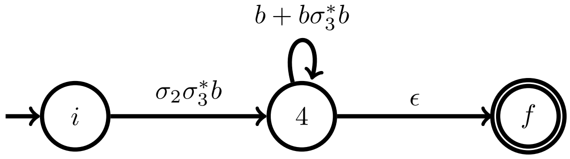

We now remove state 1 and merge transitions along the way.



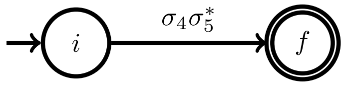

Let $\sigma_1 := a + bb^*b$. We now remove state 2.




Let $\sigma_2 := b^*b\sigma_1^*a$ and $\sigma_3 := a\sigma_1^*a$. We now remove state 3.




Let $\sigma_4 := \sigma_2\sigma_3^*b$ and $\sigma_5 := b + b\sigma_3^*b$. We now remove state 4.




Hence, the final regular expression is $\sigma_4\sigma_5^*$.

**Note:** 1 point each has been awarded for removing states 1, 2 and 3. Half a point each has been awarded for preprocessing and removing state 4. For the solution of this subproblem, if the initial automaton (i.e., the NFA obtained from the subproblem 2a)) is false then half a point has been deducted.

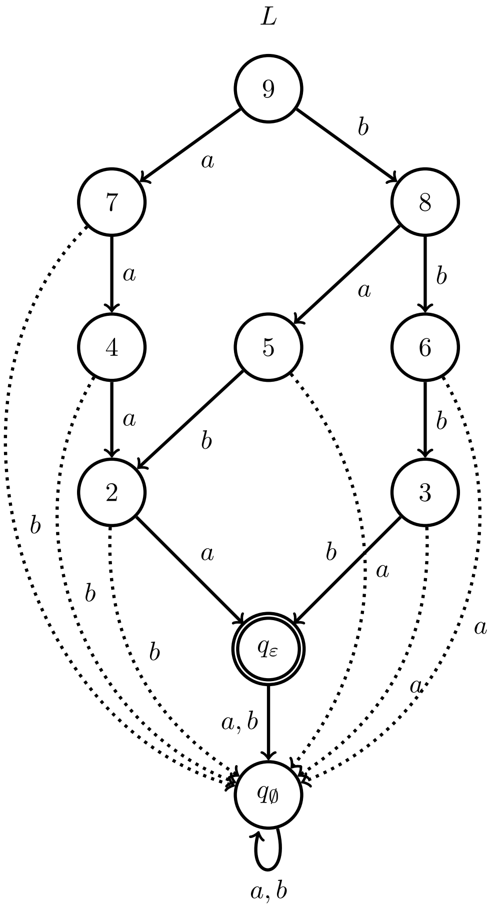
**Solution 3**

1. $L_1$ is not regular. Notice that the middle letter of a word of the form $(bab)^{2n+1}$ for some $n$ is always an $a$. It follows that $T(L_1) = \{(bab)^n bb(bab)^n : n \in \mathbb{N}\}$. This language is not regular, because there are infinitely many residuals of the form $L_1^{(bab)^i bb}$, for $i \geq 0$. Namely, if $i \neq j$, then the residual of $(bab)^i bb$ is different from $(bab)^j bb$, since the word $(bab)^i$ belongs to $L_1^{(bab)^i bb}$ but it does not belong to $L_1^{(bab)^j bb}$.

2. $L_2$ is regular: $T(L_2)$ is the set of all even-length words from $L_2$. The regular expression describing it is $a(aa)^*b(bb)^* + aa(aa)^*bb(bb)^*$.

**Solution 4**

1. There are many possible solutions. Here is one for the language $L = \{aaaa, baba, bbbb\}$.




| Ident. | Language |
|---|---|
| 0 | $\emptyset$ |
| 1 | $\{\varepsilon\}$ |
| 2 | $\{a\}$ |
| 3 | $\{b\}$ |
| 4 | $\{aa\}$ |
| 5 | $\{ba\}$ |
| 6 | $\{bb\}$ |
| 7 | $\{aaa\}$ |
| 8 | $\{aba, bbb\}$ |
| 9 | $\{aaaa, baba, bbbb\}$ |

2. The maximal possible number of states in $M(L)$ is 12. Let us consider what is the maximal number of states in each layer.

   Layer 4: one state representing the language $L$
   Layer 3: since we have 2 letters in $\Sigma$, we can have at most 2 states representing residuals $L^a$ and $L^b$
   Layer 2: again, there are at most 2 residuals for each of the 2 states from the 3rd layer, thus 4 states
   Layer 1: there are at most 3 languages of fixed length 1, namely $\{a\}$, $\{b\}$ and $\{a, b\}$
   Layer 0: there are 2 states, namely $q_\emptyset$ and $q_\varepsilon$.

   In total, this is $1 + 2 + 4 + 3 + 2 = 12$.

3. The minimal possible number of states in $M(L)$ is 6. This is the case when there is only one state in each layer, and of course 2 states in layer 0.

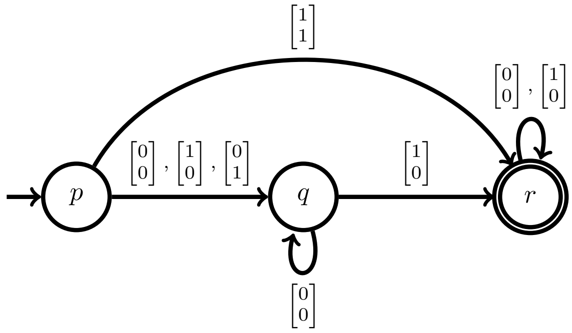
   ★ Note: In this exercise we assume that $L$ is nonempty. But since we did not state this explicitly, we also accept this alternative answer:

   The empty language is considered to be the language of every size, and therefore also of size 4. For this language, we have $M(\emptyset)$ with only one state $q_\emptyset$. Thus, the answer is 1.

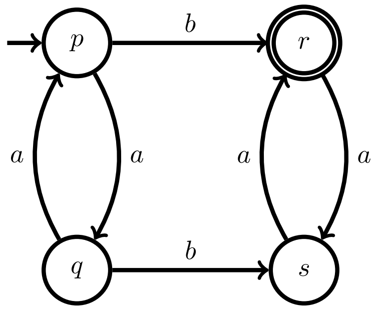
★ A common mistake: Note that for $L = \Sigma^4$ we have $M(L)$ with only 6 states. This is indeed the language with maximal possible number of words of length 4, but still, $M(L)$ is not maximal possible, but rather minimal possible.

**Solution 5**

(a) The given formula is essentially equivalent to $(x + y \geq 2) \wedge (y \leq 1)$. Here is one possible solution:

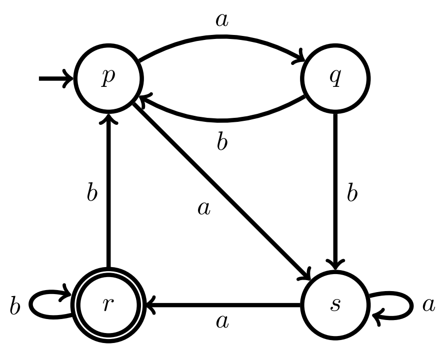



(b) The language of the MSO formula is the set of all words which have exactly one $b$ and an even number of $a$'s. A regular expression for this language is $(aa)^*b(aa)^* + (aa)^*aba(aa)^*$. An NFA for this language is




**Solution 6**

(a) The following is a solution.




(b) Pick an infinite path that visits all 4 states $p, q, s, r$ infinitely often. That path visits the accepting state $r$ infinitely often, which has to have an even rank. Therefore, eventually the rank will stabilize and from some point on all nodes visited by this path will have an even rank $k$. In particular, all nodes representing $p$ and $q$ visited by this path will have rank $k$ from some point on, let us say from level $l_0$. Having that in mind, if we now consider the path that visits only $p$ and $q$, we obtain that all nodes $(p, l)$ and $(q, l)$ for every $l \geq l_0$, will have the same rank $k$.

Assume now, by means of contradiction, that there is an infinite path in dag$((ab)^\omega)$ that visits nodes of **odd** rank infinitely often. Note that this path must visit $p$ and $q$ infinitely often. Thus, from level $l_0$ all nodes visited by this path will have the same **even** rank $k$, which yields a contradiction.

(c) No, it doesn't. A ranking is odd if every infinite path visits nodes of odd rank infinitely often. Thus, it is enough to find an infinite path that does not visit nodes of odd rank i.o. For example, the path $p, q, s, r, p, q, s, r, p, q, s, r \ldots$ visits $r$ infinitely often, and $r$ has to have an even rank. Thus, from some point on, all nodes visited by this path will have an even rank.

An alternative answer: note that in part (b) we have shown that every path has even stable ranking, which implies that there is no odd ranking.

**Solution 7**

(a) The formula is not a tautology. Here are two counterexamples: $\{p\}^\omega$ and $\{\}^\omega$.

(b) This formula is a tautology. Let us prove the equivalence in both directions.

($\Leftarrow$) If $p$ holds, then trivially we have $\varphi\ \mathbf{U}\ p$ for any formula $\varphi$.

($\Rightarrow$) Let us now prove that $(\mathbf{G}\neg p)\ \mathbf{U}\ p$ implies $p$. Let $\sigma$ be an arbitrary computation with $\sigma \models (\mathbf{G}\neg p)\ \mathbf{U}\ p$. Then we need to show that $\sigma \models p$.

By definition of the $\mathbf{U}$ operator, there exists an $i \geq 0$ such that $\sigma^i \models p$ and for every $k$ with $0 \leq k < i$ we have $\sigma^k \models \mathbf{G}\neg p$. Our goal is to show that $i = 0$, as this gives us directly that $\sigma = \sigma^i \models p$.

Assume by contradiction that $i > 0$ and $\sigma^i \models p$. Recall that for every $k$ with $0 \leq k < i$ we have $\sigma^k \models \mathbf{G}\neg p$. It is enough to focus on $k = 0$, as this yields $\sigma^k = \sigma \models \mathbf{G}\neg p$. By definition of the $\mathbf{G}$ operator, this means that for every $j \geq 0$ we have $\sigma^j \models \neg p$. This is a contradiction with $\sigma^i \models p$.

(c) According to the previous tautology, we know that $\big(((\mathbf{G}\neg p)\ \mathbf{U}\ p) \land \neg p\big)$ is equivalent to $\big(p \land \neg p\big)$. Since this is false, the implication must always be true.

**Solution 8**

Let $\mathcal{F}_i = \{F_i\}$, i.e., $\mathcal{F}_i$ is the subset of $\mathcal{F}$ which contains only $F_i$. For each $i$, we will show that $M_{S,\mathcal{F}_i}$ is $\omega$-regular. Since $\omega$-regular languages are closed under union, the required result will then follow.

Fix an $i$. Construct the semi-automaton $S_i = (F_i \times 2^{F_i}, \Sigma, \delta_i, Q_0^i)$, where

- $((q, C), a, (q', C')) \in \delta_i$ if and only if $(q, a, q') \in \delta$ and $C' = C \cup \{q'\}$
- $Q_0^i := \{(q, \{q\}) : q \in Q_0 \cap F_i\}$

Intuitively, the first component of our state space simulates a run of $S$ on the states $F_i$ and the second component tracks the subset of states visited so far.

Let $F_i' = \{(q, F_i) : q \in F_i\}$ be a set of final states interpreted as a Büchi condition. From the construction it follows that the Büchi automaton given by $(S_i, F_i')$ recognizes the language $M_{S,\mathcal{F}_i}$.