Tutorial 3: Solutions

Exercise 1: The alphabet Σ is the set of all printable ASCII characters:

```
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```

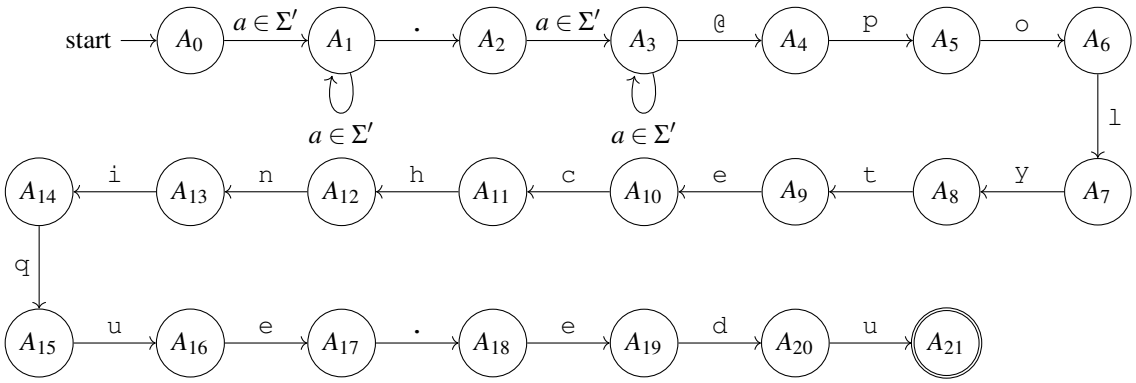
For each of the following languages, find an accepting finite automaton, then deduce the corresponding grammar (or the other way round):

1. the email addresses following the pattern *<firstname>*`.`*<lastname>*`@polytechnique.fr`
2. the Python language identifiers: «it starts with a letter (`A` ... `Z`) or (`a` ... `z`) or an underscore (_) followed by zero or more letters, underscores and digits (`0` ... `9`).»
3. the decimal representation of $3\mathbb{N}$ (Hint: if $d_1 \cdots d_k$ is the decimal representation of n, then 3 divides n iff 3 divides $d_1 + \cdots + d_k$).

Hint: any state of the automaton/nonterminal symbol of the grammar should be associated with a «meaning». For the first question there should be one state/symbol which means «before dot», another one to mean «after dot and before `@`» etc.

Solution:

1. We define $\Sigma' = \{a-z, A-Z\}$ as the set of letters. The automaton that recognize the language is:



The initial state of the automaton is A_0 and the only final state is A_{22}. We can easily write the automaton as structure (Σ, Q, Q_0, F, R) (e.g., by listing all the states, the initial and final state, all the automaton edges in R). The grammar $(\Sigma, \{A_0, \ldots, A_{20}\}, A_0, \mathscr{P})$, where the set of $\mathscr{P}$ contains the following rules:

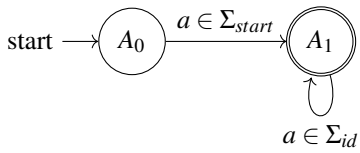
$A_0 \to aA_1, \forall a \in \Sigma'$ $A_1 \to aA_1, \forall a \in \Sigma'$ $A_1 \to \texttt{.}A_2$ $A_2 \to aA_3, \forall a \in \Sigma'$ $A_3 \to aA_3, \forall a \in \Sigma'$

$A_3 \to \texttt{@}A_4$ $A_4 \to \texttt{p}A_5$ $A_5 \to \texttt{o}A_5$ $A_6 \to \texttt{l}A_7$ $A_7 \to \texttt{i}A_8$ $A_8 \to \texttt{t}A_9$ $A_9 \to \texttt{e}A_{10}$

$A_{10} \to \texttt{c}A_{11}$ $A_{11} \to \texttt{h}A_{12}$ $A_{12} \to \texttt{n}A_{13}$ $A_{13} \to \texttt{i}A_{14}$ $A_{14} \to \texttt{q}A_{15}$ $A_{15} \to \texttt{u}A_{16}$

$A_{16} \to \texttt{e}A_{17}$ $A_{17} \to \texttt{.}A_{18}$ $A_{18} \to \texttt{e}A_{19}$ $A_{19} \to \texttt{d}A_{20}$ $A_{20} \to \texttt{u}$

2. Let define the set of symbols $\Sigma_{start} = \{\texttt{a}, \ldots, \texttt{z}, \texttt{A}, \ldots, \texttt{Z}, \_\}$, and the set of identifier symbols $\Sigma_{id} = \Sigma_{start} \cup \{\texttt{0}, \ldots, \texttt{9}\}$. The automaton that recognize the set of Python ids is:



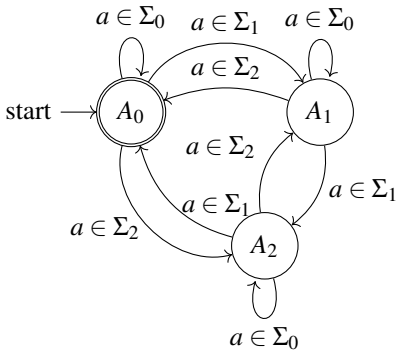
The grammar corresponding to the automaton is $(\Sigma, \{A_0, A_1\}, A_0, \mathscr{P})$, where the set of $\mathscr{P}$ contains the following rules:

$$A_0 \to aA_1, \forall a \in \Sigma_{start} \qquad A_1 \to a, \forall a \in \Sigma_{id}$$

3. Let define three subset of the alphabet $\Sigma_0 = \{0,3,6,9\}$ $\Sigma_1 = \{1,4,7\}$ $\Sigma_2 = \{2,5,8\}$. Each alphabet contains all the digits such that the digit modulo 3 has the same result (e.g., 0 for Σ_0, ...).

We define three states for the automaton to "store" the fact that the sum of the digits that we have seen so far modulo 3 is either 0 (A_0), 1 (A_1), and 2 (A_2). We accept a word when we are in the state where the modulo of the sum is 0. If a path of the automaton ends in that state then it means that the digit represented by the recognized word is a multiple of 3.

The automaton is defined as follows:



A state A_i transition to a state A_j with a digit n only if $(i+n) \mod 3 = j$.

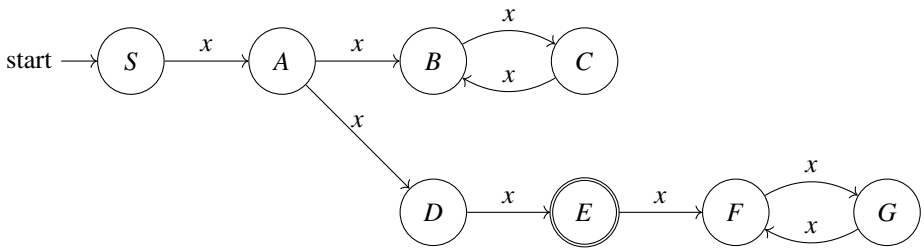
The grammar corresponding to the automaton is $(\Sigma, \{A_0, A_1, A_2\}, A_0, \mathscr{P})$, where the set of $\mathscr{P}$ contains the following rules:

$$A_0 \to \varepsilon \qquad A_i \to dA_{(i+d) \mod 3}, \forall i \in \{0,1,2\} \text{ and } \forall d \in \{\texttt{0}, \ldots, \texttt{9}\}$$

Exercise 2: Let $\Sigma = \{\texttt{x}\}$. Draw the finite automaton associated to the grammar below, then deduce its trim automaton and the language it accepts.

$S \to \texttt{x}A \quad A \to \texttt{x}B \quad B \to \texttt{x}C \quad C \to \texttt{x}B \quad A \to \texttt{x}D \quad D \to \texttt{x}E \quad F \to \texttt{x}E \quad F \to \texttt{x}G \quad G \to \texttt{x}F \quad E \to \varepsilon$

Solution: We write the automaton A corresponding to the grammar:



The trim automaton of A contains only the state (and the transitions from such states) of the automaton A that are on a path from the initial state to the final state (i.e., the automaton paths that accept a word).

To compute the trim automaton we have to:

- Compute the set of states reachable from the initial states of A, computing the closure of the transition relation R.

  For A we compute the reachable states as follows:

$$\begin{aligned}
Q_0 &= I = \{S\} \\
Q_1 &= Q_0 \cup \{A\} = \{S,A\} \\
Q_2 &= Q_1 \cup \{B,D\} = \{S,A,B,D\} \\
Q_3 &= Q_2 \cup \{C,E\} = \{S,A,B,D,C,E\} \\
Q_4 &= Q_3 \cup \{\} = \{S,A,B,D,C,E\}
\end{aligned}$$

  Since $Q_4 = Q_3$ the set of reachable states is $R = \{S,A,B,D,C,E\}$

- Compute the set of co-reachable states: these are the states that are reachable from the final states of A following the transitions of A "backward" (they are the reachable states of the reverse automaton of A).

$$\begin{aligned}
C_0 &= F = \{E\} \\
C_1 &= C_0 \cup \{D,F\} = \{D,E,F\} \\
C_2 &= C_1 \cup \{G,A\} = \{D,E,F,G,A\} \\
C_3 &= C_2 \cup \{S\} = \{D,E,F,G,A,S\} \\
C_4 &= C_4 \cup \{\} = \{D,E,F,G,A,S\}
\end{aligned}$$

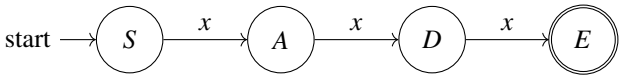
  Since $C_3 = C_4$ the set of co-reachable states is $C = \{D,E,F,G,A,S\}$

- Take the intersection of the reachable and co-reachable states: these are the states that can either be reached from the initial states of A and that can reach a final state of A.

  The intersection of reachable and co-reachable states of A is $\{S,A,B,D,C,E\} \cap \{D,E,F,G,A,S\} = \{S,A,D,E\}$

- Compute the sub-automaton of A generated by the intersection of reachable and co-reachable states. The restriction keeps only such states of A, and only transitions between those states.

  The trim automaton for A is:



From the trim automaton we can easily see if the language recognized by A is empty — and more easily understand what such language is.

The language recognized by A is $\{xxx\}$.

Exercise 3: Prove that the language $\{a^n b^n \mid n \in \mathbb{N}\}$ on the alphabet $\{a,b\}$ is *not* regular (Use the contrapositive form of the pumping lemma).

Solution: We prove that the language $L = \{a^n b^n \mid n \in \mathbb{N}\}$ is not regular using the contrapositive version of the pumping lemma.

To show that L is not regualar we have to show that for all $k \in \mathbb{N}$, there exist a strings x, y, z such that $xyz \in L$, $|y| \geq k$ and for all u, v, w with $y = uvw$ and $v \neq \varepsilon$, there exists an $i \geq 0$ such that $xuv^i wz \notin L$.

In practice, we assume an arbitrary $k \in \mathbb{N}$ and we pick the strings xyz:

$$x = a^k, y = b^k, z = \varepsilon$$

Note that $|y| \geq k$.

Then, we assume y to be a concatenation of three strings $y = uvw$ (that we don't know), with $|v| > 0$.

To satisfy the condition of the contrapositive version of the pumping lemma we have to find an i such that $xuv^i wz \notin L$.

We can pick $i = 2$:

$$\begin{aligned} xuv^2wz &= a^k b^{|u|} b^{2|v|} b^{|w|} \\ &= a^k b^{|u|+|v|+|w|+|v|} \\ &= a^k b^{k+|v|} && |y| = k = |u| + |v| + |w| \end{aligned}$$

We know that $a^k b^{k+|v|} \notin L$ since $|v| > 0$ (and hence $k \neq k + |v|$). Thus L is not regular.

Exercise 4: Let $(\Sigma_1, Q_1, I_1, F_1, R_1)$ and $(\Sigma_2, Q_2, I_2, F_2, R_2)$ be two finite automata accepting the (regular) languages L_1 and L_2 respectively.

1. What is the language recognized by the automaton $(\Sigma_1 \times \Sigma_2, Q_1 \times Q_2, I_1 \times I_2, F_1 \times F_2, R_1 \times R_2)$ where $R_1 \times R_2$ is actually seen as the set of triples

$$\Big\{ \big((A_1, A_2), (a_1, a_2), (B_1, B_2)\big) \;\Big|\; (A_1, a_1, B_1) \in R_1 \quad \text{and} \quad (A_2, a_2, B_2) \in R_2 \Big\}$$

2. Describe a finite automaton accepting the intersection $L_1 \cap L_2$ of the languages L_1 and L_2. What can be said about the intersection of two regular languages?

Solution:

1. The automaton recognize the cartesian product $L_1 \times L_2$ of two languages:

$$\begin{aligned} L_1 \times L_2 = \{ w \in \Sigma_1 \times \Sigma_2)^* \mid & w_1 \in L_1, w_2 \in L_2, |w| = |w_1| = |w_2|, \\ & \forall_{1 \leq i \leq |w|} (w(i) = (w_1(i), w_2(i))) \} \end{aligned}$$

2. We assume the automaton $A_1 = (\Sigma_1, Q_1, I_1, F_1, R_1)$ recognizes the language L_1, and the automaton $A_2 = (\Sigma_2, Q_2, I_2, F_2, R_2)$ recognizes the language L_2).

   The following automaton recognize the intersection $L_1 \cap L_2$:

$$A_1 \cap A_2 = (\Sigma_1 \times \Sigma_2, Q_1 \times Q_2, I_1 \times I_2, F_1 \times F_2, R_1 \cap R_2)$$

where we define $R_1 \cap R_2$ as:

$$R_1 \cap R_2 = \{ ((A_1, A_2), a, (B_1, B_2)) \mid (A_1, a, B_1) \in R_1 \quad \text{and} \quad (A_2, a, B_2) \in R_2 \}$$

We can say that the intersection of two regular language is a regular language — regular languages are closed under intersection.

Note that, formally, we should prove that $[\![A_1 \cap A_2]\!] = L_1 \cap L_2$.

Exercise 5: A word is well-parenthesized when

- it contains as many opening parentheses as closing ones, and
- in each of its prefixes, there is more opening parentheses than closing ones.

Prove that:

1. the language $\{a^n b^m \mid n, m \in \mathbb{N}\}$ on the alphabet $\{a, b\}$ is regular
2. the language of well-parenthesized word on the alphabet $\{(,)\}$ (a.k.a. the Dyck language) is *not* regular.

Solution:

1. We show that the language $\{a^n b^m \mid n, m \in \mathbb{N}\}$ is regular.

   We know that the set is the concatenation of the set $\{a^n \mid n \in \mathbb{N}\}$ and $\{b^n \mid n \in \mathbb{N}\}$:

   $$\{a^n b^m \mid n, m \in \mathbb{N}\} = \{a^n \mid n \in \mathbb{N}\}\{b^n \mid n \in \mathbb{N}\}$$

   The set $\{a^n \mid n \in \mathbb{N}\}$ is regular since the following regular grammar recognizes the same language $(\{a, b\}, \{S\}, S, \{S \to \varepsilon | aS\})$ (we could easily prove it by induction).

   Similary, the set $\{b^n \mid n \in \mathbb{N}\}$ is regular.

   Since regular languages are closed under concatenation, $\{a^n b^m \mid n, m \in \mathbb{N}\}$ is regular.

2. Let L_D be the language of well-parenthesised words over the alphabet $\{(,)\}$.

   We know that the set $\{(^n)^n \mid n \in \mathbb{N}\}$ is not regular (from Exercise 4). We show that L_D is not regular using the knowledge that $\{(^n)^n \mid n \in \mathbb{N}\}$ is not regular and $\{(^n)^m \mid n, m \in \mathbb{N}\}$ is regular.

   We have that $L_D \cap \{(^n)^m \mid n, m \in \mathbb{N}\} = \{(^n)^n \mid n \in \mathbb{N}\}$. To see this consider the two cases:

   - If $w \in \{(^n)^n \mid n \in \mathbb{N}\}$, then $w = (^n)^n$ for some n.
     $(^n)^n$ is a well-parenthesised word (it contains as many opening parenthesis as closing ones, and in each of its prefixes the number of opening parenthesis is less or equal than the number of closing parenthesis).
     $(^n)^n$ is also in $\{(^n)^m \mid n, m \in \mathbb{N}\}$, when $n = m$.
   - If $w \in L_D \cap \{(^n)^m \mid n, m \in \mathbb{N}\}$, then $w \in \{(^n)^m \mid n, m \in \mathbb{N}\}$ and $w \in L_D$. Thus $w = (^n)^m$, but also w has the same number of opening and closing parenthesis. Hence $w = (^n)^n \in \{(^n)^n \mid n \in \mathbb{N}\}$

   Thus, we have that the intersection of a regular language $\{(^n)^m \mid n, m \in \mathbb{N}\}$ with L_D is not a regular language. Since regular languages are closed under intersection, L_D cannot be regular.

Exercise 6: Let G be the grammar in which $\Sigma = \{\text{digits}\} = \{0,1,\ldots,9\}$, $\mathscr{N} = \{A_0, A_1, A_2, A_3, A_4, A_5, A_6\}$, the start symbol is A_0, and the production rules are, for all $d \in \Sigma$ and $n \in \{0,1,\ldots,6\}$

$$A_0 \to \varepsilon \qquad\qquad A_n \to dA_{(3n+d) \bmod 7}$$

Find some elements of $[\![G]\!]$ then guess what $[\![G]\!]$ is. Explain that result. In particular, what becomes $[\![G]\!]$ is we replace the rules $A_n \to dA_{(3n+d) \bmod 7}$ by the rules $A_n \to dA_{(10n+d) \bmod 7}$? State a generalization of that result (just try to make a guess).

Solution:

1. Find some elements of $[\![G]\!]$:

$$A_0 \to \varepsilon$$
$$A_0 \to 1A_1 \to 14A_0 \to 14$$
$$A_0 \to 1A_1 \to 10A_3 \to 105A_0 \to 105$$
$$A_0 \to 2A_2 \to 21A_0 \to 21$$
$$A_0 \to 3A_3 \to 35A_0 \to 35$$

   Conjecture: $[\![G]\!]$ is the language containing the decimal representation of $7\mathbb{N}$.

2. If we replace the rules $A_n \to dA_{(3n+d) \bmod 7}$ by the rules $A_n \to dA_{(10n+d) \bmod 7}$ we obtain the same language. Indeed, since $3 = 10 \bmod 7$, we have that $(3n+d) \bmod 7 = (10n+d) \bmod 7$, so this is exactly the same grammar.

   The idea is that the grammar reaches a derivation wA_n containing a non-terminal A_n only if, so far, w represents a number that is equal to n modulo 7:

$$\forall n \in \{0,\ldots,6\}, \forall w \in \Sigma^*, \text{ if } A_0 \to^* wA_n \text{ then } \langle w \rangle = n \mod 7$$

   where $\langle w \rangle \in \mathbb{N}$ is the number represented by $w \in \Sigma^*$.

   We can prove this fact formally by induction on the length of the derivation (or, equivalently, on the size of w).

   - base case: a derivation of length 1 is of the form $A_0 \to dA_{d \bmod 7}$, with $d \in \Sigma$. Then we trivially have $d = d \bmod 7$.
   - inductive case: let $n \in \mathbb{N}$ and assume that the property is true for derivations of length n. Now suppose we have a derivation of length $n+1$. Necessarily, it has the following form:

$$A_0 \to^* wA_n \to wdA_{10n+d \bmod 7}$$

     where $w \in \Sigma^*$ and $d \in \Sigma$. The first part of the derivation $A_0 \to^* wA_n$ has length n: by induction hypothesis, we obtain $\langle w \rangle = n \bmod 7$. Our goal is to show that $\langle wd \rangle = 10n + d \bmod 7$. But since $\langle wd \rangle = 10 * \langle w \rangle + d$, this follows directly from the induction hypothesis.

   Thus, we have proved the result for derivations of any length. Since A_0 is the only non-terminal which can be rewritten to the empty word, an accepting derivation will have the form $A_0 \to^* wA_0 \to w$, so every accepting word is such that $\langle w \rangle = 0 \bmod 7$.

3. We generalize the intuition to the grammar that recognize the decimal representation of $m\mathbb{N}$ for an arbitrary natural number m:

$$(\Sigma = \{0,\ldots,0\}, \mathscr{N} = \{A_0,\ldots,A_m\}, S = A_0, \mathscr{P} = \{A_0 \to \varepsilon\} \cup \bigcup_{n \in \{0,\ldots,m-1\}} \bigcup_{d \in \Sigma} \{A_n \to dA_{(10n+d) \bmod m}\})$$