CIS 570 — Introduction to Formal Language Theory — Fall 2007

Exercise Set 6

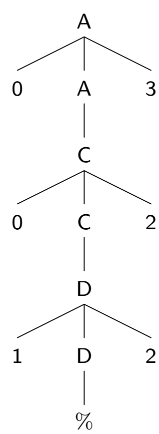
Model Answers

Exercise 1

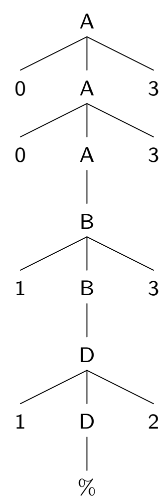
(a)

$$
\begin{aligned}
\mathsf{A} &\to \mathsf{0A3} \mid \mathsf{B} \mid \mathsf{C}\\
\mathsf{B} &\to \mathsf{1B3} \mid \mathsf{D}\\
\mathsf{C} &\to \mathsf{0C2} \mid \mathsf{D}\\
\mathsf{D} &\to \mathsf{1D2} \mid \%
\end{aligned}
$$

(b) Let pt_1 and pt_2 be the following parse trees:



(pt_1)



(pt_2)

Then pt_1 is a parse of 001223, and pt_2 is a parse of 00112333.

To check that our answers are correct, we place the description

```
{variables}
A, B, C, D
{start variable}
A
{productions}
A -> 0A3 | B | C;
B -> 1B3 | D;
C -> 0C2 | D;
D -> 1D2 | %
```

of G in the file `es6-ex1-gram`, and then load G into Forlan:

```
- val gram = Gram.input "es6-ex1-gram";
val gram = - : gram
```

Next, we put the description

```
A(0,
  A(C(0,
      C(D(1,
          D(%),
          2)),
      2)),
  3)
```

of pt_1 in the file `es6-ex1-pt1`, load pt_1 into Forlan, and check that it has the required properties:

```
- val pt1 = PT.input "es6-ex1-pt1";
val pt1 = - : pt
- Gram.validPT gram pt1;
val it = true : bool
- Sym.output("", PT.rootLabel pt1);
A
val it = () : unit
- Str.output("", PT.yield pt1);
001223
val it = () : unit
```

Finally, we put the description

```
A(0,
  A(0,
    A(B(1,
        B(D(1,
            D(%),
            2)),
        3)),
    3),
  3)
```

of pt_2 in the file `es6-ex1-pt2`, load pt_2 into Forlan, and check that it has the required properties:

```
- val pt2 = PT.input "es6-ex1-pt2";
val pt2 = - : pt
- Gram.validPT gram pt2;
val it = true : bool
- Sym.output("", PT.rootLabel pt2);
A
val it = () : unit
```

```
- Str.output("", PT.yield pt2);
00112333
val it = () : unit
```

(c) First, we define a testing function, and check that some strings in X are generated by G:

```
- fun generated s = Gram.generated gram (Str.fromString s);
val generated = fn : string -> bool
- map generated
=     ["%", "12", "03", "02", "13", "1122", "0033", "0123", "001233",
=      "00012233", "011333", "0111122333"];
val it = [true,true,true,true,true,true,true,true,true,true,true,true]
  : bool list
```

Second, we check that some strings that are not in X are not generated by G:

```
- map generated
=     ["0", "1", "2", "3", "30", "21", "31", "0213", "3120", "00123",
=      "01233", "01123", "01223", "00122", "00133", "00233", "00223",
=       "001133", "011123", "012233", "012333"];
val it =
  [false,false,false,false,false,false,false,false,false,false,false,false,
   false,false,false,false,false,false,false,false,false] : bool list
```

(d) Let

$$
\begin{aligned}
Y &= \{\, \mathsf{1}^j\mathsf{2}^k\mathsf{3}^l \mid j,k,l \in \mathbb{N} \text{ and } j = k + l \,\},\\
Z &= \{\, \mathsf{0}^i\mathsf{1}^j\mathsf{2}^k \mid i,j,k \in \mathbb{N} \text{ and } i + j = k \,\},\\
W &= \{\, \mathsf{1}^n\mathsf{2}^n \mid n \in \mathbb{N} \,\}.
\end{aligned}
$$

We will show that $\Pi_{\mathsf{A}} = X$, $\Pi_{\mathsf{B}} = Y$, $\Pi_{\mathsf{C}} = Z$, $\Pi_{\mathsf{D}} = W$.

Lemma ES6.1.1
$W \subseteq \Pi_{\mathsf{D}}$.

Proof. It will suffice to show that, for all $n \in \mathbb{N}$, $\mathsf{1}^n\mathsf{2}^n \in \Pi_{\mathsf{D}}$. We proceed by mathematical induction.

(Basis Step) Because $\mathsf{D} \rightarrow \% \in P$, we have that $\mathsf{1}^0\mathsf{2}^0 = \%\% = \% \in \Pi_{\mathsf{D}}$.

(Inductive Step) Suppose $n \in \mathbb{N}$, and assume the inductive hypothesis: $\mathsf{1}^n\mathsf{2}^n \in \Pi_{\mathsf{D}}$. Because $\mathsf{D} \rightarrow \mathsf{1D2} \in P$, it follows that $\mathsf{1}^{n+1}\mathsf{2}^{n+1} = \mathsf{1}(\mathsf{1}^n\mathsf{2}^n)\mathsf{2} \in \Pi_{\mathsf{D}}$. $\square$

Lemma ES6.1.2
$Y \subseteq \Pi_{\mathsf{B}}$.

Proof. First, we show a fact that we call (†) below: for all $x \in W$ and $n \in \mathbb{N}$, $\mathsf{1}^n x \mathsf{3}^n \in \Pi_{\mathsf{B}}$. Let $x \in W$. We use mathematical induction to show that, for all $n \in \mathbb{N}$, $\mathsf{1}^n x \mathsf{3}^n \in \Pi_{\mathsf{B}}$.

(Basis Step) By Lemma ES6.1.1, we have that $x \in W \subseteq \Pi_D$. Thus, since $\mathsf{B} \rightarrow \mathsf{D} \in P$, it follows that $\mathsf{1}^0 x \mathsf{3}^0 = \% x \% = x \in \Pi_B$.

(Inductive Step) Suppose $n \in \mathbb{N}$, and assume the inductive hypothesis: $1^n x 3^n \in \Pi_{\mathsf{B}}$. Thus, since $\mathsf{B} \to 1\mathsf{B}3 \in P$, it follows that $1^{n+1} x 3^{n+1} = 1(1^n x 3^n)3 \in \Pi_B$.

Now, we use (†) to show that $Y \subseteq \Pi_{\mathsf{B}}$. Suppose $w \in Y$, so that $w = 1^j 2^k 3^l$, for some $j, k, l \in \mathbb{N}$ such that $j = k + l$. Let $x = 1^k 2^k$. Since $x \in W$, we have that $w = 1^j 2^k 3^l = 1^{k+l} 2^k 3^l = 1^{l+k} 2^k 3^l = 1^l 1^k 2^k 3^l = 1^l x 3^l \in \Pi_{\mathsf{B}}$, by (†). □

Lemma ES6.1.3
$Z \subseteq \Pi_{\mathsf{C}}$.

Proof. First, we show a fact that we call (†) below: for all $x \in W$ and $n \in \mathbb{N}$, $0^n x 2^n \in \Pi_{\mathsf{C}}$. Let $x \in W$. We use mathematical induction to show that, for all $n \in \mathbb{N}$, $0^n x 2^n \in \Pi_{\mathsf{C}}$.

(Basis Step) By Lemma ES6.1.1, we have that $x \in W \subseteq \Pi_D$. Thus, since $\mathsf{C} \to \mathsf{D} \in P$, it follows that $0^0 x 2^0 = \% x \% = x \in \Pi_C$.

(Inductive Step) Suppose $n \in \mathbb{N}$, and assume the inductive hypothesis: $0^n x 2^n \in \Pi_{\mathsf{C}}$. Thus, since $\mathsf{C} \to 0\mathsf{C}2 \in P$, it follows that $0^{n+1} x 2^{n+1} = 0(0^n x 2^n)2 \in \Pi_C$.

Now, we use (†) to show that $Z \subseteq \Pi_{\mathsf{C}}$. Suppose $w \in Z$, so that $w = 0^i 1^j 2^k$, for some $i, j, k \in \mathbb{N}$ such that $i + j = k$. Let $x = 1^j 2^j$. Since $x \in W$, we have that $w = 0^i 1^j 2^k = 0^i 1^j 2^{i+j} = 0^i 1^j 2^{j+i} = 0^i 1^j 2^j 2^i = 0^i x 2^i \in \Pi_{\mathsf{C}}$, by (†). □

Lemma ES6.1.4
$X \subseteq \Pi_{\mathsf{A}}$.

Proof. First, we show a fact that we call (†) below: for all $x \in Y \cup Z$ and $n \in \mathbb{N}$, $0^n x 3^n \in \Pi_{\mathsf{A}}$. Let $x \in Y \cup Z$. We use mathematical induction to show that, for all $n \in \mathbb{N}$, $0^n x 3^n \in \Pi_{\mathsf{A}}$.

(Basis Step) Since $x \in Y \cup Z$, there are two cases to consider.

- Suppose $x \in Y$. By Lemma ES6.1.2, we have that $x \in Y \subseteq \Pi_B$. Thus, since $\mathsf{A} \to \mathsf{B} \in P$, it follows that $0^0 x 3^0 = \% x \% = x \in \Pi_A$.

- Suppose $x \in Z$. By Lemma ES6.1.3, we have that $x \in Z \subseteq \Pi_C$. Thus, since $\mathsf{A} \to \mathsf{C} \in P$, it follows that $0^0 x 3^0 = \% x \% = x \in \Pi_A$.

(Inductive Step) Suppose $n \in \mathbb{N}$, and assume the inductive hypothesis: $0^n x 3^n \in \Pi_{\mathsf{A}}$. Thus, since $\mathsf{A} \to 0\mathsf{A}3 \in P$, it follows that $0^{n+1} x 3^{n+1} = 0(0^n x 3^n)3 \in \Pi_A$.

Now, we use (†) to show that $X \subseteq \Pi_{\mathsf{A}}$. Suppose $w \in X$, so that $w = 0^i 1^j 2^k 3^l$, for some $i, j, k, l \in \mathbb{N}$ such that $i + j = k + l$. There are two cases to consider.

- Suppose $i \leq l$. Thus $l = i + m$, for some $m \in \mathbb{N}$. Let $x = 1^j 2^k 3^m$. Since $i + j = k + l = k + i + m = i + k + m$, we have that $j = k + m$, and thus that $x \in Y \subseteq Y \cup Z$. Hence, by (†), we have that $w = 0^i 1^j 2^k 3^l = 0^i 1^j 2^k 3^{m+i} = 0^i 1^j 2^k 3^m 3^i = 0^i x 3^i \in \Pi_{\mathsf{A}}$.

- Suppose $i > l$. Thus $i = l + m$, for some $m \in \mathbb{N} - \{0\}$. Let $x = 0^m 1^j 2^k$. Since $l + m + j = i + j = k + l$, we have that $m + j = k$, and thus that $x \in Z \subseteq Y \cup Z$. Hence, by (†), we have that $w = 0^i 1^j 2^k 3^l = 0^{l+m} 1^j 2^k 3^l = 0^l 0^m 1^j 2^k 3^l = 0^l x 3^l \in \Pi_{\mathsf{A}}$.

□

Lemma ES6.1.5
(A) $\Pi_{\mathsf{A}} \subseteq X$.

(B) $\Pi_{\mathsf{B}} \subseteq Y$.

(C) $\Pi_{\mathsf{C}} \subseteq Z$.

(D) $\Pi_{\mathsf{D}} \subseteq W$.

Proof. It will suffice to show that:

(A) For all $w \in \Pi_{\mathsf{A}}$, $w \in X$.

(B) For all $w \in \Pi_{\mathsf{B}}$, $w \in Y$.

(C) For all $w \in \Pi_{\mathsf{C}}$, $w \in Z$.

(D) For all $w \in \Pi_{\mathsf{D}}$, $w \in W$.

We proceed by induction on Π. There are nine productions to consider.

- $(\mathsf{A} \to \mathsf{0A3})$ Suppose $w \in \Pi_{\mathsf{A}}$, and assume the inductive hypothesis: $w \in X$. Then $w = \mathsf{0}^i\mathsf{1}^j\mathsf{2}^k\mathsf{3}^l$, for some $i, j, k, l \in \mathbb{N}$ such that $i+j = k+l$. Hence $\mathsf{0}w\mathsf{3} = \mathsf{0}\mathsf{0}^i\mathsf{1}^j\mathsf{2}^k\mathsf{3}^l\mathsf{3} = \mathsf{0}^{i+1}\mathsf{1}^j\mathsf{2}^k\mathsf{3}^{l+1} \in X$, since $(i+1)+j = (i+j)+1 = (k+l)+1 = k+(l+1)$.

- $(\mathsf{A} \to \mathsf{B})$ Suppose $w \in \Pi_{\mathsf{B}}$, and assume the inductive hypothesis: $w \in Y$. Then $w = \mathsf{1}^j\mathsf{2}^k\mathsf{3}^l$, for some $j, k, l \in \mathbb{N}$ such that $j = k+l$. Hence $w = \%w = \%\mathsf{1}^j\mathsf{2}^k\mathsf{3}^l = \mathsf{0}^0\mathsf{1}^j\mathsf{2}^k\mathsf{3}^l \in X$, since $0+j = j = k+l$.

- $(\mathsf{A} \to \mathsf{C})$ Suppose $w \in \Pi_{\mathsf{C}}$, and assume the inductive hypothesis: $w \in Z$. Then $w = \mathsf{0}^i\mathsf{1}^j\mathsf{2}^k$, for some $i, j, k \in \mathbb{N}$ such that $i+j = k$. Hence $w = w\% = \mathsf{0}^i\mathsf{1}^j\mathsf{2}^k\% = \mathsf{0}^i\mathsf{1}^j\mathsf{2}^k\mathsf{3}^0 \in X$, since $i+j = k = k+0$.

- $(\mathsf{B} \to \mathsf{1B3})$ Suppose $w \in \Pi_{\mathsf{B}}$, and assume the inductive hypothesis: $w \in Y$. Then $w = \mathsf{1}^j\mathsf{2}^k\mathsf{3}^l$, for some $j, k, l \in \mathbb{N}$ such that $j = k+l$. Hence $\mathsf{1}w\mathsf{3} = \mathsf{1}\mathsf{1}^j\mathsf{2}^k\mathsf{3}^l\mathsf{3} = \mathsf{1}^{j+1}\mathsf{2}^k\mathsf{3}^{l+1} \in Y$, since $j+1 = (k+l)+1 = k+(l+1)$.

- $(\mathsf{B} \to \mathsf{D})$ Suppose $w \in \Pi_{\mathsf{D}}$, and assume the inductive hypothesis: $w \in W$. Then $w = \mathsf{1}^n\mathsf{2}^n$, for some $n \in \mathbb{N}$. Hence $w = w\% = \mathsf{1}^n\mathsf{2}^n\% = \mathsf{1}^n\mathsf{2}^n\mathsf{3}^0 \in Y$, since $n = n+0$.

- $(\mathsf{C} \to \mathsf{0C2})$ Suppose $w \in \Pi_{\mathsf{C}}$, and assume the inductive hypothesis: $w \in Z$. Then $w = \mathsf{0}^i\mathsf{1}^j\mathsf{2}^k$, for some $i, j, k \in \mathbb{N}$ such that $i+j = k$. Hence $\mathsf{0}w\mathsf{2} = \mathsf{0}\mathsf{0}^i\mathsf{1}^j\mathsf{2}^k\mathsf{2} = \mathsf{0}^{i+1}\mathsf{1}^j\mathsf{2}^{k+1} \in Z$, since $(i+1)+j = (i+j)+1 = k+1$.

- $(\mathsf{C} \to \mathsf{D})$ Suppose $w \in \Pi_{\mathsf{D}}$, and assume the inductive hypothesis: $w \in W$. Then $w = \mathsf{1}^n\mathsf{2}^n$, for some $n \in \mathbb{N}$. Hence $w = \%w = \%\mathsf{1}^n\mathsf{2}^n = \mathsf{0}^0\mathsf{1}^n\mathsf{2}^n \in Z$, since $0+n = n$.

- $(\mathsf{D} \to \mathsf{1D2})$ Suppose $w \in \Pi_{\mathsf{D}}$, and assume the inductive hypothesis: $w \in W$. Then $w = \mathsf{1}^n\mathsf{2}^n$, for some $n \in \mathbb{N}$. Hence $\mathsf{1}w\mathsf{2} = \mathsf{1}\mathsf{1}^n\mathsf{2}^n\mathsf{2} = \mathsf{1}^{n+1}\mathsf{2}^{n+1} \in W$.

- $(\mathsf{D} \to \%)$ We have that $\% = \%\% = \mathsf{1}^0\mathsf{2}^0 \in W$.

$\square$

By Lemmas ES6.1.4 and ES6.1.5(A), we have that $X \subseteq \Pi_{\mathsf{A}} \subseteq X$, so that $L(G) = \Pi_{\mathsf{A}} = X$.

(e) Suppose, toward a contradiction, that X is regular. Thus there is an $n \in \mathbb{N}$ with the property of the Pumping Lemma, where X has been substituted for L. Let $z = \mathsf{0}^n\mathsf{3}^n$. Since $z = \mathsf{0}^n\mathsf{3}^n = \mathsf{0}^n\%\%\mathsf{3}^n = \mathsf{0}^n\mathsf{1}^0\mathsf{2}^0\mathsf{3}^n$ and $n + 0 = 0 + n$, we have that $z \in X$. And, $|z| = 2n \geq n$. Thus, by the property of the Pumping Lemma, we have that there are $u, v, w \in \mathbf{Str}$ such that $z = uvw$ and

(1) $|uv| \leq n$;

(2) $v \neq \%$; and

(3) $uv^iw \in X$, for all $i \in \mathbb{N}$.

Since $\mathsf{0}^n\mathsf{3}^n = z = uvw$, (1) tells us that there are $i, j, k \in \mathbb{N}$ such that

$$u = \mathsf{0}^i, \quad v = \mathsf{0}^j, \quad w = \mathsf{0}^k\mathsf{3}^n, \quad i + j + k = n.$$

By (2), we have that $j \geq 1$, and thus that $i + k = n - j < n$. By (3), we have that

$$\mathsf{0}^{i+k}\mathsf{3}^n = \mathsf{0}^i\mathsf{0}^k\mathsf{3}^n = uw = u\%w = uv^0w \in X.$$

Hence $\mathsf{0}^{i+k}\mathsf{3}^n = \mathsf{0}^{i'}\mathsf{1}^{j'}\mathsf{2}^{k'}\mathsf{3}^{l'}$ for some $i', j', k', l' \in \mathbb{N}$ such that $i' + j' = k' + l'$, so that $i + k = i'$, $0 = j'$, $0 = k'$ and $n = l'$. Thus $i + k = i + k + 0 = i' + j' = k' + l' = 0 + n = n$—contradiction. Thus X is not regular.

Exercise 2

(a)

```
(* val zero : sym *)

val zero = Sym.fromString "0";

(* val one : sym *)

val one = Sym.fromString "1";

(* val minAndRen : dfa -> dfa *)

val minAndRen =
      DFA.renameStatesCanonically o DFA.minimize o DFA.renameStatesCanonically;

(* val efaToDFA : efa -> dfa *)

val efaToDFA = nfaToDFA o efaToNFA;

(* val allStrDFA : dfa

   allStrDFA accepts {0, 1}* *)
```

```
val allStrDFA =
      minAndRen(efaToDFA(EFA.closure(EFA.union(EFA.fromSym zero,
                                               EFA.fromSym one))));

(* val swapRel : sym_rel *)

val swapRel = SymRel.fromString "(0, 1), (1, 0)";

(* val zeroAllPrefPosEFA : int -> efa

   if n >= 0, then zeroAllPrefPosEFA n returns an efa that accepts all
   w in {0, 1}* such that diff w = 0 and all prefixes of w have diff's
   between 0 and n *)

fun zeroAllPrefPosEFA 0 = EFA.emptyStr
  | zeroAllPrefPosEFA n =
      EFA.closure(EFA.concat(EFA.fromSym one,
                             EFA.concat(zeroAllPrefPosEFA(n - 1),
                                        EFA.fromSym zero)));

(* val zeroAllPrefPosDFA : int -> dfa

   if n >= 0, then zeroAllPrefPosDFA n returns a dfa that accepts
   all w in {0, 1}* such that diff w = 0 and all prefixes of w
   have diff's between 0 and n *)

fun zeroAllPrefPosDFA n = minAndRen(efaToDFA(zeroAllPrefPosEFA n));

(* val justBadPosEFA : int -> efa

   if n >= 0, then justBadPosEFA n returns an efa that accepts all w
   in {0, 1}* such that diff w = n + 1 and all proper prefixes of w
   have diff's between 0 and n *)

fun justBadPosEFA 0 = EFA.fromSym one
  | justBadPosEFA n =
      EFA.concat(injDFAToEFA(zeroAllPrefPosDFA n),
                 EFA.concat(EFA.fromSym one, justBadPosEFA(n - 1)));

(* val justBadPosDFA : int -> dfa

   if n >= 0, then justBadPosDFA n returns a dfa that accepts all w in
   {0, 1}* such that diff w = n + 1 and all proper prefixes of w have
   diff's between 0 and n *)

fun justBadPosDFA n = minAndRen(efaToDFA(justBadPosEFA n));
```

```
(* val justBadNegDFA : int -> dfa

   if n >= 0, then justBadNegDFA n returns a dfa that accepts all w in
   {0, 1}* such that diff w = ~(n + 1) and all proper prefixes of w
   have diff's between 0 and ~n *)

fun justBadNegDFA n = DFA.renameAlphabet(justBadPosDFA n, swapRel);

(* val justBadPosOrNegDFA : int -> dfa

   if n >= 0, then justBadPosOrNegDFA n returns a dfa accepting all w
   in {0, 1}* such that either:

     diff w = n + 1 and all proper prefixes of w have diff's between 0
     and n; or

     diff w = ~(n + 1) and all proper prefixes of w have diff's
     between 0 and ~n *)

fun justBadPosOrNegDFA n =
      minAndRen(efaToDFA(EFA.union(injDFAToEFA(justBadPosDFA n),
                                   injDFAToEFA(justBadNegDFA n))));

(* val someSubBadEFA : int -> efa

   if n >= 0, then someSubBadEFA n returns an efa that accepts all w in
   {0, 1}* such that some substring of w has a diff that is < ~n or > n *)

fun someSubBadEFA n =
      EFA.concat(injDFAToEFA allStrDFA,
                 EFA.concat(injDFAToEFA(justBadPosOrNegDFA n),
                            injDFAToEFA allStrDFA));

(* val someSubBadDFA : int -> dfa

   if n >= 0, then someSubBadDFA n returns a dfa that accepts all w in
   {0, 1}* such that some substring of w has a diff that is < ~n or > n *)

fun someSubBadDFA n = minAndRen(efaToDFA(someSubBadEFA n));

(* val allSubGoodDFA : int -> dfa

   if n >= 0, then allSubGoodDFA n returns a minimized dfa that
   accepts all w in {0, 1}* such that all substrings of w have diff's
   between ~n and n *)

fun allSubGoodDFA n = minAndRen(DFA.minus(allStrDFA, someSubBadDFA n));
```

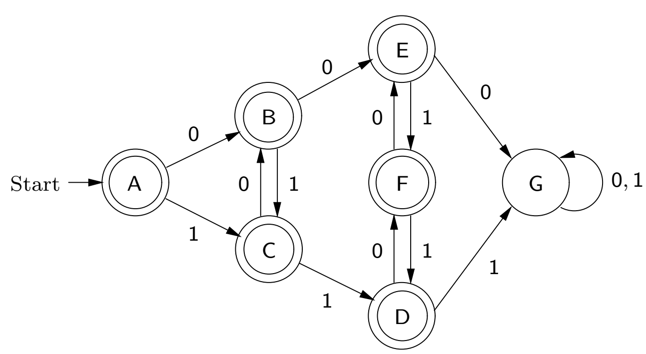
(b) First, we put the solution to Part (a) in the file `all-sub-good.sml`. Then we invoke Forlan and load this file:

```
- use "all-sub-good.sml";
[opening all-sub-good.sml]
val zero = - : sym
val one = - : sym
val minAndRen = fn : dfa -> dfa
val efaToDFA = fn : efa -> dfa
val allStrDFA = - : dfa
val swapRel = - : sym_rel
val zeroAllPrefPosEFA = fn : int -> efa
val zeroAllPrefPosDFA = fn : int -> dfa
val justBadPosEFA = fn : int -> efa
val justBadPosDFA = fn : int -> dfa
val justBadNegDFA = fn : int -> dfa
val justBadPosOrNegDFA = fn : int -> dfa
val someSubBadEFA = fn : int -> efa
val someSubBadDFA = fn : int -> dfa
val allSubGoodDFA = fn : int -> dfa
val it = () : unit
```

Next, we generate and display a DFA accepting **AllSubGood**(2):

```
- val dfa1 = allSubGoodDFA 2;
val dfa1 = - : dfa
- DFA.output("", dfa1);
{states}
A, B, C, D, E, F, G
{start state}
A
{accepting states}
A, B, C, D, E, F
{transitions}
A, 0 -> B; A, 1 -> C; B, 0 -> E; B, 1 -> C; C, 0 -> B; C, 1 -> D; D, 0 -> F;
D, 1 -> G; E, 0 -> G; E, 1 -> F; F, 0 -> E; F, 1 -> D; G, 0 -> G; G, 1 -> G
val it = () : unit
```

Here is a drawing of `dfa1`:



Finally, we generate and display a DFA accepting **AllSubGood**(3):

```
- val dfa2 = allSubGoodDFA 3;
val dfa2 = - : dfa
- DFA.output("", dfa2);
{states}
A, B, C, D, E, F, G, H, I, J, K
{start state}
A
{accepting states}
A, B, C, D, E, F, G, H, I, J
{transitions}
A, 0 -> B; A, 1 -> C; B, 0 -> F; B, 1 -> C; C, 0 -> B; C, 1 -> G; D, 0 -> E;
D, 1 -> J; E, 0 -> I; E, 1 -> D; F, 0 -> I; F, 1 -> H; G, 0 -> H; G, 1 -> J;
H, 0 -> F; H, 1 -> G; I, 0 -> K; I, 1 -> E; J, 0 -> D; J, 1 -> K; K, 0 -> K;
K, 1 -> K
val it = () : unit
```

Here is a drawing of `dfa2`: