HW3: Exercise 2.3 [Magherna]

Lexical Analysis, building on Dr. Fenner's guest lecture, and following Ch. 2 [Magherna].

Definition 2.1 A Non-deterministic Finite-state Automaton (NFA) consists of:

- $S$ - set of states,
- $S_0 \in S$ - starting state
F ⊆ S - (set of) accepting states

T - set of transitions, which

- connect states
- are labeled with either a symbol from the alphabet (Σ) of the grammar, or ε

A transition from state s to state t on symbol c is written set
Example: (Fig 2.3)

\[
S = \{1, 2, 3\} \\
S_0 = \{1\} \\
F = \{3\} \\
T = \{1 \varepsilon, a, a^2, 1, 2, 3, 1^3\} ...
\]

By the way, this NFA recognizes \( (a+b)^* \) whose language is the set of the following strings: \( b \) along any nonempty sequence of \( a \)'s, and \( a \)
(possibly empty) sequence of a's followed by b.

A regexp can be converted into an NFA compositionally, i.e., from conversions of the regexp's subexpressions. The rules for the construction are in Fig. 2.4.

Example. Build an NFA for \([0-9]^*\).

\[
\begin{align*}
0 \Rightarrow & \quad -\bigcirc -0 - \ldots -9 \Rightarrow \\
& \quad -\bigcirc -1
\end{align*}
\]
\[(0 - 9)^* \text{ (and no more!)}\]

There are some "optimized" (but not optimal) rules in Figure 2.6. Using the optimized
rules to construct $\{0-9\}^+$, we get:

There is a smaller NFA for $\{0-9\}^+$:
Deterministic Finite-state Automata (DFA) are NFA's with two (additional) restrictions:
- there are no ε-transitions
- there may not be two identically labeled transitions out of the same state.
The transition relation \( \delta \) in DFAs is a (partial) function, which we will move.
move(s, c) is the state (if any) reached from s by a transition on symbol c

A DFA can be implemented by two tables; one represents the accepting states (an array of Booleans). The other represents the move function.
The generic entry \((s, t)\) contains symbol \(c\) if there is a transition from \(s\) to \(t\) on symbol \(c\).

Maybe surprisingly, DFA and NFA have the same expressive power—they both recognize regular
languages. One way of this equivalence is obvious, b/c every DFA is an NFA. To prove the other way, one should prove the correctness of the algorithm to convert an NFA to a DFA in section 2.6 [Hopcroft]. To deal with ε-transitions, we introduce a notion.

Definition 2.2 (ε-closure). Let M be a set of
NFA states. \( \varepsilon \)-closure \( (M) \) is the least (by set inclusion) solution to the set equation

\[
\varepsilon \text{-closure} \ (M) = M \cup \{ t \mid s \in \varepsilon \text{-closure} \ (M) \text{ and } s \varepsilon t \in T \}\]

where \( T \) is the set of transitions of the NFA.

Appendix A.4 of Morgenstern's book has a detailed
Discussion of set equations.

A set equation has the form \( X = F(X) \).

In our case, we are interested in

\[ E - \text{close} (M), \text{ so} \]

\[ X = M \cup \{ t \mid s \in X \text{ and } s^2 + t \in T \} \].

So, if we define \( F_n \) to be

\[ F_n (X) = M \cup \{ t \mid s \in X \text{ and } s^2 + t \in T \} \]
Then a solution to $X = F_{\pi}(X)$ will be

3-closure ($\Pi$).

$F$ is monotonic when, if $X \leq Y$, then $F(X) \leq F(Y)$.

Solution technique for $X = F(X)$, where $F$ is monotonic.

1. Guess $X = \phi$. If $\phi = F(\phi)$, done

2. Otherwise, try $X = F(\phi)$. If true, done
3. Otherwise, try $F(F(\phi))$. Repeat. An algorithm to convert NFA's to DFA's will be given next time.