Week 8

Sunday, March 13, 2022 4:43 PM

3.7 From Regular Expressions to Automata

Regular Expression is the notation for describing lexical analyzers
↳ implementation of this pattern recognition software requires a simulation of a DFA or NFA.
↓
since DFA is simpler, we need to convert NFA → DFA's that accept the same language.

Convert NFA to DFA: technique known as The Subset Construction
→ General Idea: Each state of the constructed DFA corresponds to a set of NFA states.

→ After reading input $a_1 a_2 \dots a_n$, the DFA is in that state which corresponds to the set of states that the NFA can reach: from its start state onto following a path $a_1 a_2 \dots a_n$.

<u>Algorithm</u>:

Input: an NFA, N.
Output: A DFA D, accepting the same language as N.
Method:
→ Construct a Transition Table Dtran for D.
↳ Each state of D is a set of NFA states
↳ D will simulate "in parallel" all possible moves N can make w/a given input.

→ The 1st problem is to deal with ε transitions
→ The table shows functions that describe the basic computations <u>on N states</u> needed in the algorithm.

↳ s is a single state of N.
↳ T is a set of states of N.

OPERATION	DESCRIPTION
ϵ-*closure*(s)	Set of NFA states reachable from NFA state s on ϵ-transitions alone.
ϵ-*closure*(T)	Set of NFA states reachable from some NFA state s in set T on ϵ-transitions alone; $= \cup_{s \text{ in } T}\ \epsilon\text{-}closure(s)$.
move(T, a)	Set of NFA states to which there is a transition on input symbol a from some state s in T.

s_0: start state.

1. Before reading the first symbol from input, N can be in any of the states of ε-closure (s_0)
2. Suppose N can be in set of states T after reading string x.
   ↳ If it next reads input a, N can immediately go to any of the states in move(T, a)
   ↳ But also after reading a, it may also make ε-transitions ∴ N could be in any state ε-closure (move (T, a)) after input xa.
3. The construction of set Dstates and its transition function Dtran, is shown below:

```
initially, ε-closure(s0) is the only state in Dstates, and it is unmarked;
while ( there is an unmarked state T in Dstates ) {
        mark T;
        for ( each input symbol a ) {
                U = ε-closure(move(T, a));
                if ( U is not in Dstates )
                        add U as an unmarked state to Dstates;
                Dtran[T, a] = U;
        }
}
```

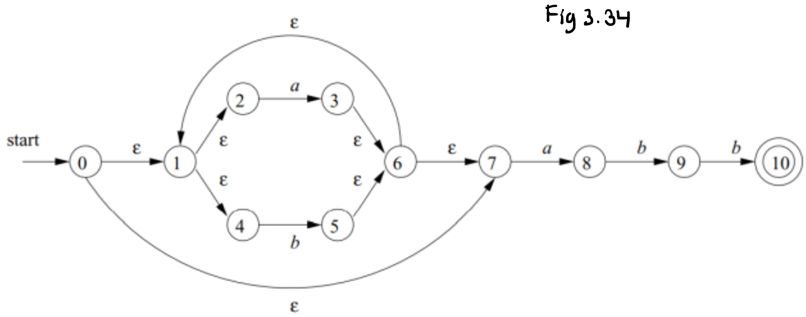
4. The start state of D is ε-closure (S_0)
   ↳ the accepting states of D are all those sets of N's states that include at least one accepting state of N.

5. We only need to show how ε-closure (T) is computed for any set of NFA states T. This process is a graph search from a set of states. Imagine that only ε-labeled edges are available in the graph:

```
push all states of T onto stack;
initialize ε-closure(T) to T;
while ( stack is not empty ) {
        pop t, the top element, off stack;
        for ( each state u with an edge from t to u labeled ε )
                if ( u is not in ε-closure(T) ) {
                        add u to ε-closure(T);
                        push u onto stack;
                }
}
```

Example, let's apply algorithm to NFA:

Fig 3.34



1. At the start state A of the equivalent DFA is ε-closure (0), or
$A = \{0, 1, 2, 4, 7\}$: exactly the states reachable from S_0 via a path all of whose edges have label ε.
↳ a path can have $\emptyset$ edges, so state 0 is reachable from itself by an ε path.

→ The input alphabet is $\Sigma = \{a, b\}$, So

2. Mark A and compute

Dtran $[A, a] = \varepsilon$-closure $(move(A, a))$

Dtran $[A, b] = \varepsilon$-closure $(move(A, b))$

3. Among A states $\{0, 1, 2, 4, 7\}$, only 2 and 7 have transitions on a, to 3 and 8, respectively.

↳ Thus move $(A, a) = \{3, 8\}$

ε-closure $(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\}$

So we conclude

$B =$ Dtran $[A, a] = \varepsilon$-closure $(move(A, a)) = \varepsilon$-closure $(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\}$

4. Computing Dtran $[A, b]$: Among states in A, only 4 has a transition on b, and it goes to 5. Thus,

$C =$ Dtran $[A, b] = \varepsilon$-closure $(\{5\}) = \{1, 2, 4, 5, 6, 7\}$

$C = Dtran[A, b] = \epsilon\text{-closure}(\{5\}) = \{1,2,4,5,6,7\}$

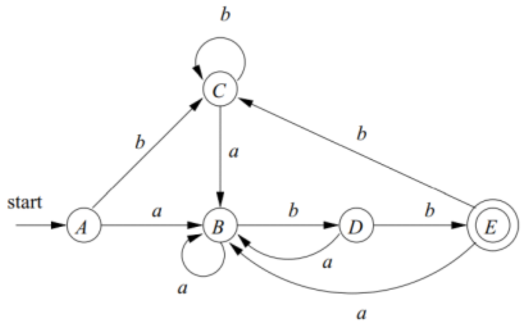
• A contains 0: start state

• E has 10: accepting state

NFA State	DFA State	a	b
$\{0,1,2,4,7\}$	A	B	C
$\{1,2,3,4,6,7,8\}$	B	B	D
$\{1,2,4,5,6,7\}$	C	B	C
$\{1,2,4,5,6,7,9\}$	D	B	E
$\{1,2,4,5,6,7,10\}$	E	B	C

Transition Table Dtran for DFA "D".

5. Continue this process with the unmarked sets D and C
   - ↳ eventually all the states of the DFA are marked.
     - ↳ guaranteed, since there are 2^{11} different subsets of a set of 11 NFA states.



↳ Result of applying the Subset Construction to Fig 3.34