Compiler Design

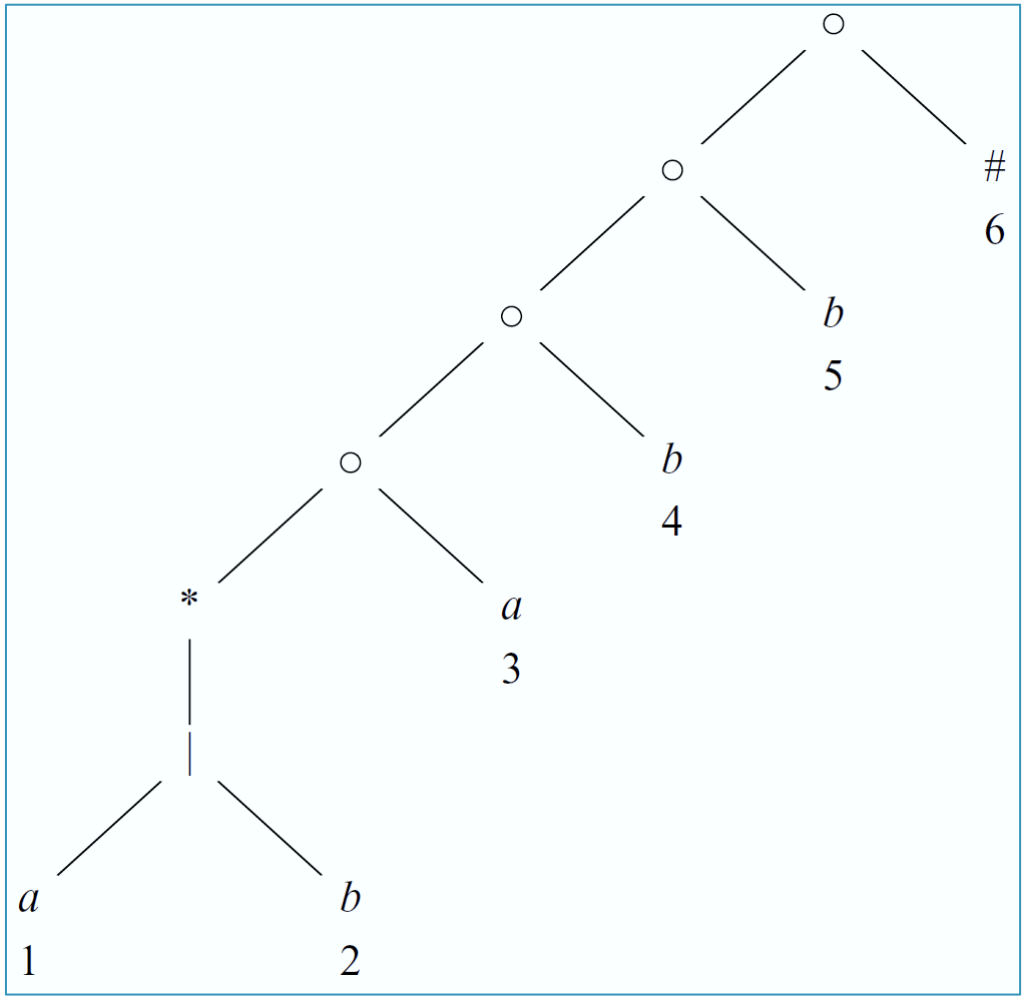
Fatemeh Deldar

Isfahan University of Technology

1403-1404

Converting a Regular Expression Directly to a DFA

- **Example:** Regular expression $(a|b)^*abb$
- **Syntax tree for $(a|b)^*abb\#$**
  - To each leaf not labeled ϵ, we attach a unique integer as the position of the leaf



Converting a Regular Expression Directly to a DFA

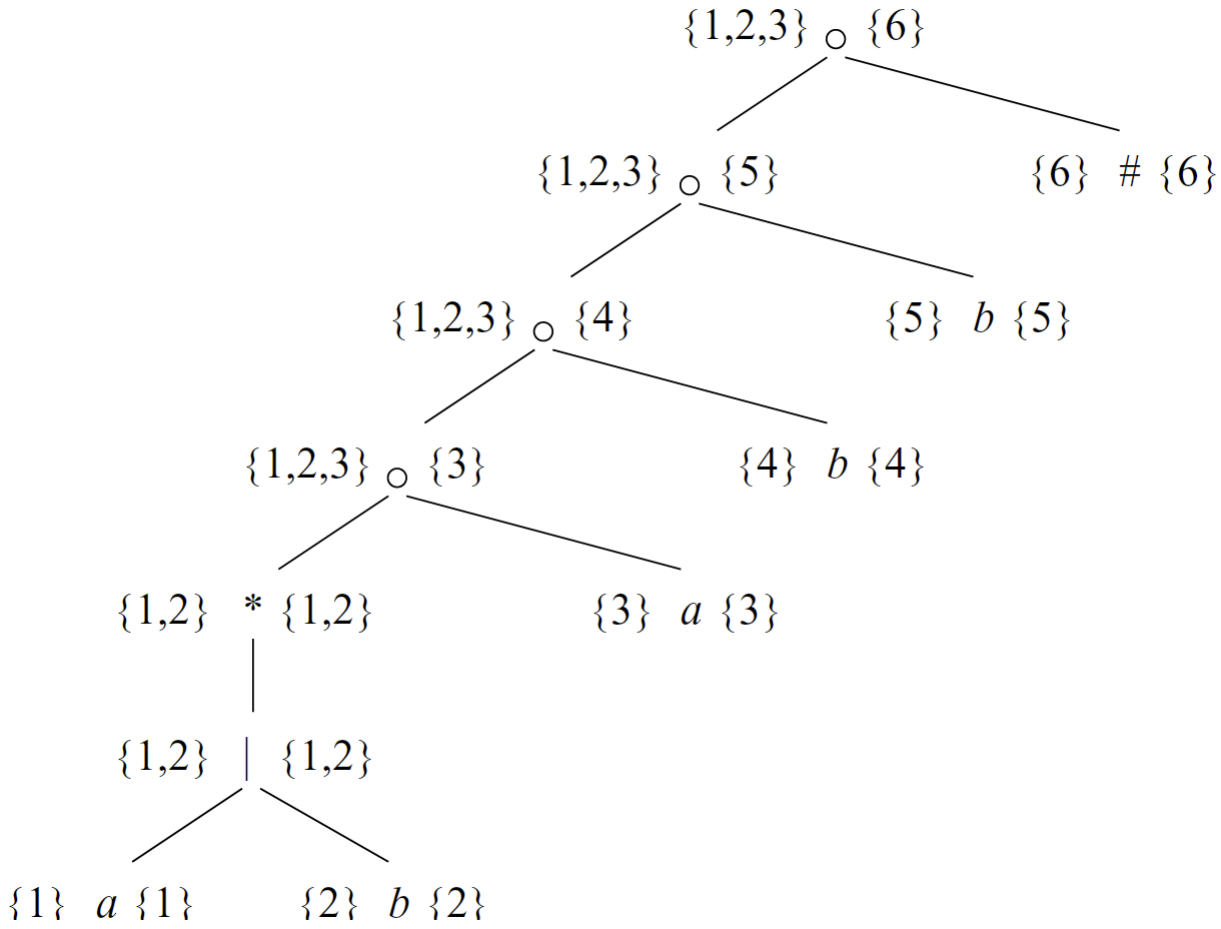
- **Functions Computed From the Syntax Tree**
  - ***nullable(n)*** is true for a syntax-tree node n if and only if the subexpression represented by n has ϵ in its language
  - ***firstpos(n)*** is the set of positions in the subtree rooted at n that correspond to the first symbol of at least one string in the language of the subexpression rooted at n
  - ***lastpos(n)*** is the set of positions in the subtree rooted at n that correspond to the last symbol of at least one string in the language of the subexpression rooted at n
  - ***followpos(p)***, for a position p, is the set of positions q in the entire syntax tree that can come after p

Converting a Regular Expression Directly to a DFA

NODE n	$nullable(n)$	$firstpos(n)$
A leaf labeled ϵ	**true**	$\emptyset$
A leaf with position i	**false**	$\{i\}$
An or-node $n = c_1 \mid c_2$	$nullable(c_1)$ **or** $nullable(c_2)$	$firstpos(c_1) \cup firstpos(c_2)$
A cat-node $n = c_1 c_2$	$nullable(c_1)$ **and** $nullable(c_2)$	**if** ($nullable(c_1)$) $firstpos(c_1) \cup firstpos(c_2)$ **else** $firstpos(c_1)$
A star-node $n = c_1{}^*$	**true**	$firstpos(c_1)$

Converting a Regular Expression Directly to a DFA

- **Example**
  - ***firstpos*** and ***lastpos*** for nodes in the syntax tree for $(a|b)^*abb\#$



Converting a Regular Expression Directly to a DFA

- **Computing followpos**
  1. If n is a cat-node with left child c_1 and right child c_2, then for every position i in $lastpos(c_1)$, all positions in $firstpos(c_2)$ are in $followpos(i)$
  2. If n is a star-node, and i is a position in $lastpos(n)$, then all positions in $firstpos(n)$ are in $followpos(i)$

- **Example**

POSITION n	$followpos(n)$
1	$\{1, 2, 3\}$
2	$\{1, 2, 3\}$
3	$\{4\}$
4	$\{5\}$
5	$\{6\}$
6	$\emptyset$

Converting a Regular Expression Directly to a DFA

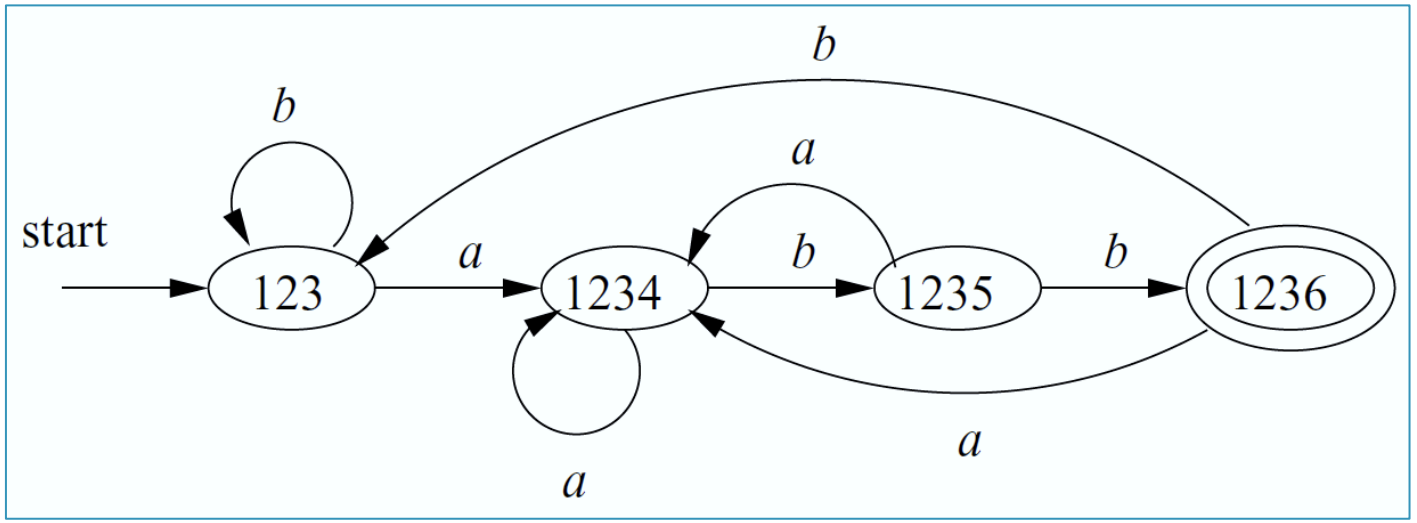
- **Algorithm**
  1. Construct a syntax tree T from the augmented regular expression $(r)\#$
  2. Compute *nullable*, *firstpos*, *lastpos*, and *followpos* for T
  3. Construct *Dstates*, the set of states of DFA D, and *Dtran*, the transition function for D, using the following algorithm

```
initialize Dstates to contain only the unmarked state firstpos(n0),
        where n0 is the root of syntax tree T for (r)#;
while ( there is an unmarked state S in Dstates ) {
        mark S;
        for ( each input symbol a ) {
                let U be the union of followpos(p) for all p
                        in S that correspond to a;
                if ( U is not in Dstates )
                        add U as an unmarked state to Dstates;
                Dtran[S, a] = U;
        }
}
```

Converting a Regular Expression Directly to a DFA

- **Example**

$(a|b)^*abb\#$



وضعیت شروع:
ریشه **firstpos**

وضعیت نهایی:
وضعیت‌هایی که در آن‌ها مکان کاراکتر # آمده باشد.