Direct Conversion of RE to DFA

From Regular Expression to DFA Directly (Algorithm)

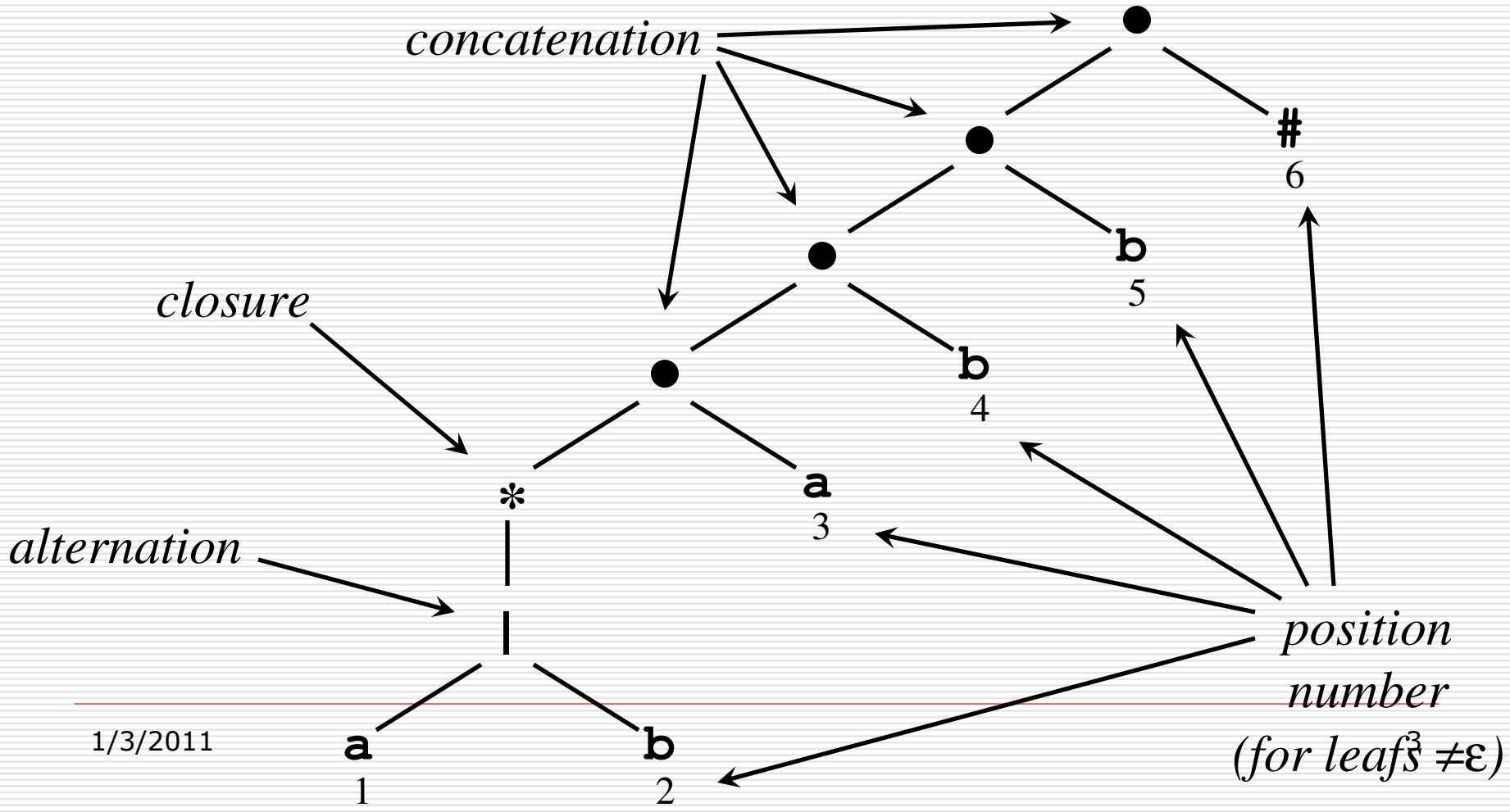
- Augment the regular expression *r* with a special end symbol # to make accepting states important
  r → *(r)#*

- Construct a syntax tree for *(r)#*
  Alphabets → leaf node
  Operators → inner node

- Number each alphabet including #

- Traverse the tree to construct functions *nullable*, *firstpos*, *lastpos*, and *followpos*

Syntax Tree of (a|b)*abb#



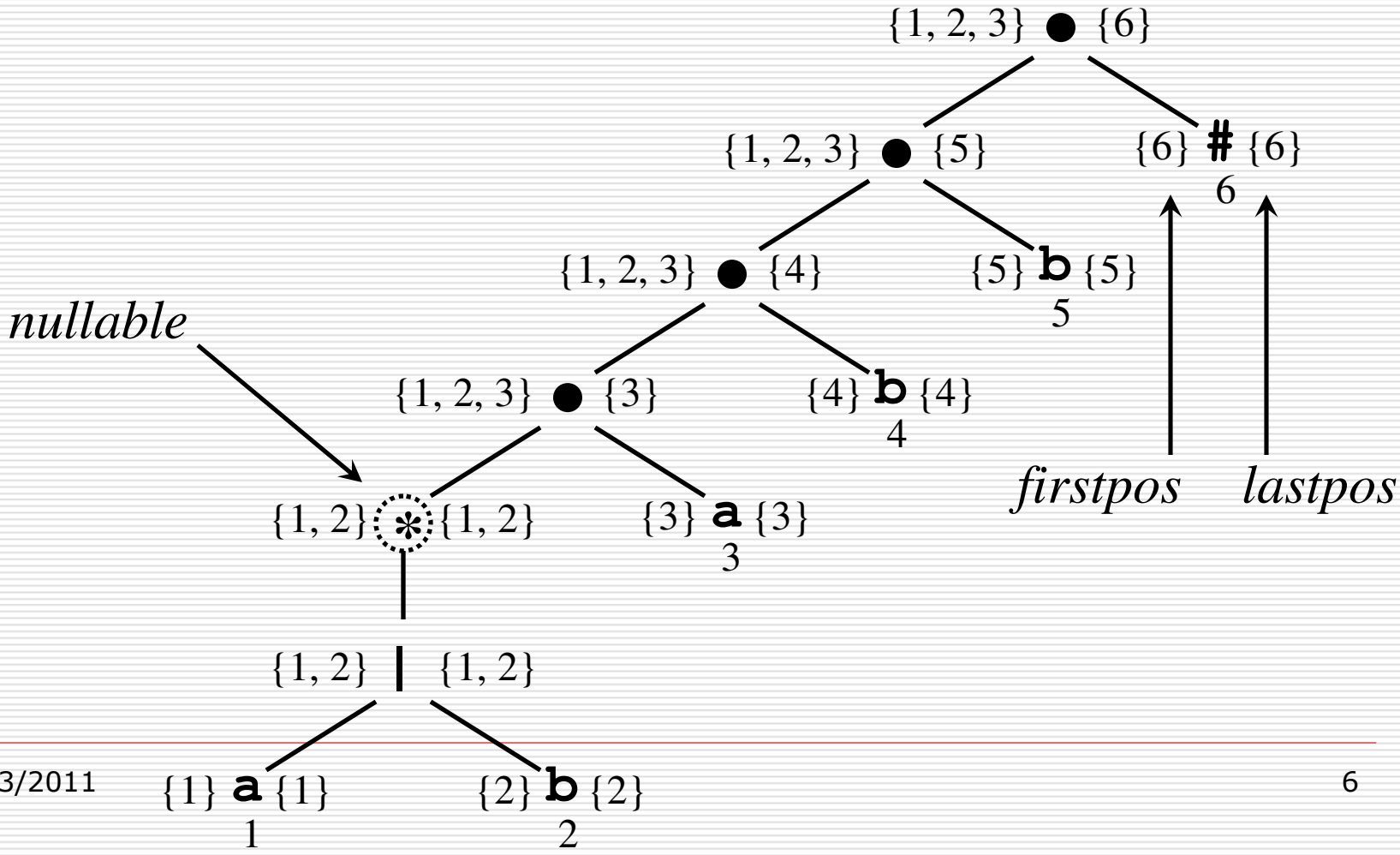
Annotating the Tree

- ***nullable*(*n*):** the subtree at node *n* generates languages including the empty string
- ***firstpos*(*n*):** set of positions that can match the first symbol of a string generated by the subtree at node *n*
- ***lastpos*(*n*):** the set of positions that can match the last symbol of a string generated be the subtree at node *n*
- ***followpos*(*i*):** the set of positions that can follow position *i* in the tree

Annotating the Tree Cont…

Node *n*	*nullable*(*n*)	*firstpos*(*n*)	*lastpos*(*n*)
Leaf ε	true	∅	∅
Leaf *i*	false	{*i*}	{*i*}
	*nullable*(c_1) or *nullable*(c_2)	*firstpos*(c_1) ∪ *firstpos*(c_2)	*lastpos*(c_1) ∪ *lastpos*(c_2)
	nullable(c_1) and *nullable*(c_2)	**if** *nullable*(c_1) **then** *firstpos*(c_1) ∪ *firstpos*(c_2) **else** *firstpos*(c_1)	**if** *nullable*(c_2) **then** *lastpos*(c_1) ∪ *lastpos*(c_2) **else** *lastpos*(c_2)
	true	*firstpos*(c_1)	*lastpos*(c_1)

Syntax Tree of (a|b)*abb#



followpos

for each node n in the tree **do**
    if n is a cat-node with left child c_1 and right child c_2 **then**
        for each i in $lastpos(c_1)$ **do**
            $followpos(i) := followpos(i) \cup firstpos(c_2)$
        end do
    else if n is a star-node
        for each i in $lastpos(n)$ **do**
            $followpos(i) := followpos(i) \cup firstpos(n)$
        end do
    end if
end do

Algorithm

s_0 := *firstpos*(*root*) where *root* is the root of the syntax tree
Dstates := {s_0} and is unmarked
while there is an unmarked state *T* in *Dstates* **do**
    mark *T*
    for each input symbol $a \in \Sigma$ **do**
        let *U* be the set of positions that are in *followpos*(*p*)
            for some position *p* in *T*,
            such that the symbol at position *p* is *a*
        if *U* is not empty and not in *Dstates* **then**
            add *U* as an unmarked state to *Dstates*
        end if
        Dtran[*T*,*a*] := *U*
    end do
end do

Example

Node	*followpos*
1	{1, 2, 3}
2	{1, 2, 3}
3	{4}
4	{5}
5	{6}
6