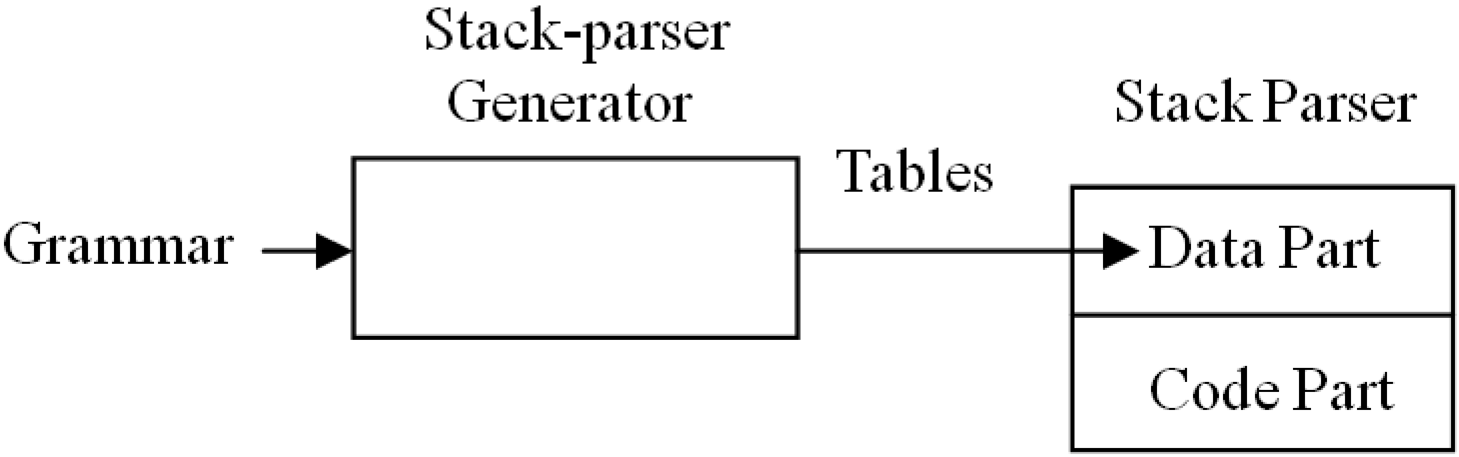
Chapter 8

Table-Driven Stack Parser

Parser generator



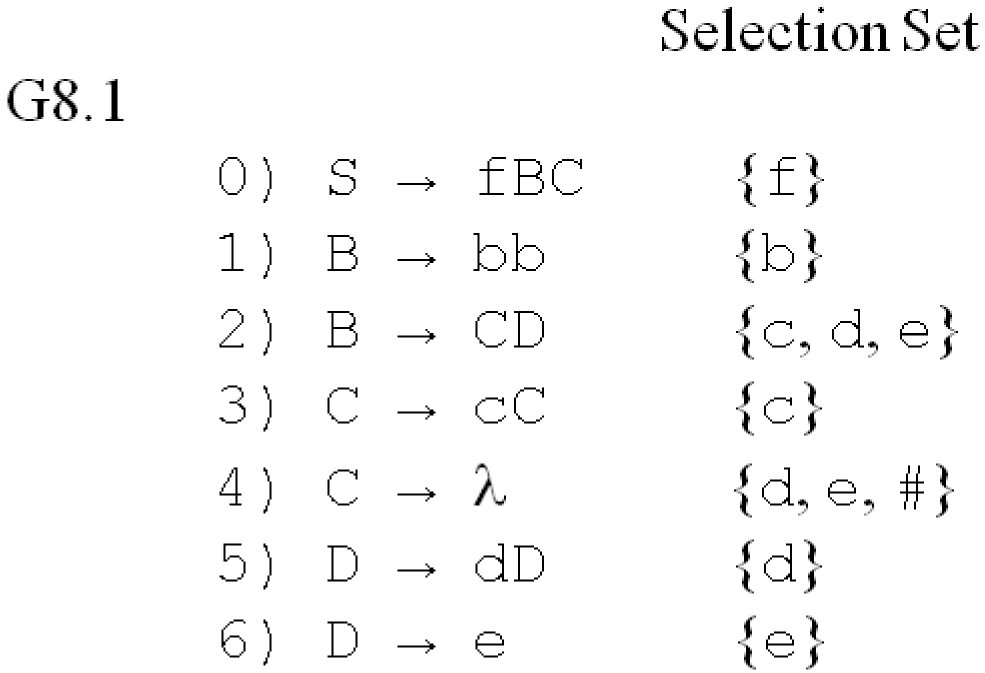
Uniform stack parser

Handle all productions in the same way:

1. pop
2. push entire right side

If the top of the stack is a terminal and it matches the current input, then pop and advance.

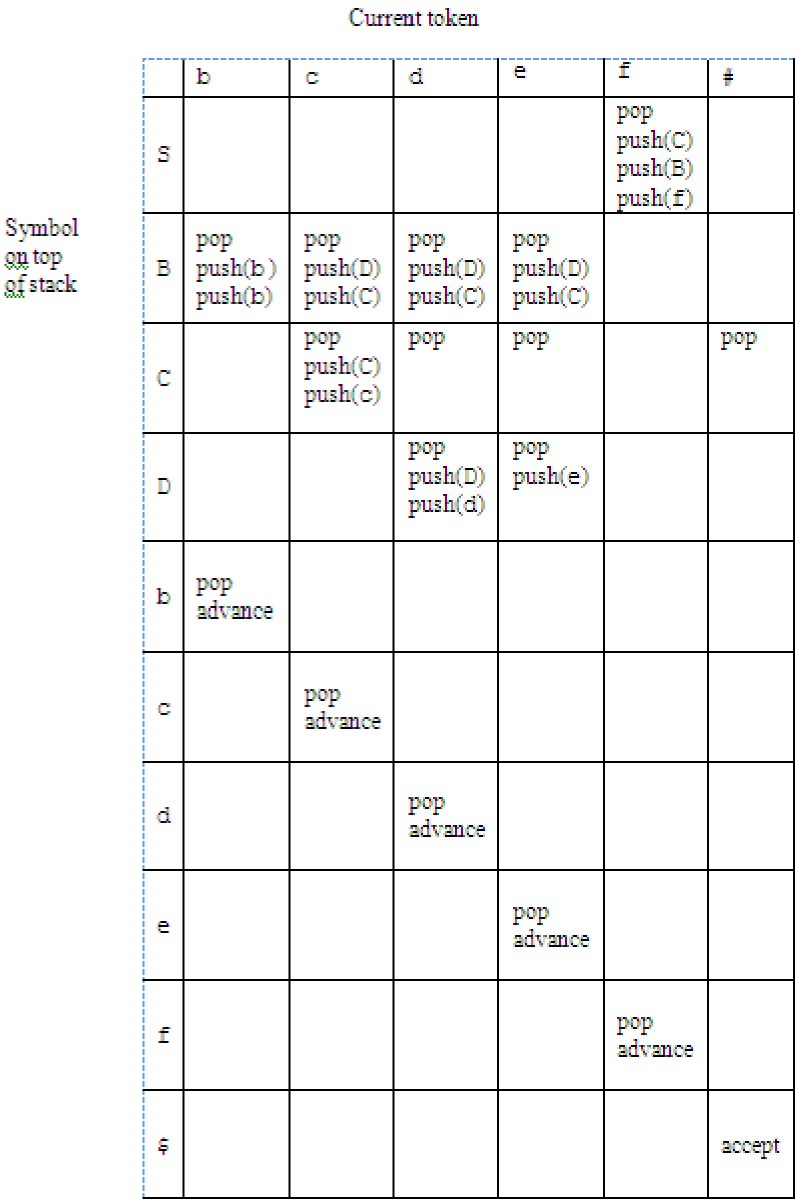
Construct uniform parser for



G8.1

	Selection Set
0) S → fBC	{f}
1) B → bb	{b}
2) B → CD	{c, d, e}
3) C → cC	{c}
4) C → λ	{d, e, #}
5) D → dD	{d}
6) D → e	{e}

Parse table



Current token

Symbol on top of stack

	b	c	d	e	f	#

S					pop
					push(C)
					push(B)
					push(f)	

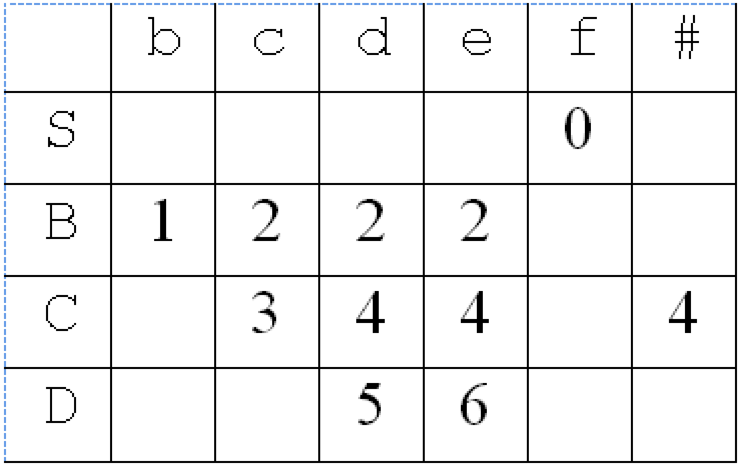
B	pop
	push(b)
	push(b)
		pop
		push(D)
		push(C)
			pop
			push(D)
			push(C)
				pop
				push(D)
				push(C)		

C		pop
		push(C)
		push(c)
			pop	pop		pop

D			pop
			push(D)
			push(d)
				pop
				push(e)		

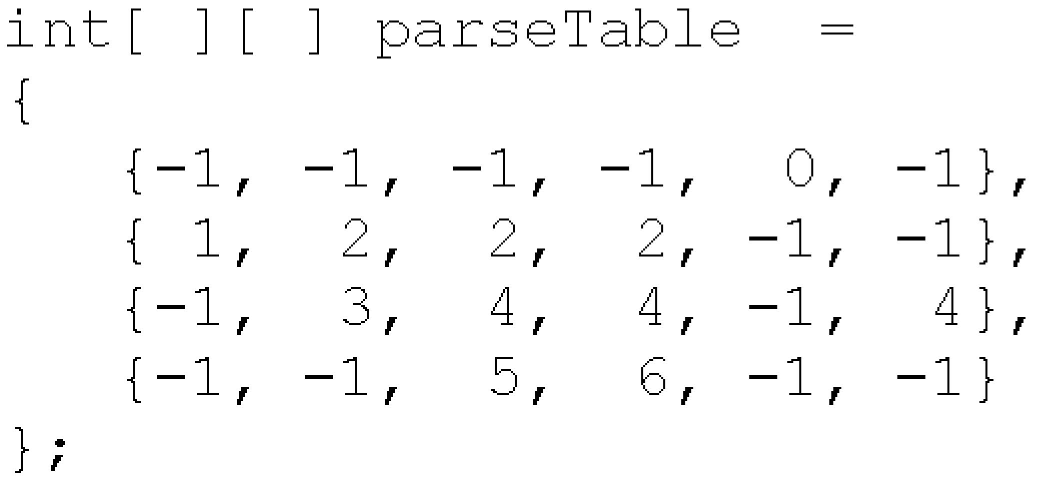
b	pop advance					
c		pop advance				
d			pop advance			
e				pop advance		
f					pop advance	
$						accept

Abbreviated parse table



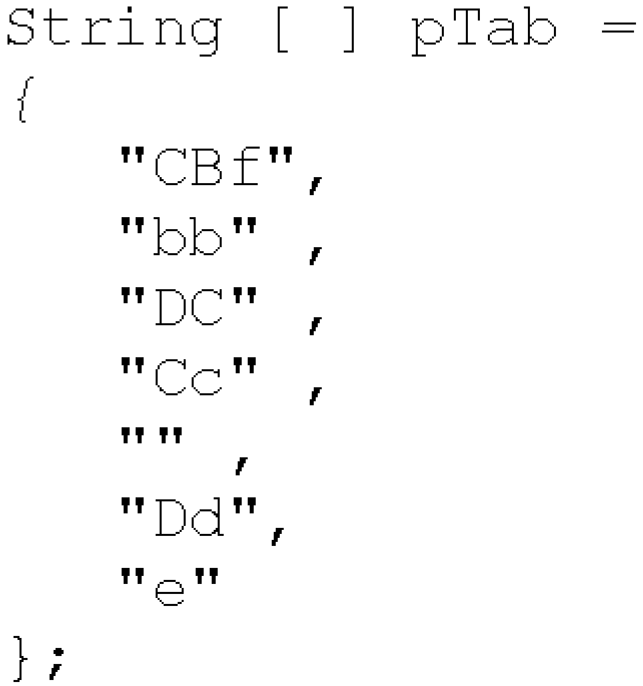
	b	c	d	e	f	#
S					0	
B	1	2	2	2		
C		3	4	4		4
D			5	6		

Implementing uniform stack parser



```
int[ ][ ] parseTable  =
{
    {-1, -1, -1, -1,  0, -1},
    { 1,  2,  2,  2, -1, -1},
    {-1,  3,  4,  4, -1,  4},
    {-1, -1,  5,  6, -1, -1}
};
```

Representing productions



```
String [ ] pTab =
{
    "CBf",
    "bb" ,
    "DC" ,
    "Cc" ,
    "" ,
    "Dd",
    "e"
};
```

Mapping terminal/nonterminal to index

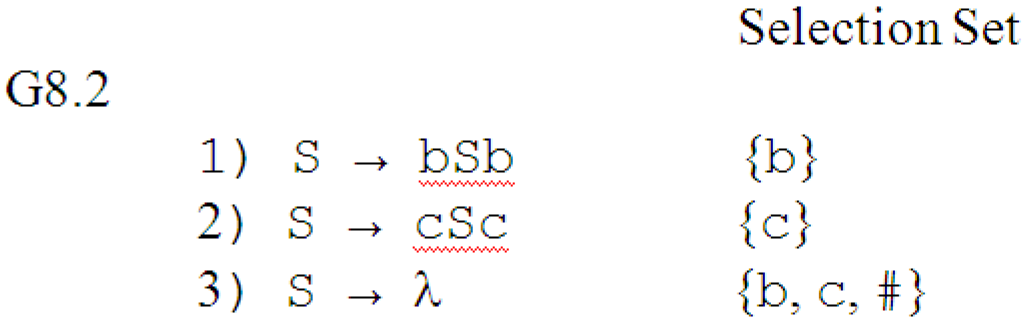
```
String nonTerms = "SBCD";

String tokens = "bcdef#";
```

Table-driven stack parser

Fig0804.txt

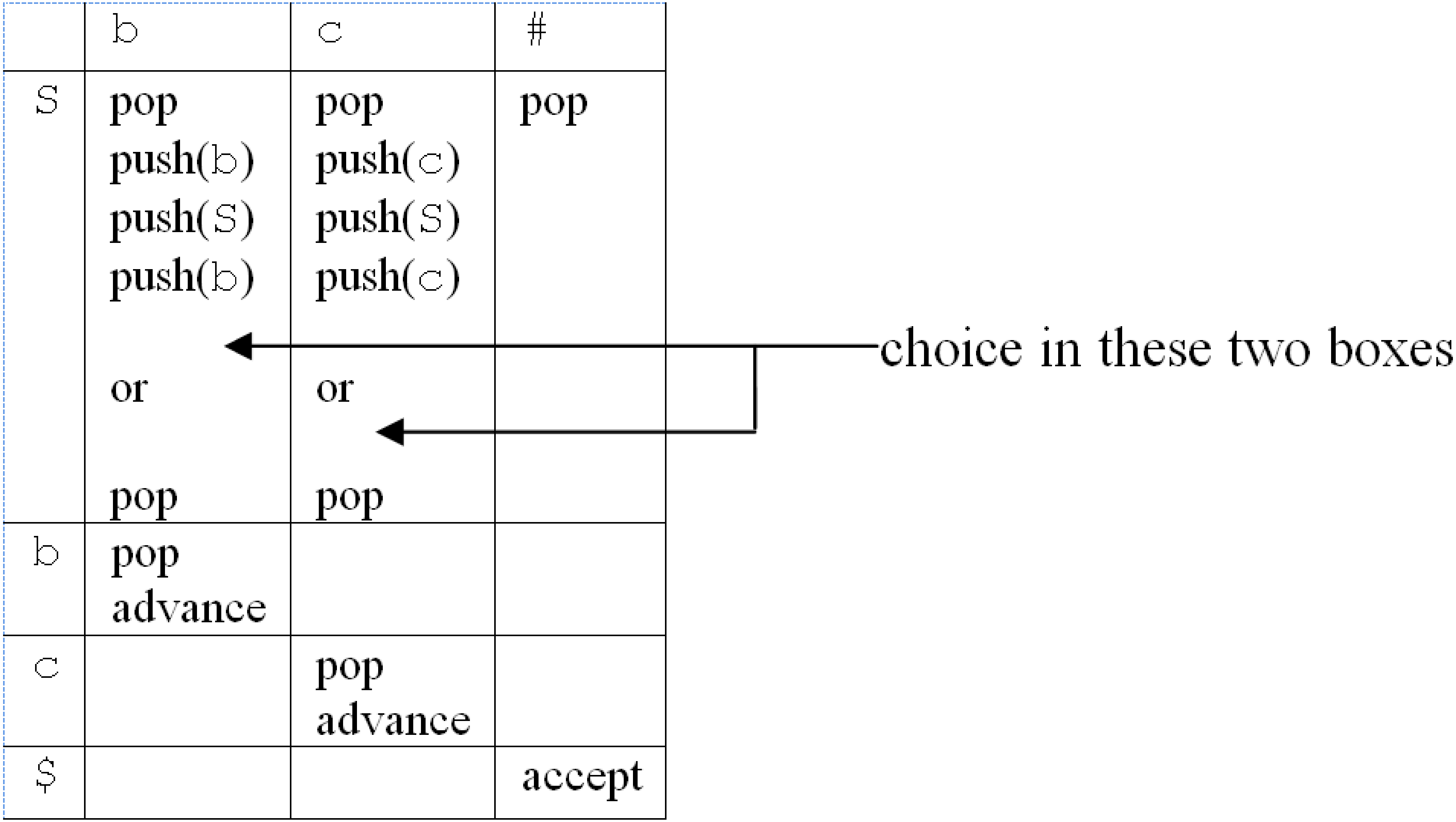
Non-deterministic parsers



G8.2

	Selection Set
1) S → bSb	{b}
2) S → cSc	{c}
3) S → λ	{b, c, #}

Parse table has choices



	b	c	#

S	pop
	push(b)
	push(S)
	push(b)
	or
	pop
		pop
		push(c)
		push(S)
		push(c)
		or
		pop
			pop

b	pop advance		
c		pop advance	
$			accept

choice in these two boxes

Interesting theoretical result

A language can be defined by a context-free grammar if and only if it can be defined by a stack parser.

In automata theory, the abstract model of a stack parser is called a pushdown automaton.