The Unger Parser

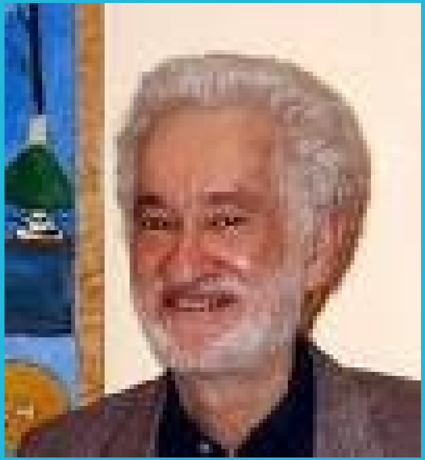
brought to you today by: Anne Brock

Outline

- Unger - the man
- Unger - the parser
- Unger's method, simple version
- some improvements
- Unger's method, including ε - rules

1. Unger: The man

Stephen H. Unger



- Politechnic Institute of Brooklyn
- doctorate at MIT
- Bell Telephone Labs
    - research in digital systems
    - head of development group (first electronic telephone switching system)
- since 1961: Prof. of Computer Science and Electrical Engeneering at Columbia University
- 1968: the Parser.
- since: published several books.

2. The Parser

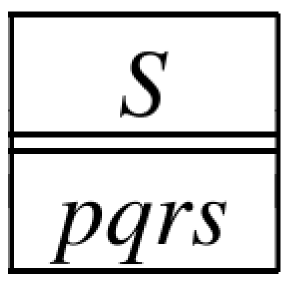
- non-directional
- top-down
- Type 2 grammars (CFG)

3. Unger's method, simplified

Input: CFG and a String/sentence, for example:

grammar: S —> ABC | DE | F

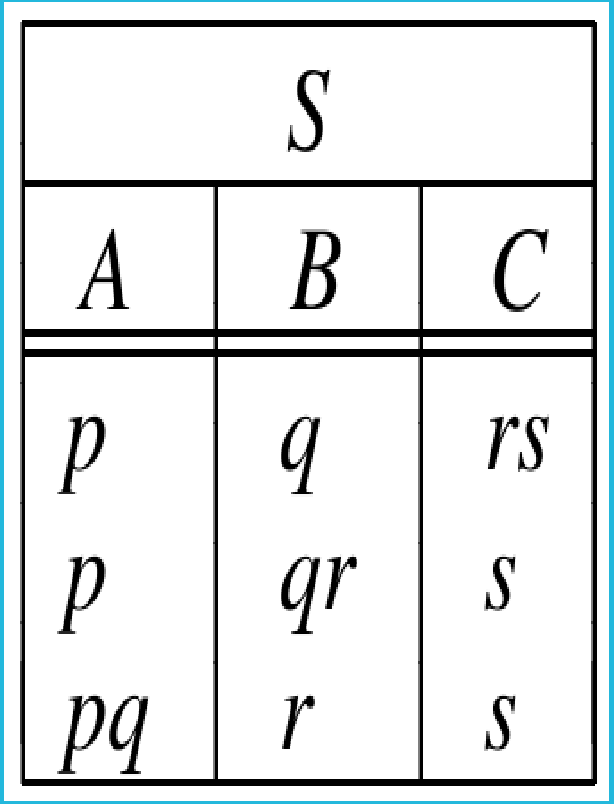
'sentence': pqrs



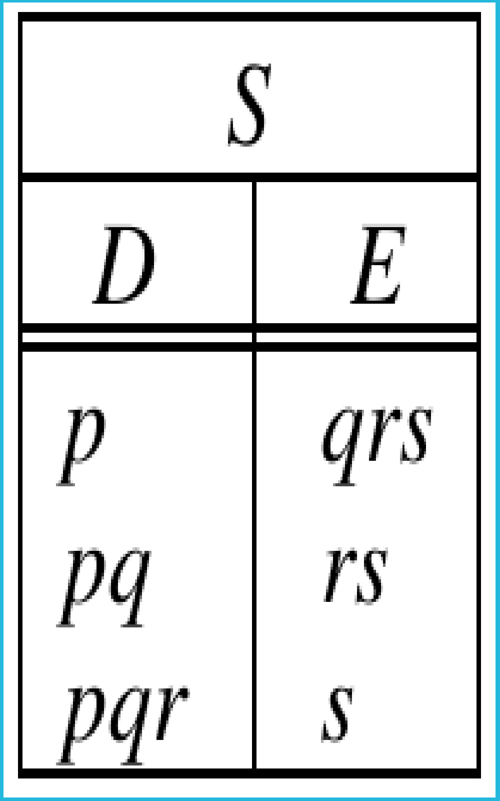
S
pqrs

Does S derive...

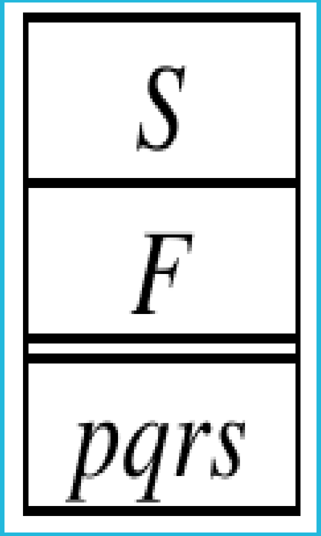
ABC | DE | F ?



S		
A	B	C
p	q	rs
p	qr	s
pq	r	s



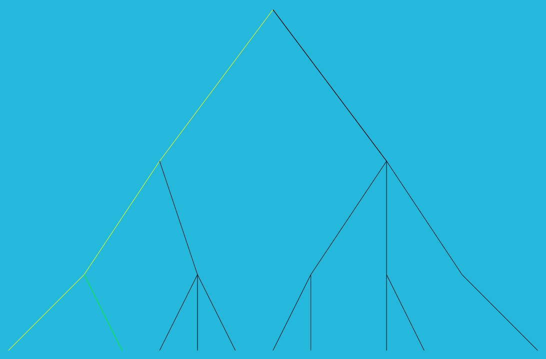
S	
D	E
p	qrs
pq	rs
pqr	s

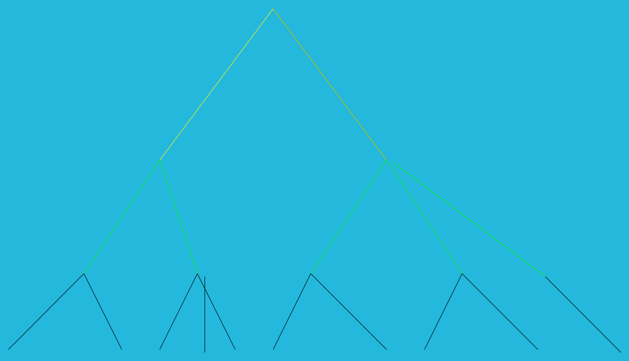


S
F
pqrs

This is a search problem.

Search: depth-first or breadth-first?





A more detailed example

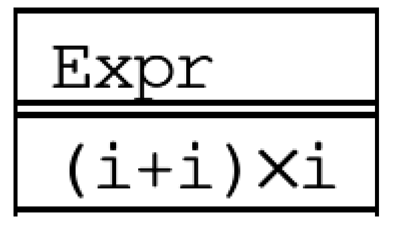
Grammar:

E -> E + T | T
T -> T x F | F
F -> (E) | i

E	= Expression
T	= Term
F	= Factor
+, x	= operators
i	= operand

Input:

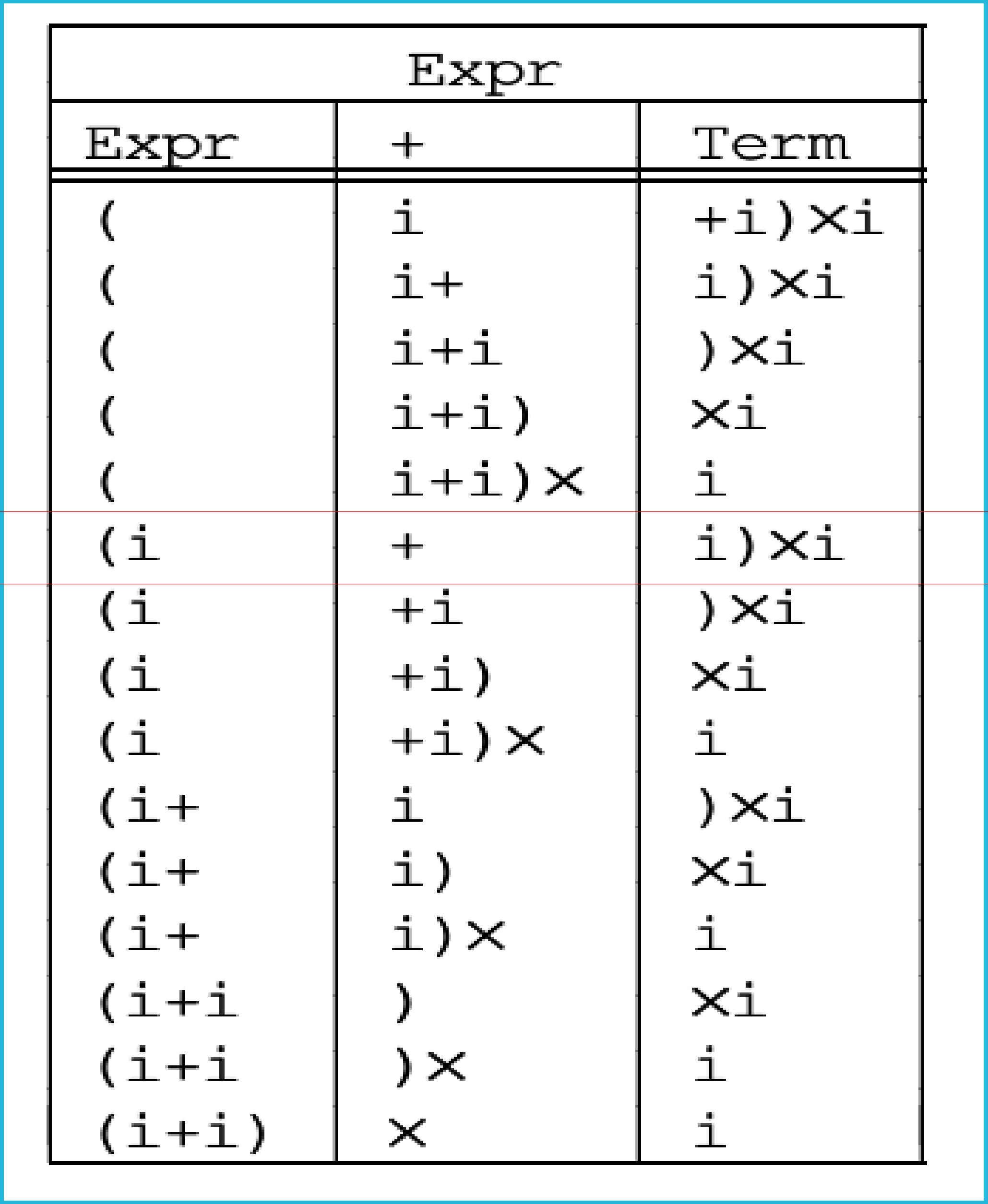
(i + i) x i



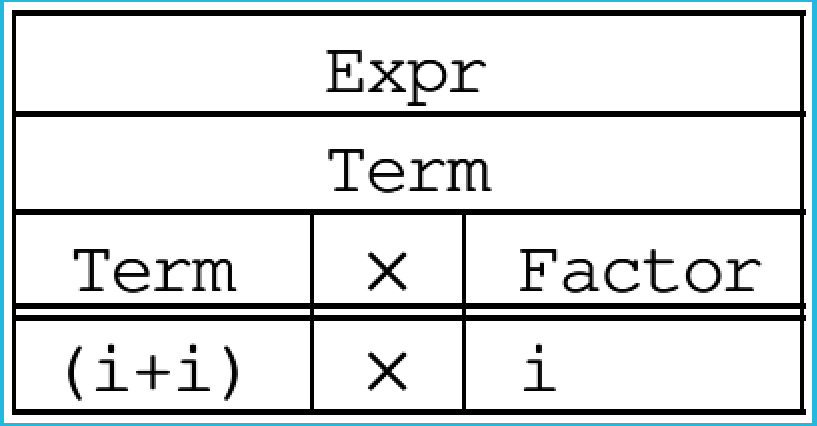
Expr

(i+i)Xi

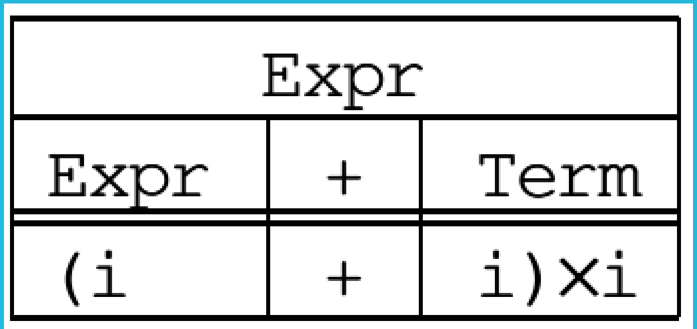
E -> E + T | T



Expr		
Expr	+	Term
(	i	+i)×i
(	i+	i)×i
(	i+i	)×i
(	i+i)	×i
(	i+i)×	i
(i	+	i)×i
(i	+i	)×i
(i	+i)	×i
(i	+i)×	i
(i+	i	)×i
(i+	i)	×i
(i+	i)×	i
(i+i	)	×i
(i+i	)×	i
(i+i)	×	i

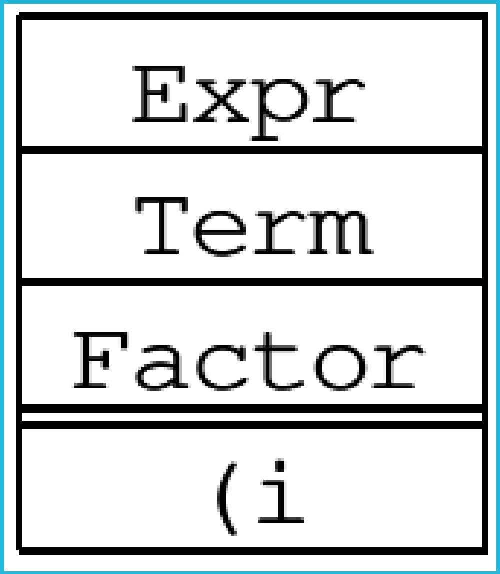


Expr		
Term		
Term	×	Factor
(i+i)	×	i



Expr		
Expr	+	Term
(i	+	i)xi

E ->* (i ?



Expr
Term
Factor
(i

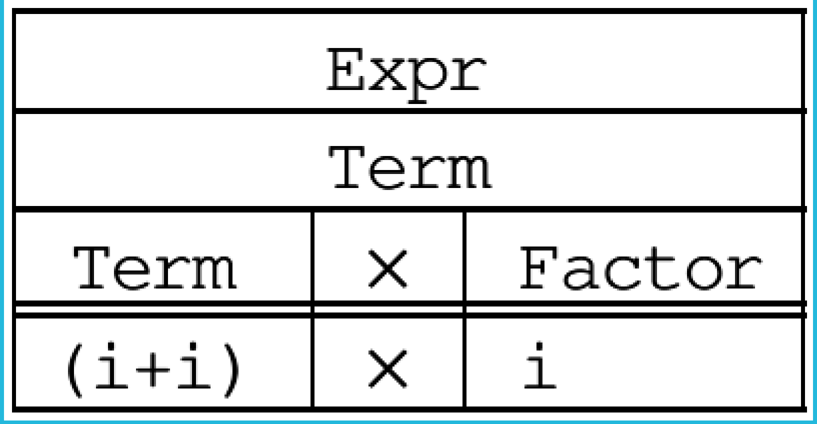
E -> E + T | T
T -> T x F | F
F -> (E) | i

fails!

to derive: (i + i) x i

E -> E + T | T

- fails!



Expr		
Term		
Term	×	Factor
(i+i)	×	i

(E -> E + T | T)

T -> T x F | F

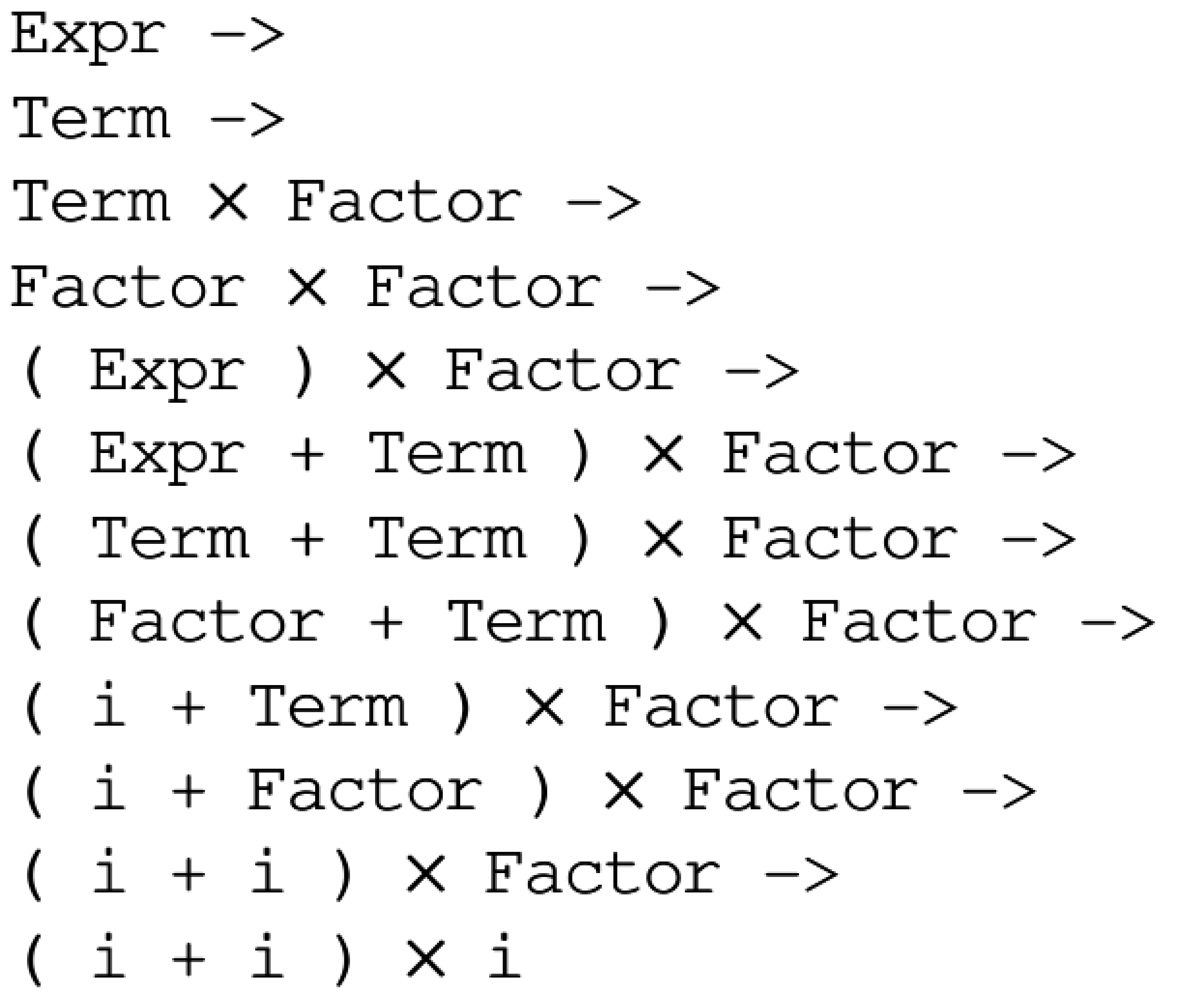
T -> T x F | F

F -> (E) | i

F -> (E) | i

- success!

E -> E + T | T
T -> T x F | F
F -> (E) | i



```
Expr ->
Term ->
Term × Factor ->
Factor × Factor ->
( Expr ) × Factor ->
( Expr + Term ) × Factor ->
( Term + Term ) × Factor ->
( Factor + Term ) × Factor ->
( i + Term ) × Factor ->
( i + Factor ) × Factor ->
( i + i ) × Factor ->
( i + i ) × i
```

4. Room for improvement...

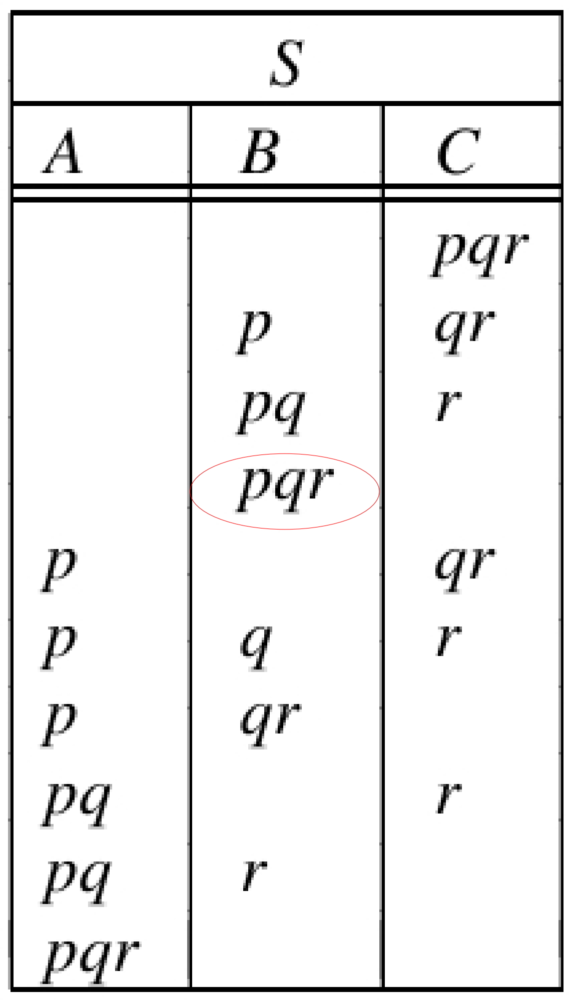
- consider the actual terminal symbols

- consider the length of your input

5. Unger's method with ε-rules

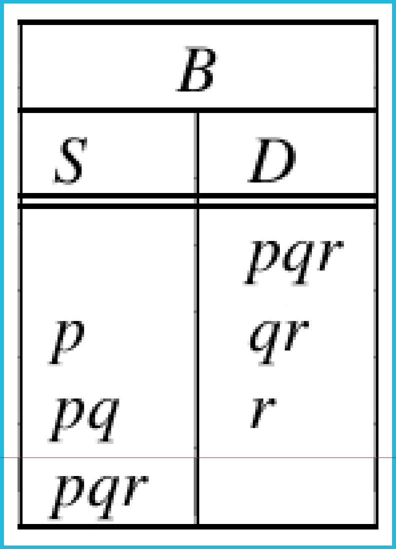
S -> ABC
B -> SD

try and derive:
B -> pqr



S		
A	B	C
		pqr
	p	qr
	pq	r
	pqr	
p		qr
p	q	r
p	qr	
pq		r
pq	r	
pqr		

S -> ABC
B -> SD
...

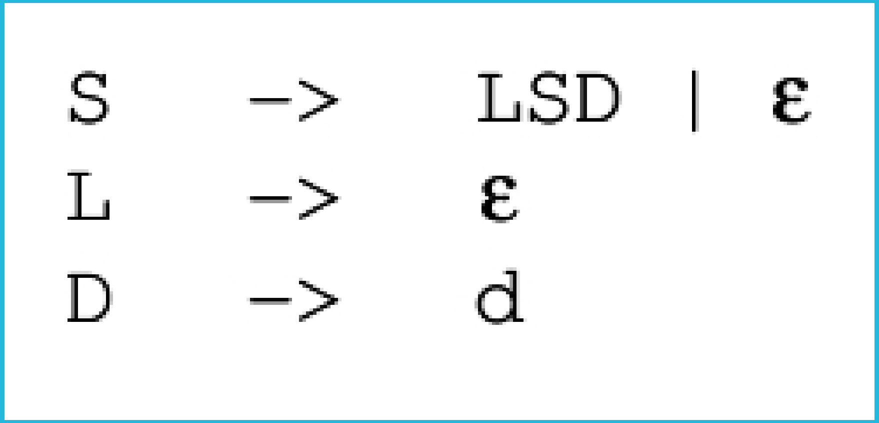


B	
S	*D*
	pqr
p	*qr*
pq	*r*
pqr	

What to do about it?

-> Keep a list of currently considered questions!

An example.



```
S    ->    LSD | ε
L    ->    ε
D    ->    d
```

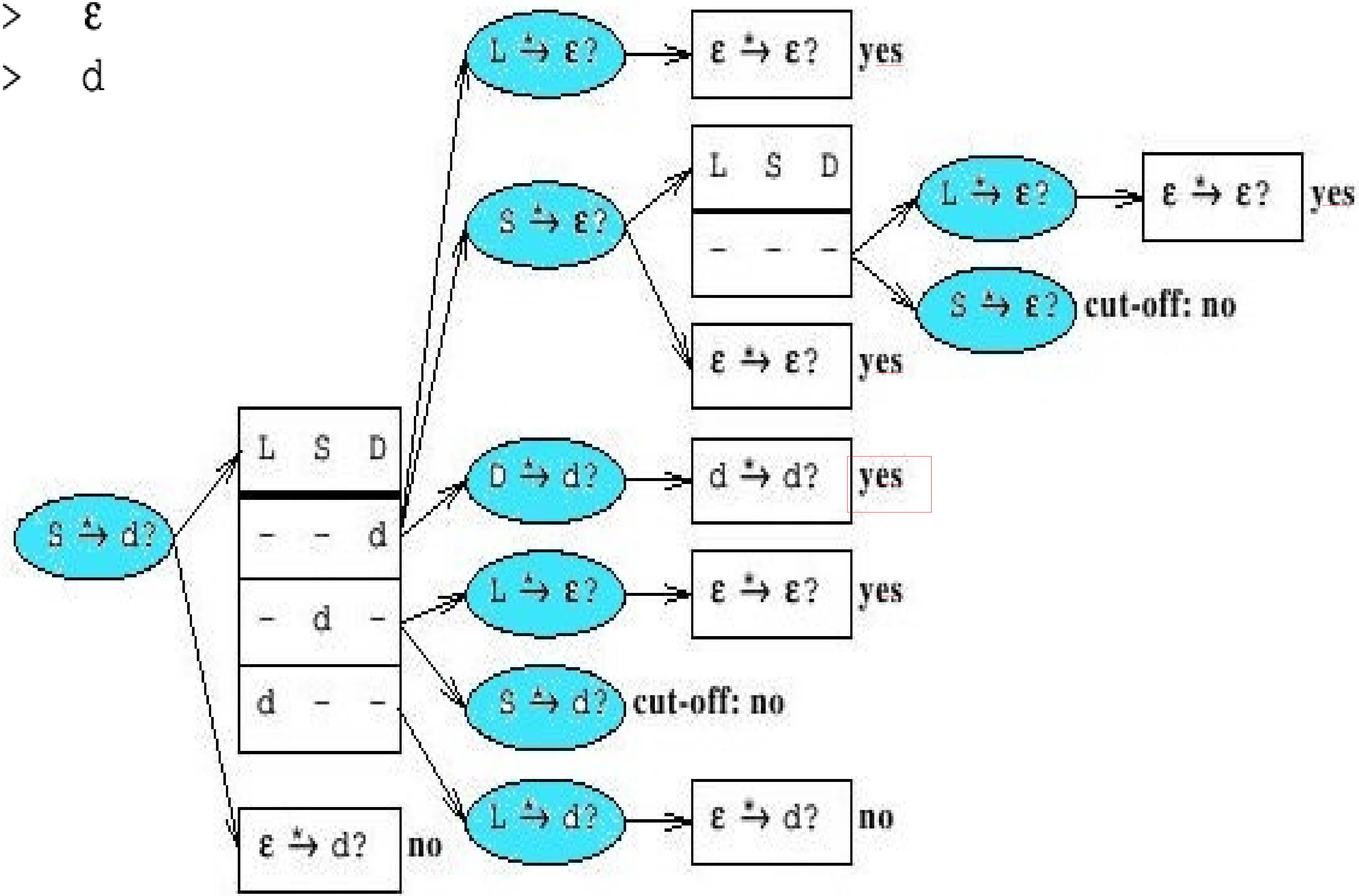
How does this grammar derive d ? dd ?

d ?

S -> LSD | ε

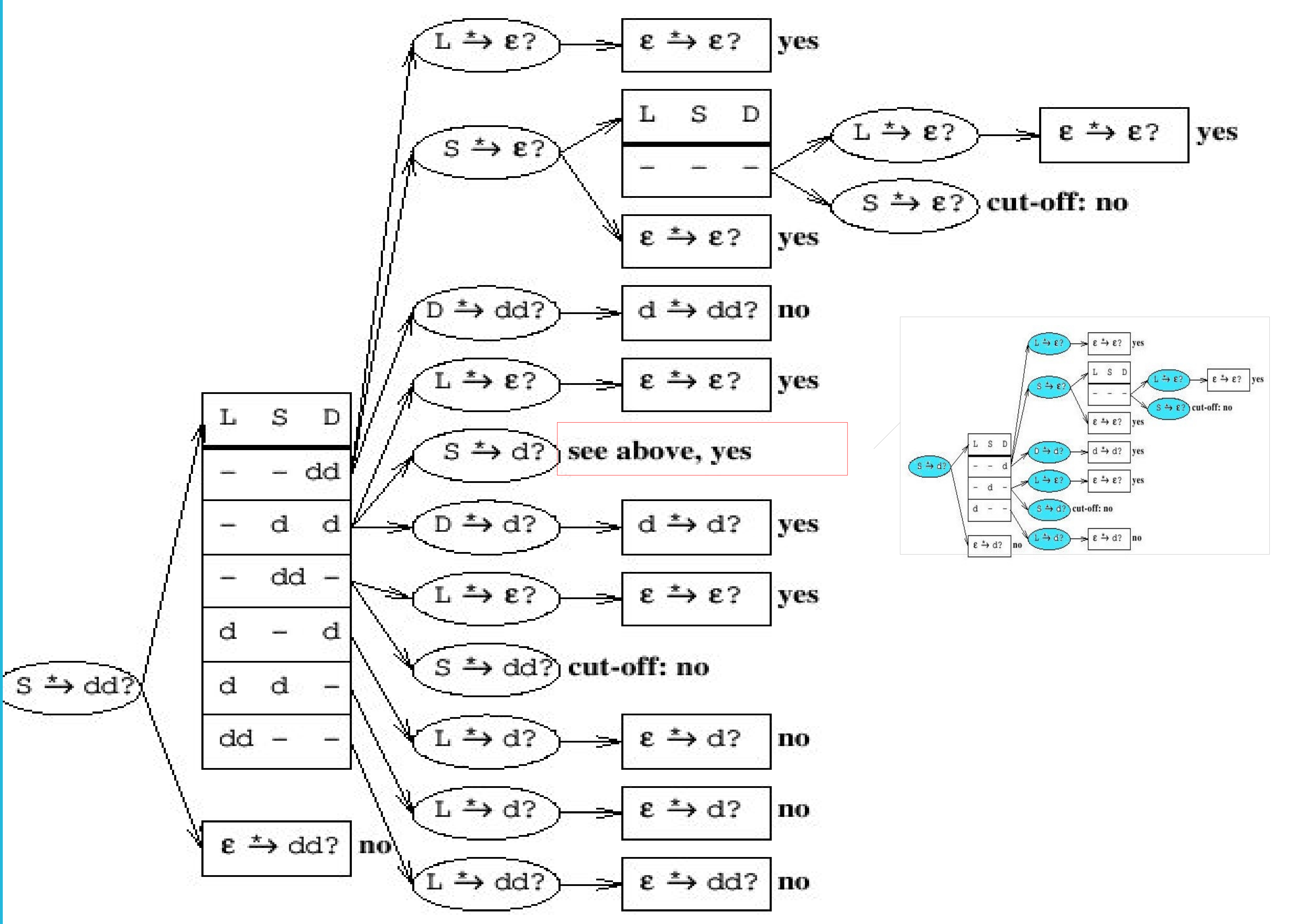
L -> ε

D -> d

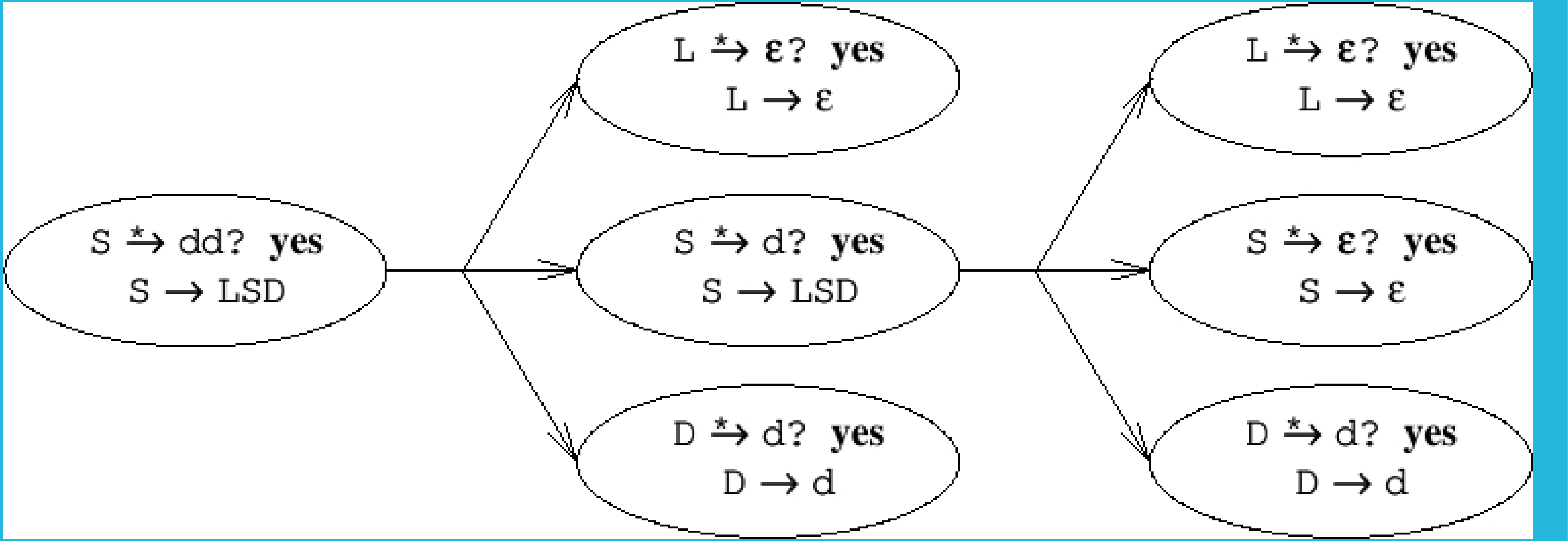


$S \overset{*}{\rightarrow} d?$
L S D
– – d
– d –
d – –
$\varepsilon \overset{*}{\rightarrow} d?$
no
$L \overset{*}{\rightarrow} \varepsilon?$
$\varepsilon \overset{*}{\rightarrow} \varepsilon?$
yes
$S \overset{*}{\rightarrow} \varepsilon?$
L S D
– – –
$L \overset{*}{\rightarrow} \varepsilon?$
$\varepsilon \overset{*}{\rightarrow} \varepsilon?$
yes
$S \overset{*}{\rightarrow} \varepsilon?$
cut-off: no
$\varepsilon \overset{*}{\rightarrow} \varepsilon?$
yes
$D \overset{*}{\rightarrow} d?$
$d \overset{*}{\rightarrow} d?$
yes
$L \overset{*}{\rightarrow} \varepsilon?$
$\varepsilon \overset{*}{\rightarrow} \varepsilon?$
yes
$S \overset{*}{\rightarrow} d?$
cut-off: no
$L \overset{*}{\rightarrow} d?$
$\varepsilon \overset{*}{\rightarrow} d?$
no

dd ?


$S \xrightarrow{*} dd?$
L S D
– – dd
– d d
– dd –
d – d
d d –
dd – –
$\varepsilon \xrightarrow{*} dd?$ no
$L \xrightarrow{*} \varepsilon?$
$\varepsilon \xrightarrow{*} \varepsilon?$ yes
$S \xrightarrow{*} \varepsilon?$
L S D
– – –
$L \xrightarrow{*} \varepsilon?$
$\varepsilon \xrightarrow{*} \varepsilon?$ yes
$S \xrightarrow{*} \varepsilon?$ cut-off: no
$\varepsilon \xrightarrow{*} \varepsilon?$ yes
$D \xrightarrow{*} dd?$
$d \xrightarrow{*} dd?$ no
$L \xrightarrow{*} \varepsilon?$
$\varepsilon \xrightarrow{*} \varepsilon?$ yes
$S \xrightarrow{*} d?$ see above, yes
$D \xrightarrow{*} d?$
$d \xrightarrow{*} d?$ yes
$L \xrightarrow{*} \varepsilon?$
$\varepsilon \xrightarrow{*} \varepsilon?$ yes
$S \xrightarrow{*} dd?$ cut-off: no
$L \xrightarrow{*} d?$
$\varepsilon \xrightarrow{*} d?$ no
$L \xrightarrow{*} d?$
$\varepsilon \xrightarrow{*} d?$ no
$L \xrightarrow{*} dd?$
$\varepsilon \xrightarrow{*} dd?$ no
$S \xrightarrow{*} d?$
L S D
– – d
– d –
d – –
$\varepsilon \xrightarrow{*} d?$ no
$L \xrightarrow{*} \varepsilon?$
$\varepsilon \xrightarrow{*} \varepsilon?$ yes
$S \xrightarrow{*} \varepsilon?$
L S D
– – –
$L \xrightarrow{*} \varepsilon?$
$\varepsilon \xrightarrow{*} \varepsilon?$ yes
$S \xrightarrow{*} \varepsilon?$ cut-off: no
$\varepsilon \xrightarrow{*} \varepsilon?$ yes
$D \xrightarrow{*} d?$
$d \xrightarrow{*} d?$ yes
$L \xrightarrow{*} \varepsilon?$
$\varepsilon \xrightarrow{*} \varepsilon?$ yes
$S \xrightarrow{*} d?$ cut-off: no
$L \xrightarrow{*} d?$
$\varepsilon \xrightarrow{*} d?$ no

S ->* d ?



$S \xrightarrow{*} dd?$ **yes**

$S \to LSD$

$L \xrightarrow{*} \varepsilon?$ **yes**

$L \to \varepsilon$

$S \xrightarrow{*} d?$ **yes**

$S \to LSD$

$D \xrightarrow{*} d?$ **yes**

$D \to d$

$L \xrightarrow{*} \varepsilon?$ **yes**

$L \to \varepsilon$

$S \xrightarrow{*} \varepsilon?$ **yes**

$S \to \varepsilon$

$D \xrightarrow{*} d?$ **yes**

$D \to d$

S -> LSD -> SD -> LSDD -> SDD -> DD -> dD -> dd.

Summary

The Unger parser:

- is a non-directional, top-down parser;
- will consider each possible (and impossible) solution;
- requires at least polynomial, if not exponential time;
- is slightly improved by
    - matching input with possible derived terminals
    - calculating possible length, special case ε
    - remembering answers.

?

Sources

Grune, Dick and Jacobs, Ceriel 1990. *Parsing Techniques. A Practical Guide.* New York: Ellis Horwood Limited.

Lukasz Kwiatowski. *Reconciling Unger's parser as a top-down parser for CF grammars for experimental purposes.* http://www.cs.vu.nl/~steven/

pictures from:

www.cs.columbia.edu/async/images/unger.jpg

http://pinker.wjh.harvard.edu/photos/cambridge_boston/pages/trees%20in%20Cambridge%20Common.htm