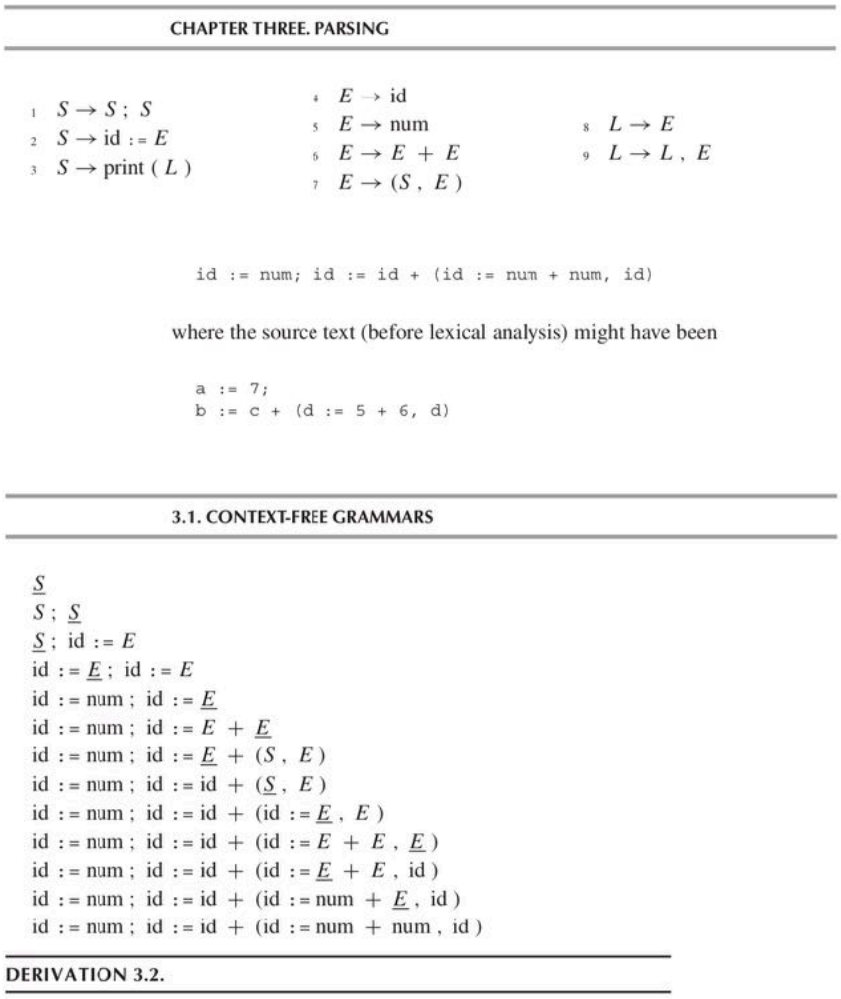
Compilers

CMPT 432

Lab 3

Goals	Turning tokens into sentences with your Parser
Notes	Now that we have tokens, we need to turn then into sentences that are (hopefully) valid in our language grammar.
Resources	*Crafting a Compiler* • Read chapters 4.1-4.4, 5.1-5.3, and 7.1 • Do exercises 4.7 (derivations) and 5.2c (recursive descent parser — pseudo code only) *Dragon* • Read chapters 2.7, 2.8.2, 4.2, 4.4.1, and 5.3.1 • Do exercises 4.2.1 a, b, and c (derivations and a parse tree)
Submitting	Commit a PDF of your work to your GitHub repository and I'll take a look at it.



CHAPTER THREE. PARSING

1 $S \rightarrow S\ ;\ S$
2 $S \rightarrow \text{id} := E$
3 $S \rightarrow \text{print}\ (\ L\ )$
4 $E \rightarrow \text{id}$
5 $E \rightarrow \text{num}$
6 $E \rightarrow E + E$
7 $E \rightarrow (S\ ,\ E\ )$
8 $L \rightarrow E$
9 $L \rightarrow L\ ,\ E$

```
id := num; id := id + (id := num + num, id)
```

where the source text (before lexical analysis) might have been

```
a := 7;
b := c + (d := 5 + 6, d)
```

3.1. CONTEXT-FREE GRAMMARS

$\underline{S}$
$S\ ;\ \underline{S}$
$\underline{S}\ ;\ \text{id} := E$
$\text{id} := \underline{E}\ ;\ \text{id} := E$
$\text{id} := \text{num}\ ;\ \text{id} := \underline{E}$
$\text{id} := \text{num}\ ;\ \text{id} := E + \underline{E}$
$\text{id} := \text{num}\ ;\ \text{id} := \underline{E} + (S\ ,\ E\ )$
$\text{id} := \text{num}\ ;\ \text{id} := \text{id} + (\underline{S}\ ,\ E\ )$
$\text{id} := \text{num}\ ;\ \text{id} := \text{id} + (\text{id} := \underline{E}\ ,\ E\ )$
$\text{id} := \text{num}\ ;\ \text{id} := \text{id} + (\text{id} := E + E\ ,\ \underline{E}\ )$
$\text{id} := \text{num}\ ;\ \text{id} := \text{id} + (\text{id} := \underline{E} + E\ ,\ \text{id}\ )$
$\text{id} := \text{num}\ ;\ \text{id} := \text{id} + (\text{id} := \text{num} + \underline{E}\ ,\ \text{id}\ )$
$\text{id} := \text{num}\ ;\ \text{id} := \text{id} + (\text{id} := \text{num} + \text{num}\ ,\ \text{id}\ )$

DERIVATION 3.2.

from *Modern Compiler Implementation in Java* by Andrew Appel