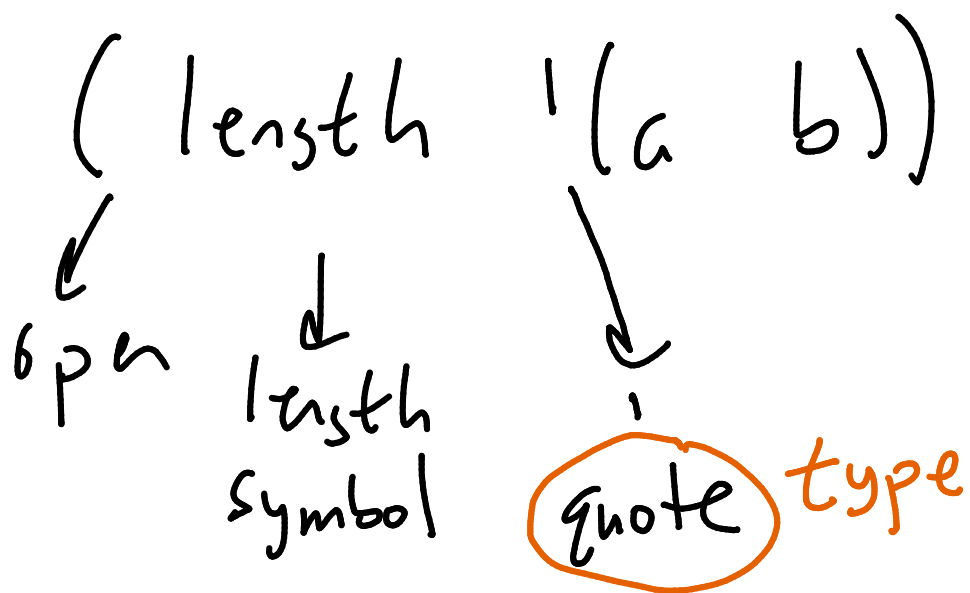
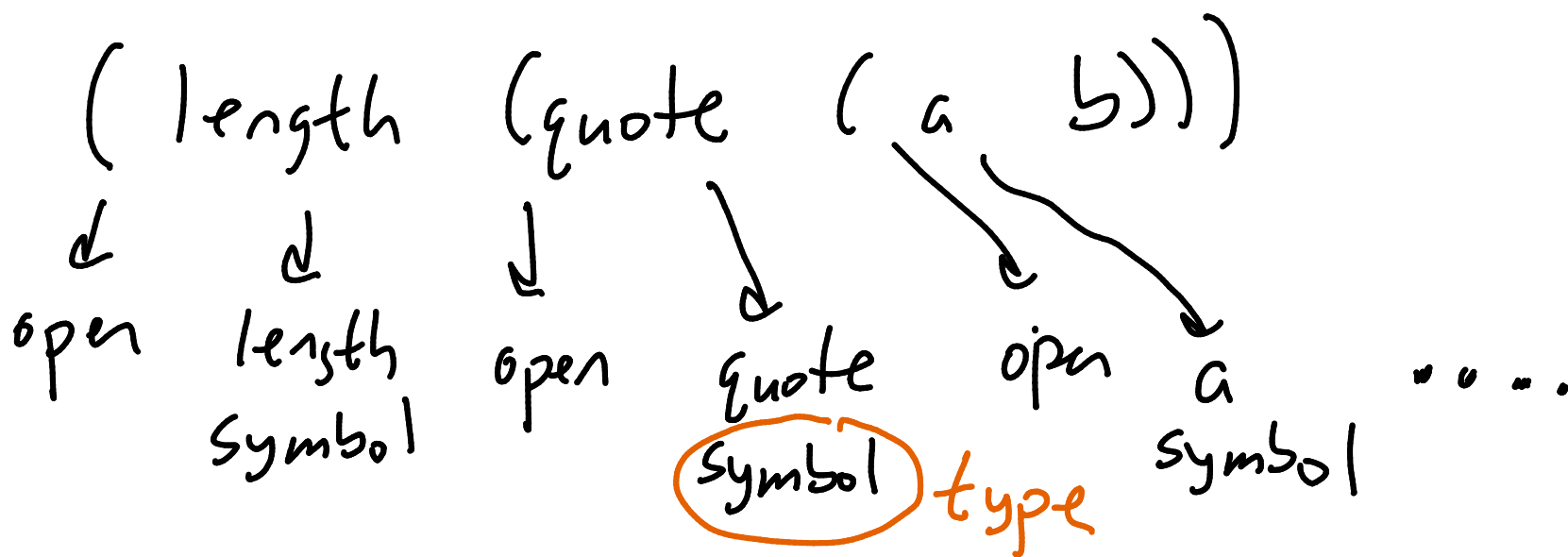
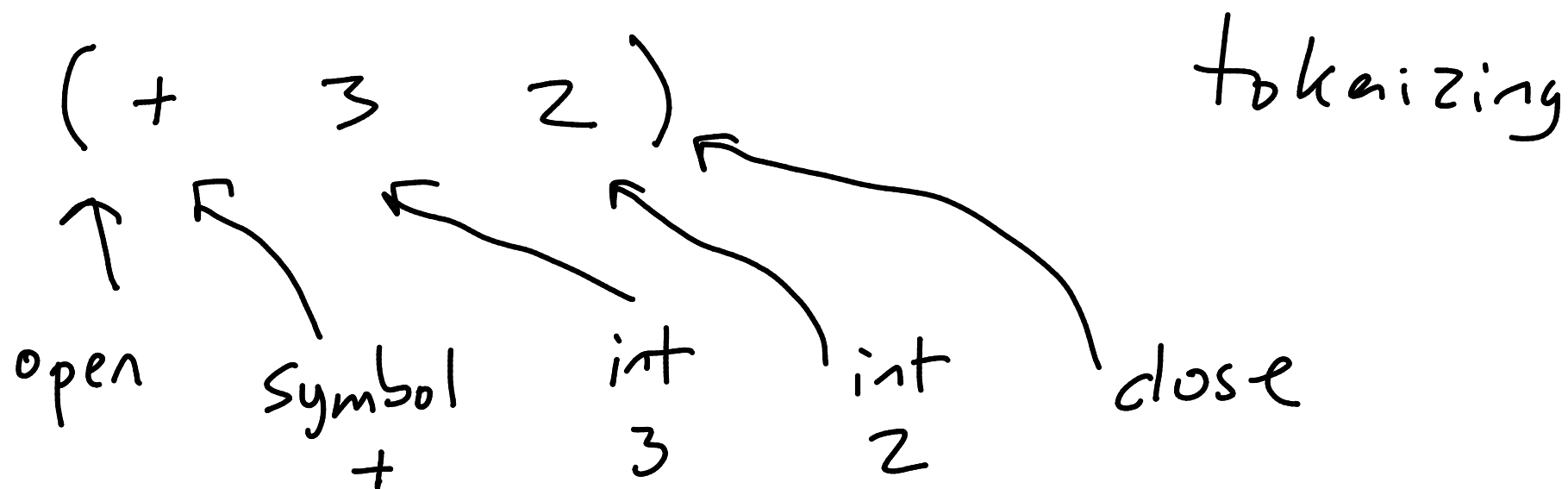


Worksheet (complete intp is on Moodle)

Solutions

A little more about tokenizer assignments

BNF and why it matters



(quote ')

(:open

quote: symbol)

' : quote ← type

) : close

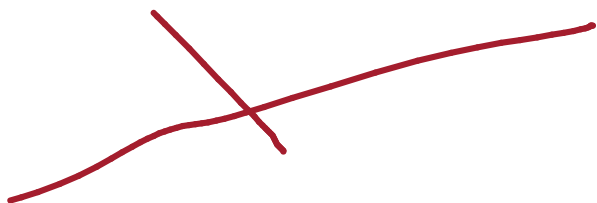
↑

↑

actual
text

type

(lexeme)



BNF (John Backus)

↳ Backus Normal Form

ALGOL → old influential programming language
wanted a way of precisely describing what code
was legal and what wasn't → syntax

Example' ↙ starting
symbol

↳ what you are
allowed type

<sentence> ::= <subject> <predicate>

<subject> ::= <article> <noun>

<predicate> ::= <verb> <article> <noun>

<verb> ::= ran | ate
↳ "or"

<article> ::= the

<noun> ::= elf | dwarf | cake

BNF
grammar

Sample derivation

<sentence>

⇒ <subject> <predicate>

⇒ <article> <noun> <verb> <article> <noun>

⇒ the elf ran the cake

All possible derivations = all possible sentences.

In practice, we use it the other way around.

Given a sentence (or a program), can it be derived? If it can't, that's an error.

The the the the cake the
Find (or fail to find) a derivation to fit
this sentence. \rightarrow this is called parsing .

Grammar for binary numbers: $\textcircled{1}$ $\textcircled{2}$ $\textcircled{3}$ $\textcircled{4}$

$$\langle B \rangle ::= 0 \langle B \rangle \mid 1 \langle B \rangle \mid 0 \mid 1$$

starting symbol

Given the program 0110, is there a derivation? (Can we parse this?)

Derivation

$\langle B \rangle$ $\downarrow \textcircled{1}$
 $0 \langle B \rangle$ $\downarrow \textcircled{2}$
 $0 1 \langle B \rangle$ $\downarrow \textcircled{2}$
 $0 1 1 \langle B \rangle$ $\downarrow \textcircled{2}$
 $0 1 1 0$ $\downarrow \textcircled{3}$

Parse tree

$\langle B \rangle$
 \swarrow \searrow
0 $\langle B \rangle$
 \swarrow \searrow
 1 $\langle B \rangle$
 \swarrow \searrow
 1 $\langle B \rangle$
 \swarrow \searrow
 1 $\langle B \rangle$
 \downarrow
 0

$\langle B \rangle$ non-terminal

\circ terminal

✓
an actual token in
your language

$\therefore =$
 $\langle \text{number} \rangle \rightarrow \langle \text{sign} \rangle \langle \text{ureal} \rangle \mid \langle \text{ureal} \rangle$
 $\langle \text{sign} \rangle \rightarrow + \mid -$
 $\langle \text{ureal} \rangle \rightarrow \langle \text{uinteger} \rangle \mid \langle \text{udecimal} \rangle$
 $\langle \text{uinteger} \rangle \rightarrow \langle \text{digit} \rangle^+ \leftarrow 1 \text{ or more}$
 $\langle \text{udecimal} \rangle \rightarrow \underline{\langle \text{digit} \rangle^+} \mid \underline{\langle \text{digit} \rangle^+} \underline{\langle \text{digit} \rangle^*} \leftarrow 0 \text{ or more}$
 $\langle \text{digit} \rangle \rightarrow 0 \mid 1 \mid \dots \mid 9$

Legal numbers in Scheme (for purposes of the assignment)

Sample derivation:

$\langle \text{number} \rangle$

$\langle \text{sign} \rangle \langle \text{ureal} \rangle$

$= \langle \text{uinteger} \rangle$

12378

