# Grammars in Action

Example: A simple programming language grammar.

```
G: <Exp>* ::= <Exp> + <Exp>
            |   <Exp> * <Exp>
            |   ( <Exp> )
            |   a
            |   b
            |   c
```

Terminal symbols!!!

```
S = a
S = a + b
S = a + b * c
S = (a + b) * c
S = ((a + b))
S = c(a + b)
S = (c)+(b)
S = b++
```

$S \in L(G)$?

# Grammars in Action

- The empty symbol: <empty>
- The only non-terminal that does not have a rule defining it.
- That is the <empty> symbol derives nothing.

# <empty>

- Consider the grammar:

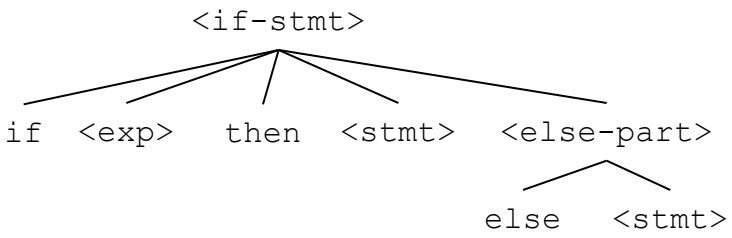
```
G:  <A>*::= a <B> | a
    <B> ::= b <B> | b
```

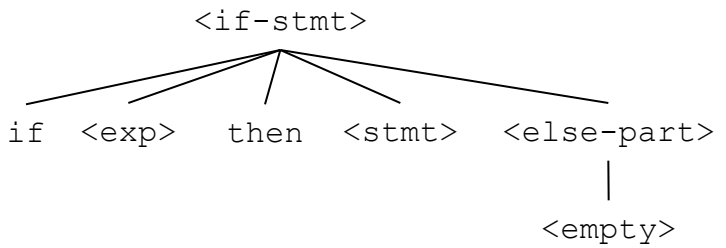
```
G': <A>*::= a <B>
    <B> ::= b <B> | <empty>
```

# Grammars in Action

Consider the following grammar fragment:

```
<if-stmt>   ::= if <exp> then <stmt> <else-part>
<else-part> ::= else <stmt> | <empty>
<exp>       ::= ...
<stmt>      ::= ...
```



# Grammars in Action

2.1 a) Let $L(G)$ be the language of all string consisting of zero or more a's.

2.1 i) Let $L(G)$ be the set of strings consisting of one or more a's with a comma between each a and the next.

2.1 d) Let $L(G)$ be the set of all strings consisting of one or more digits 0 – 9.

HW#1 – see website