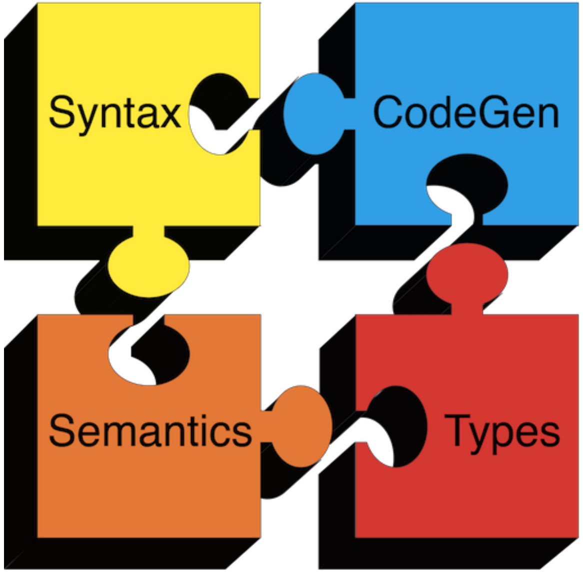


Compilers

Derivations

Alex Aiken

Derivations

A *derivation* is a sequence of productions

$$S \to \ldots \to \ldots \to \ldots \to \ldots \to \ldots$$

A derivation can be drawn as a tree

- Start symbol is the tree's root
- For a production $X \to Y_1 \ldots Y_n$ add children $Y_1 \ldots Y_n$ to node X

Derivations

- Grammar

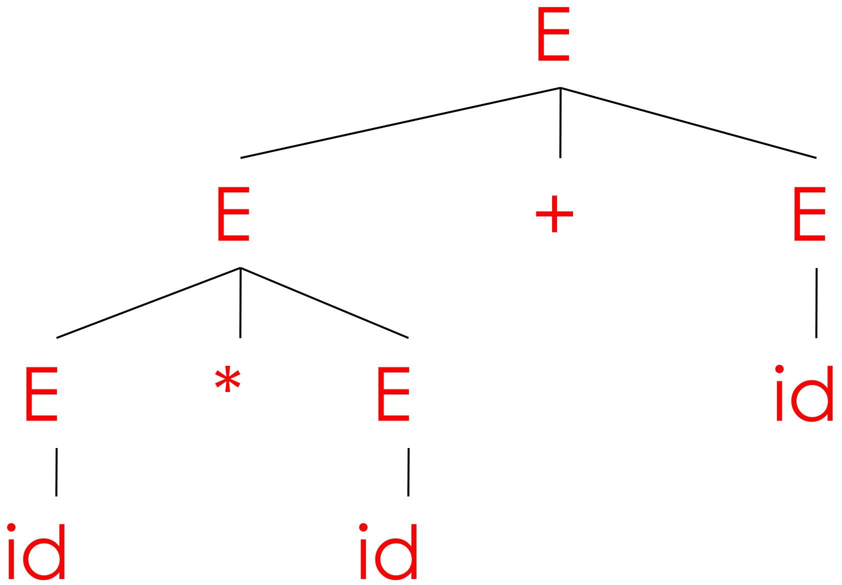
$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

- String

$$id * id + id$$

Derivations

$$
\begin{aligned}
& E \\
\rightarrow\ & E+E \\
\rightarrow\ & E * E+E \\
\rightarrow\ & id * E + E \\
\rightarrow\ & id * id + E \\
\rightarrow\ & id * id + id
\end{aligned}
$$



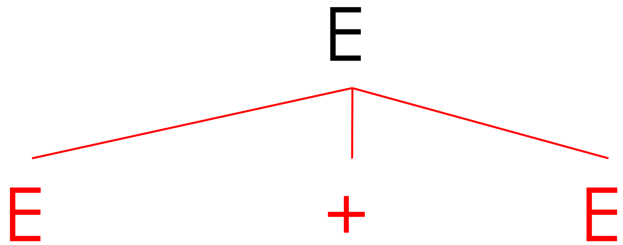
Derivations

E

$\mathbf{E}$

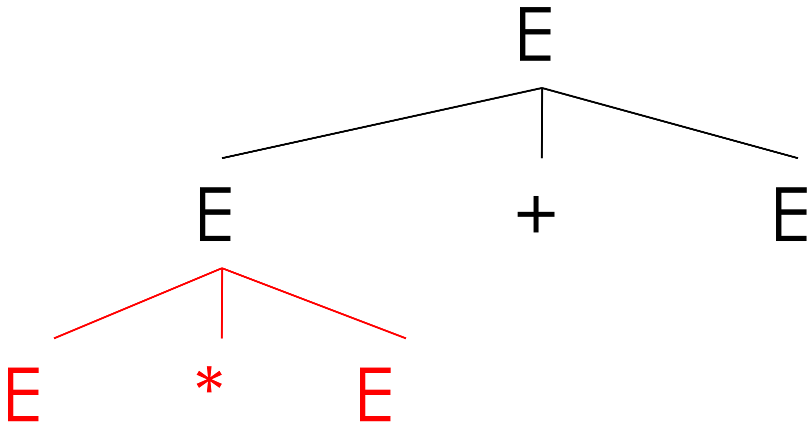
Derivations

$$\begin{aligned} & \mathbf{E} \\ \rightarrow \quad & \mathbf{E+E} \end{aligned}$$



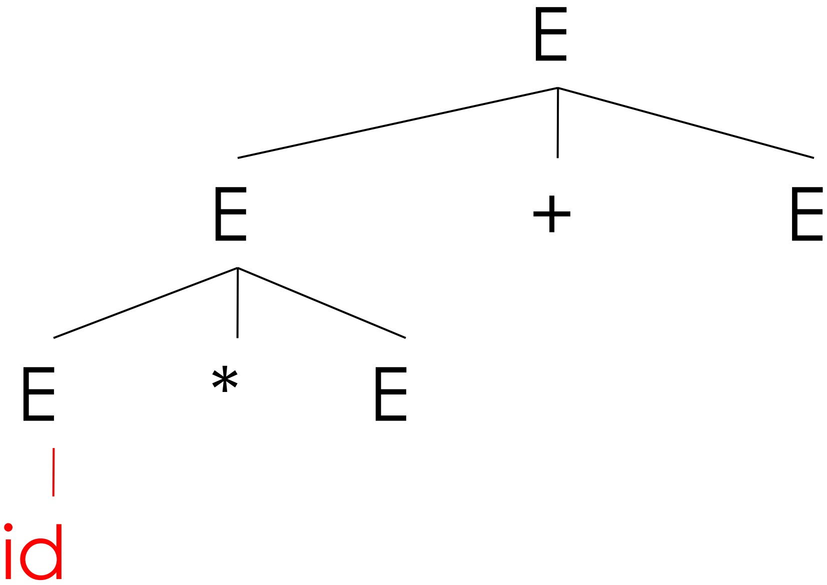
Derivations

$$\begin{aligned} & \mathbf{E} \\ \rightarrow\ & \mathbf{E+E} \\ \rightarrow\ & \mathbf{E * E+E} \end{aligned}$$



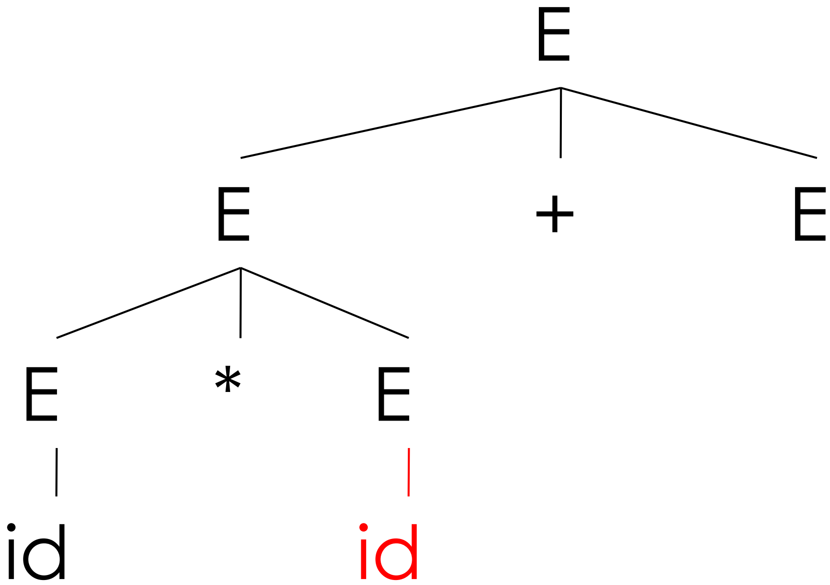
Derivations

$$
\begin{aligned}
& E \\
\rightarrow\ & E+E \\
\rightarrow\ & E * E+E \\
\rightarrow\ & id * E + E
\end{aligned}
$$



Derivations

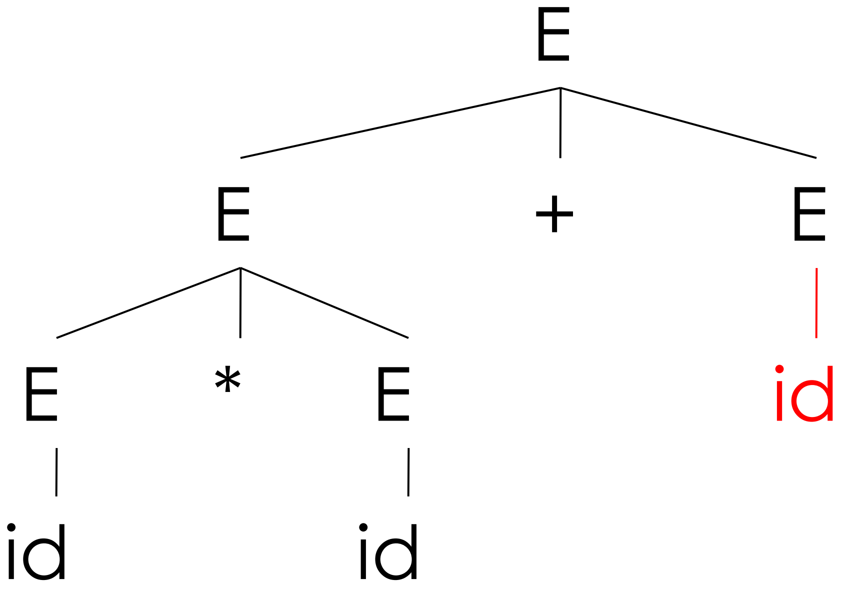
$$
\begin{aligned}
& \mathbf{E} \\
\rightarrow\ & \mathbf{E+E} \\
\rightarrow\ & \mathbf{E * E+E} \\
\rightarrow\ & \mathbf{id * E + E} \\
\rightarrow\ & \mathbf{id * id + E}
\end{aligned}
$$



Alex Aiken

Derivations

$$
\begin{array}{ll}
 & \mathbf{E} \\
\rightarrow & \mathbf{E+E} \\
\rightarrow & \mathbf{E * E+E} \\
\rightarrow & \mathbf{id * E + E} \\
\rightarrow & \mathbf{id * id + E} \\
\rightarrow & \mathbf{id * id + id}
\end{array}
$$



Derivations

- A parse tree has
  - Terminals at the leaves
  - Non-terminals at the interior nodes

- An in-order traversal of the leaves is the original input

- The parse tree shows the association of operations, the input string does not

Derivations

- The example is a *left-most* derivation
  - At each step, replace the left-most non-terminal

- There is an equivalent notion of a *right-most* derivation

$$\begin{aligned} & \mathbf{E} \\ \rightarrow\ & \mathbf{E+E} \\ \rightarrow\ & \mathbf{E+id} \\ \rightarrow\ & \mathbf{E} * \mathbf{E} + \mathbf{id} \\ \rightarrow\ & \mathbf{E} * \mathbf{id} + \mathbf{id} \\ \rightarrow\ & \mathbf{id} * \mathbf{id} + \mathbf{id} \end{aligned}$$

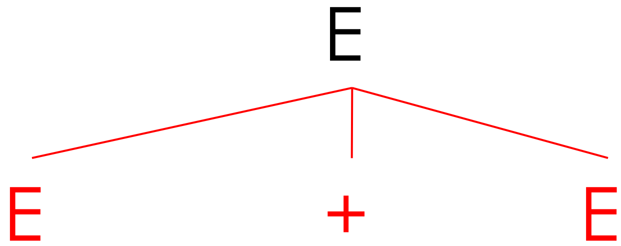
Derivations

E

E

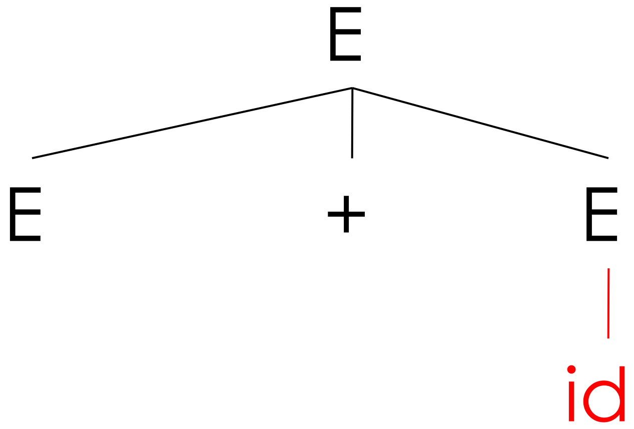
Derivations

$$
\begin{array}{ll}
 & \mathbf{E} \\
\rightarrow & \mathbf{E+E}
\end{array}
$$



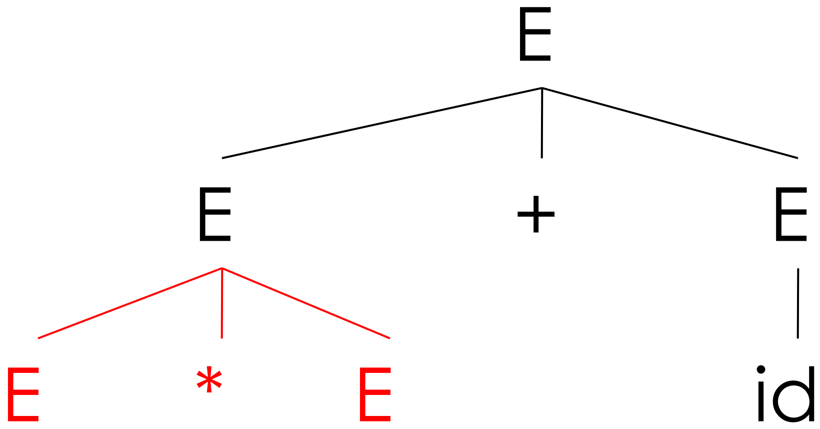
Derivations

$$\begin{aligned} & \mathbf{E} \\ \rightarrow \quad & \mathbf{E+E} \\ \rightarrow \quad & \mathbf{E+id} \end{aligned}$$



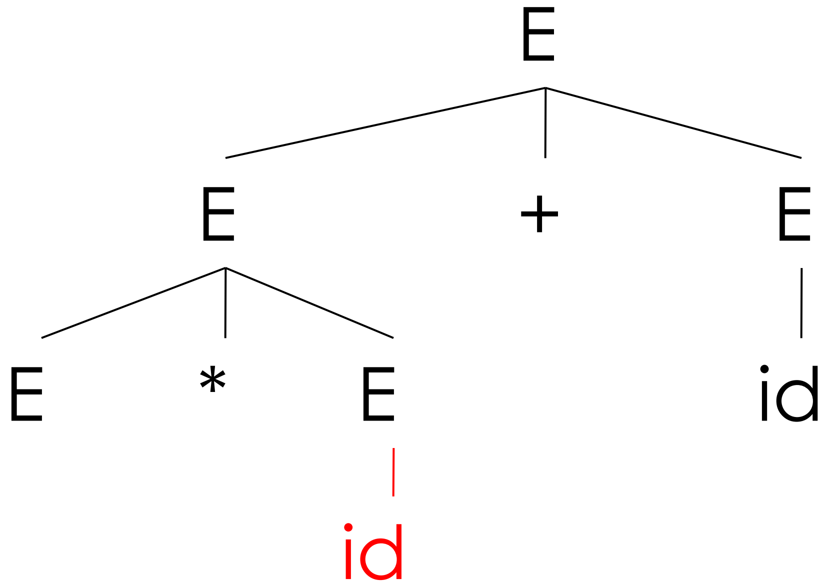
Derivations

$$
\begin{aligned}
& E \\
\rightarrow\quad & E+E \\
\rightarrow\quad & E+\mathrm{id} \\
\rightarrow\quad & E * E + \mathrm{id}
\end{aligned}
$$



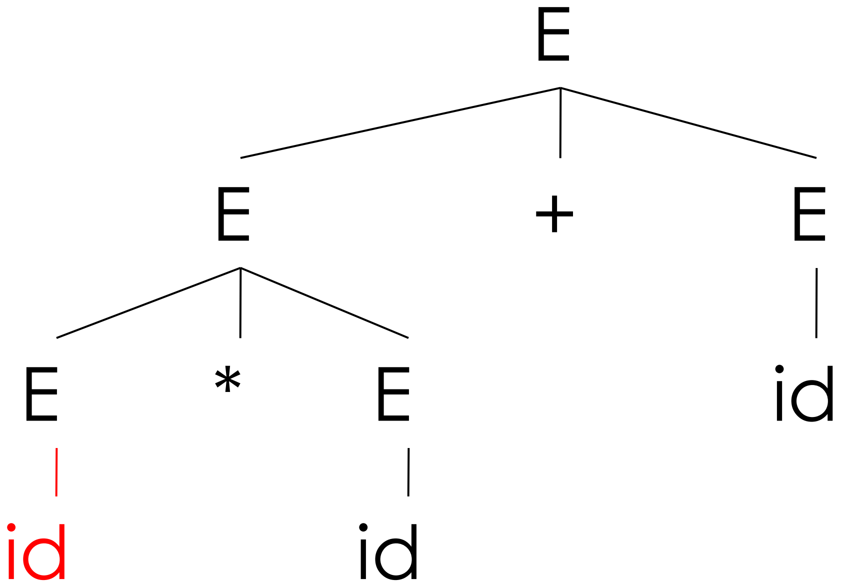
Derivations

$$
\begin{aligned}
& E \\
\rightarrow\quad & E+E \\
\rightarrow\quad & E+\mathrm{id} \\
\rightarrow\quad & E * E + \mathrm{id} \\
\rightarrow\quad & E * \mathrm{id} + \mathrm{id}
\end{aligned}
$$



Derivations

$$\begin{aligned} & E \\ \rightarrow\ & E+E \\ \rightarrow\ & E+\mathrm{id} \\ \rightarrow\ & E * E + \mathrm{id} \\ \rightarrow\ & E * \mathrm{id} + \mathrm{id} \\ \rightarrow\ & \mathrm{id} * \mathrm{id} + \mathrm{id} \end{aligned}$$



Derivations

Note that right-most and left-most derivations have the same parse tree

Derivations

Which of the following is a valid derivation of the given grammar?

S → aXa
X → ε | bY
Y → ε | cXc | d

○
S
aXa
abYa
acXca
acca

○
S
aa

○
S
aXa
abYa
abcXca
abcbYca
abcbdca

○
S
aXa
abYa
abcXcda
abccda

Derivations

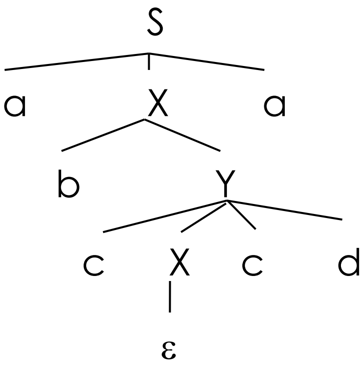
Which of the following is a valid parse tree for the given grammar?

S → aXa

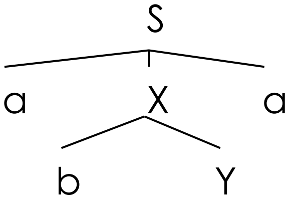
X → ε | bY

Y → ε | cXc | d

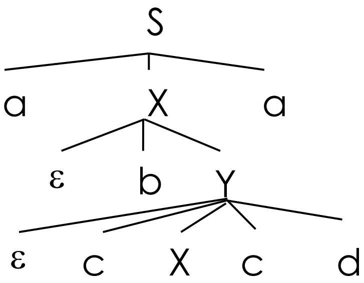
○



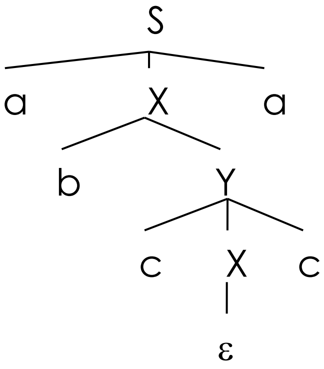
○



○



○



Derivations

- We are not just interested in whether $s \in L(G)$
  - We need a parse tree for s
- A derivation defines a parse tree
  - But one parse tree may have many derivations
- Left-most and right-most derivations are important in parser implementation