

**EXERCISE #2 ► XML Files**

We give the following grammar:

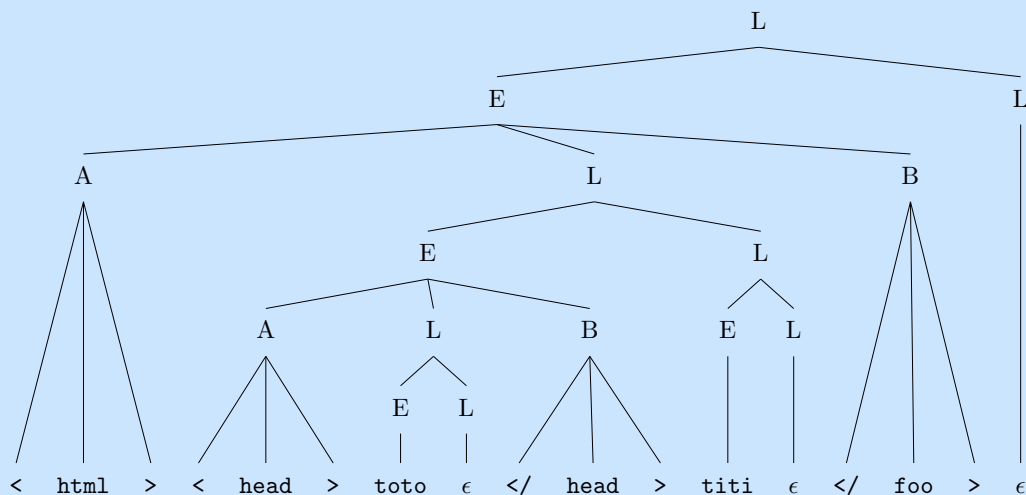
```

L -> E L
|
E -> A L B
|   ident
A -> < ident >
B -> </ ident >

```

1. Give the derivation tree for the chain `<html><head>toto</head>titi</foo>`.
2. Attribute this grammar to verify that opening and closing tags refer to the same identifiers.

**Solution.** The derivation tree is the “parsing tree”, thus still contains non terminal tokens of the grammar (picture by S. Michelland, M1IF 2018):



For the attribution, we can imagine a lot of possibilities, but a solution **MUST** use *grammar attributes*, that are **typed** new data derived from elements of the grammar, and propagated along it (thus no solution with a global structure is valid for this exercise). For instance, let us attach an attribue **ch of type string** to the non terminal A as well as B. This attribute will store the information derived from the parsing chain “ident”. Now we are able to attribute the grammar:

```

L -> E L
|
E -> A L B      { if A.ch <> B.ch failwith <some relevant error message> }
|   ident
A -> < ident >   { A.ch <- ident.strval() }
B -> </ ident > { B.ch <- ident.strval() }

```

Solutions that propagate a boolean from the non terminal *E* to the root can also be designed.

In ANTLR/Python, we could implement this attribution via semantic actions with synthesized attributes. See for instance the code of A. Duderme on his gitlab:

<https://gitlab.aliens-lyon.fr/adudorme/cap/tree/master/xml>

□