

# Exercises on Compilers

Jordi Cortadella

February 7, 2022

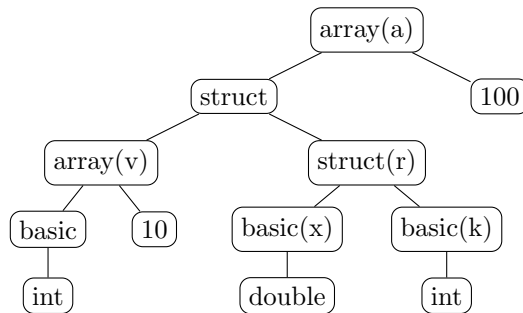
## Attribute grammars

1. Write an attribute grammar to specify binary numerals. A binary numeral is a non-empty sequence of binary digits followed by a period and another non-empty sequence of binary digits. The grammar must calculate an attribute that returns the value of the binary numeral (a real number). Use the following context-free grammar:

$$\begin{aligned} \text{Num} &\rightarrow \text{Digits } '.' \text{ Digits} \\ \text{Digits} &\rightarrow \text{Bit} \mid \text{Digits Bit} \\ \text{Bit} &\rightarrow '0' \mid '1' \end{aligned}$$

2. Write an attribute grammar to construct a syntax tree for data types with arrays and structs (see example). Assume that only `int` and `double` basic types are used.

```
struct {  
  int [10] v;  
  struct {  
    double x;  
    int k;  
  } r;  
} [100] a;
```



Synthesize an attribute that indicates the size of the data structure, assuming that an `int` takes 4 bytes and a `double` takes 8 bytes.