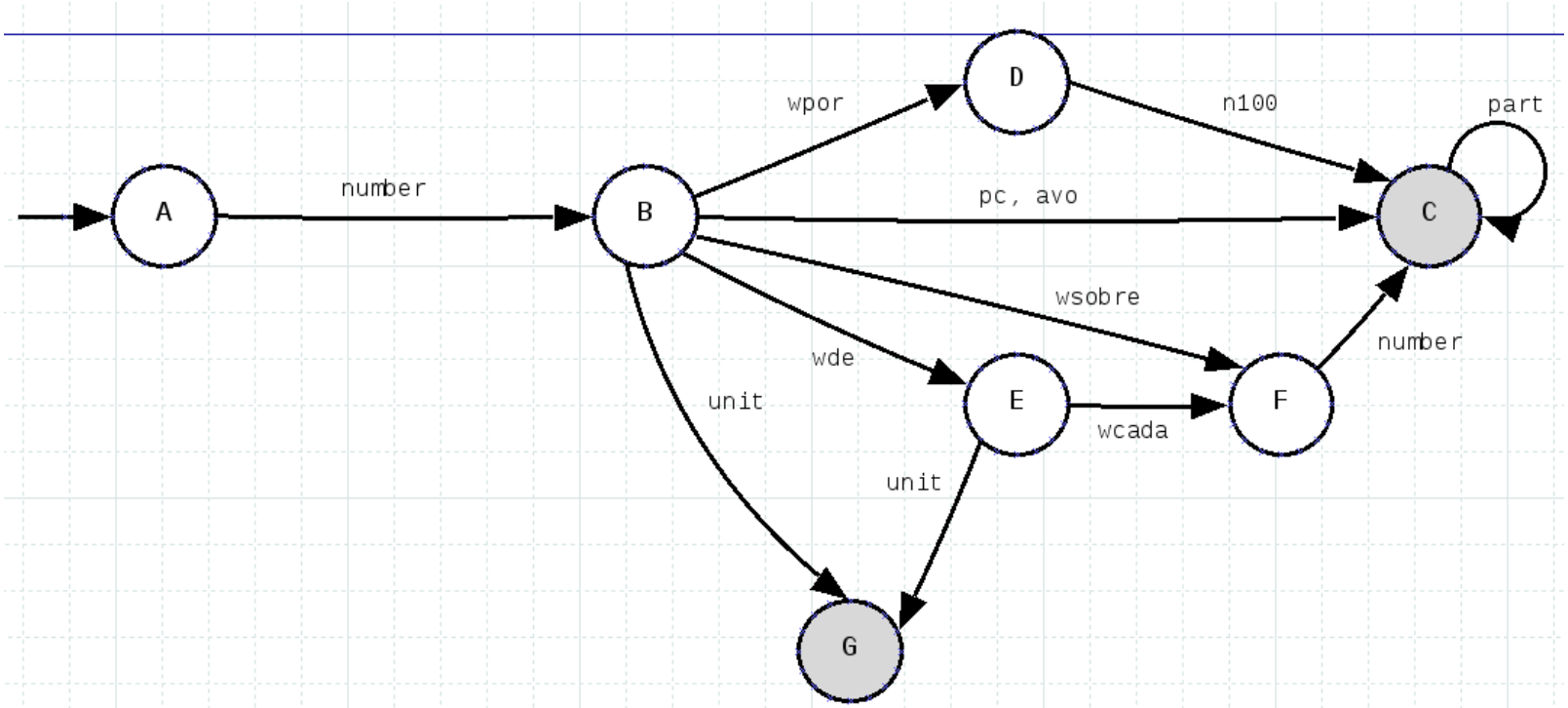
The ATNLang interpreter

By Daniel Otero and
Guido Arnau

Presentation Index

- ➔ What is and ATN?
- ➔ Our language ATNLang
- ➔ Powerful examples

What is an ATN?



What is an ATN?

- ➔ Recursively call other ATNs
- ➔ Indeterministic (explore solutions by backtracking)

Our language - ATNLang

- ➔ Variables
- ➔ Flow control
- ➔ Functions
- ➔ ATNs

ATNLang - Variables

- Basic Types:
  - Integer
  - String
  - Boolean
- Arrays
- Strong dynamically typed

ATNLang - Variables

```
a = 23;

a = true;

a = "hello";

a[5] = 9;

a = {1,2,3,4,5,6,7};
```

ATNLang - Variables

➔ Variable scopes:

- Global
- Local
- ATN local
- ATN arc local

ATNLang - Control flow

- ➔ If … else (greedy)
- ➔ While
- ➔ For

ATNLang - Control flow

```
if (a > 2) print "hola", "%n";
else print "adeu", "%n";


while (a < 10) {
        a = a+1;
}

for (a = 0; a < b.length; a = a+1)
    print b[a], " ";
```

ATNLang - Functions

- ➔ Dynamic return function data type
- ➔ Parameters by value or reference

ATNLang - Functions

```
def fib(n) {
    fib = 0;
    next = 1;
    i = 0;
    while (i < n) {
        print fib, " ";
        aux = fib;
        fib = next;
        next = aux + next;
        i = i+1;
    }
}
```

ATNLang - ATNs

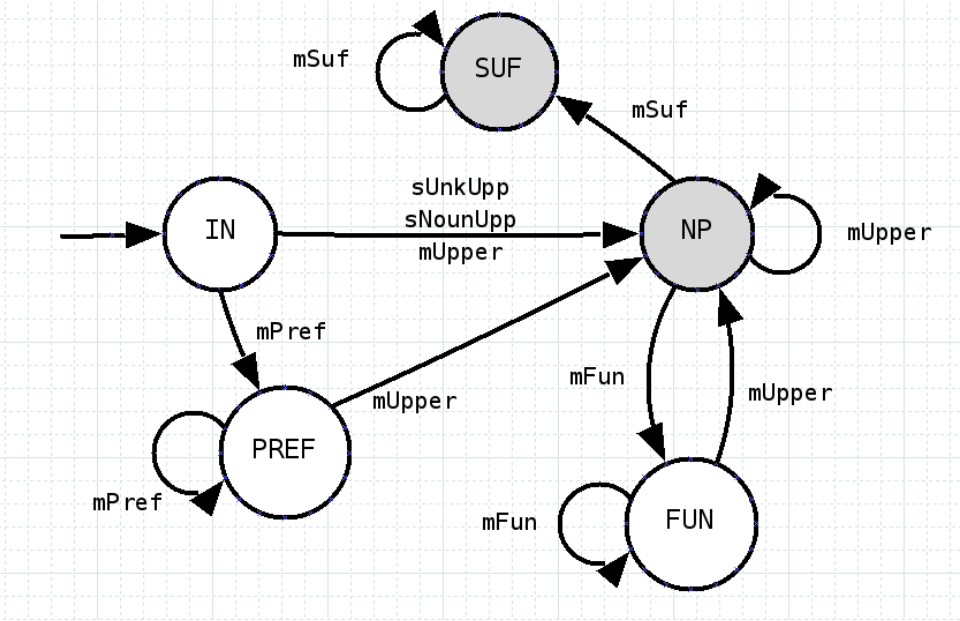
```
atn Atn_name {
          atn_var = ...

          node Node_name {
                  arc (boolean_expression) goto Node_name2 {
                      instruction;
                      ...
                  }
                  arc (...) goto Node_name1 {...}
                  ...
          }
        ....
}
```

Powerful examples

- ➔ Backtracking
- ➔ Sales interpreter
- ➔ Name recognizer
- ➔ Unit recognizer
- ➔ Number recognizer

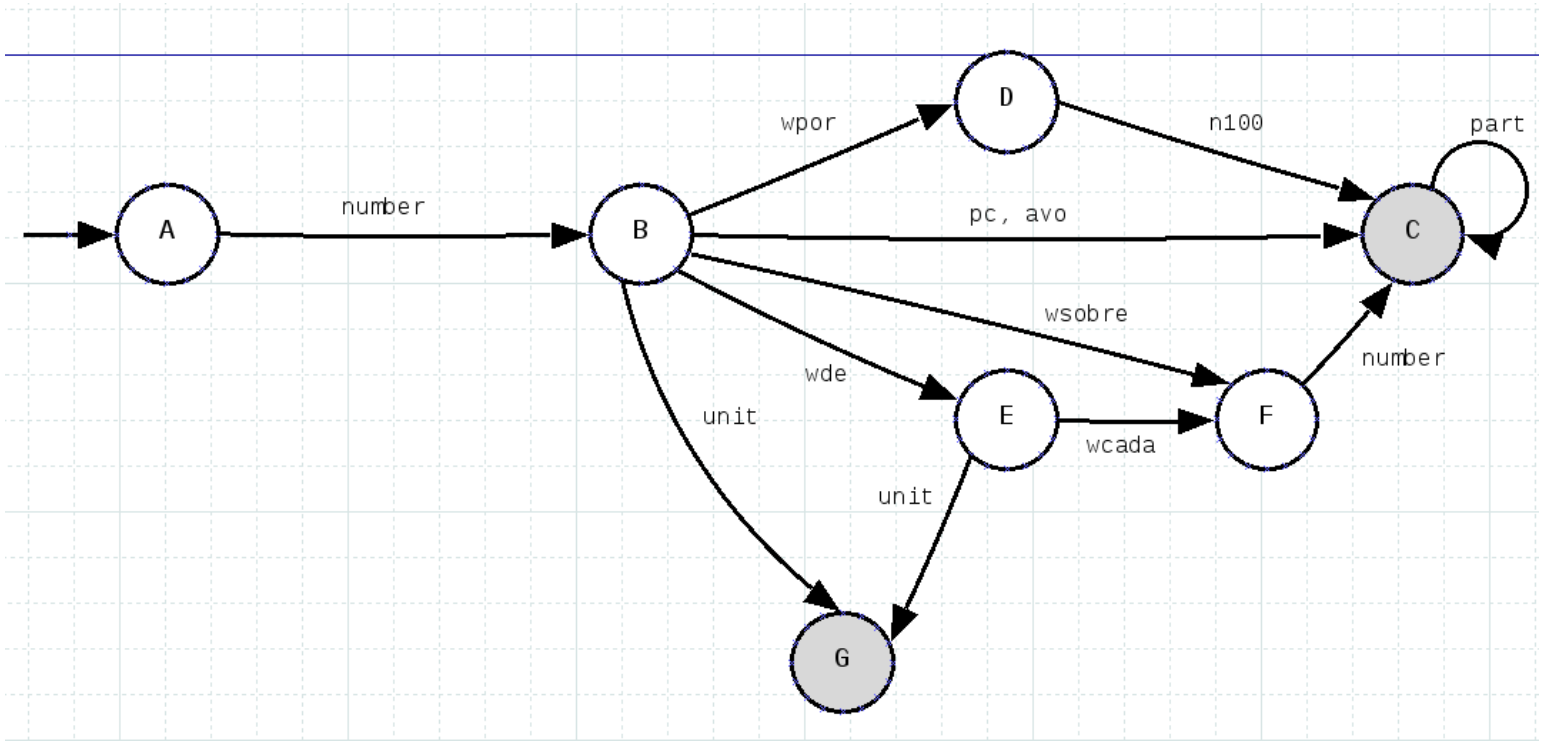
Name recognizer




```
atn np {
    pre = {"dr", "sr", "de"};
    fun = {"de", "la", "el"};
    suf = {"junior", "senior"};

    node IN {
        arc (isUpper(#.0)) goto NP {
            nombre = nombre + #.0 + " ";
        }
        arc (member(#.0, pre)) goto PREF {
            nombre = nombre + #.0 + " ";
        }
        arc(true) goto IN;
    }
    node PREF {
        arc (member(#.0, pre)) goto PREF {
            nombre = nombre + #.0 + " ";
        }
        arc (isUpper(#.0)) goto NP {
            nombre = nombre + #.0 + " ";
        }
        arc(true) goto IN {
            nombre = "";
        }
    }
    node NP {
        arc (isUpper(#.0)) goto NP {
            nombre = nombre + #.0 + " ";
        }
        arc (member(#.0, suf)) goto SUF {
            nombre = nombre + #.0 + " ";
        }
        arc (member(#.0, fun)) goto FUN {
            nombre = nombre + #.0 + " ";
        }
        arc (true) goto END;
    }
    node SUF {
        arc (member(#.0, suf)) goto SUF {
            nombre = nombre + #.0 + " ";
        }
        arc (true) goto END;
    }
    node FUN {
        arc (member(#.0, fun)) goto FUN {
            nombre = nombre + #.0 + " ";
        }
        arc (isUpper(#.0)) goto NP {
            nombre = nombre + #.0 + " ";
        }
    }
    node END accept;
}
```

Unit recognizer



Number recognizer