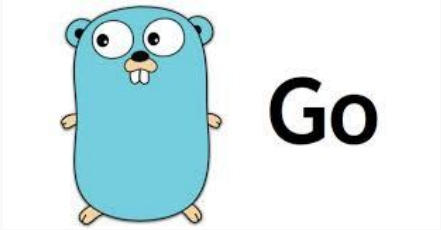
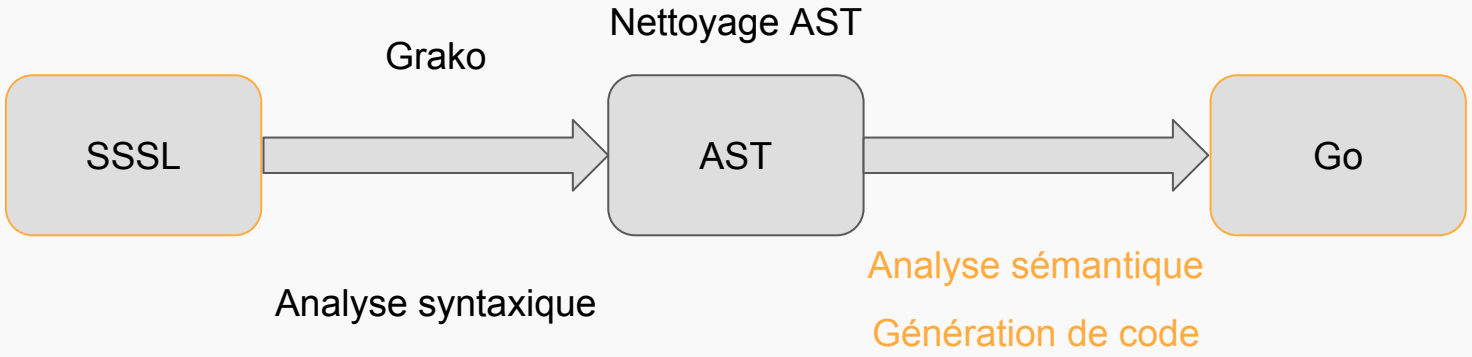
Speeding Simplified Script Language

Adèle BERTRAND-DALECHAMPS
Florian POPEK
Wei WEI

Tuteurs : Olivier RICHARD et Didier DONSEZ





int a = 4;

```
"DECLAFF": [
  {
    "TYPE": "int",
    "NAME": "a"
    "EXPR": 4
  }
]
```

var a int = 4

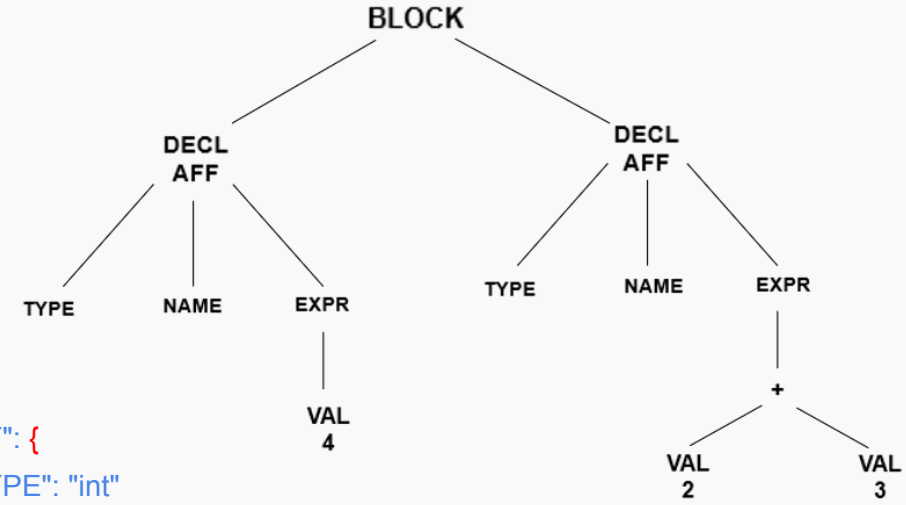
```
class MaClasse {                        // déclaration de classe
        int a                           // déclaration de variable
        float b = 1.2                   // déclaration + affectation
        MaClasse(int arg1, int arg2) {          // constructeur
                int temp = arg1 + arg2
                if (temp > 5) {         // test
                        a = 5           // accès à l'attribut de classe
                }
                else {
                        a = temp
                }
        }
        func int getA() {
                return this.a
        }
}
```

SSSL

```
func int maFonction() {                 // déclaration de fonction
        return 5
}

void Main() {
        int a = maFonction()            // appel de fonction
        MaClasse object = MaClasse(1, a) // construction d'objet
        echo(object.getA())             // accès
}
```

Abstact Syntax Tree

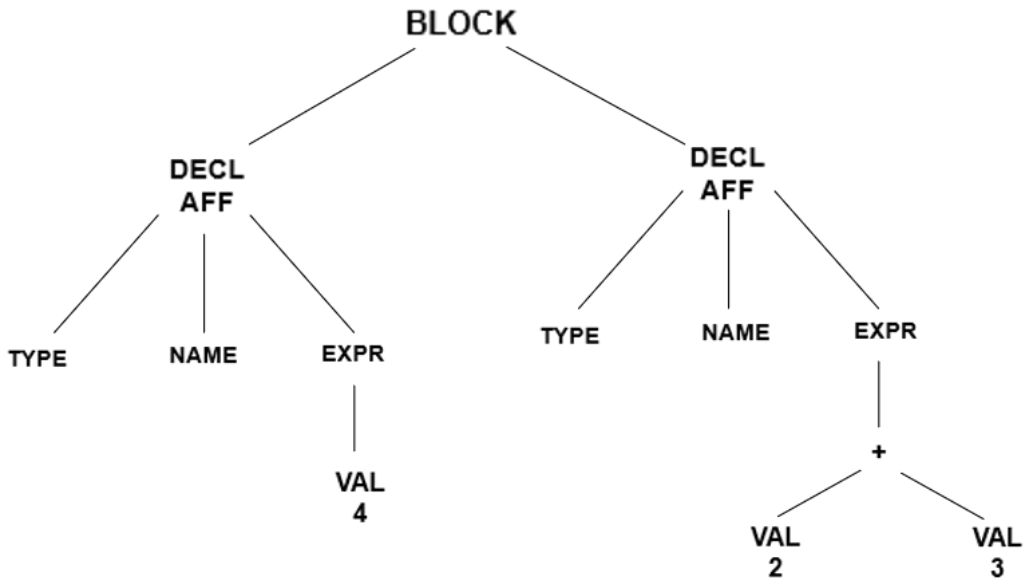



```
"BLOCK": [
    "DECLAFF":
        "TYPE": "int"
        "NAME": "a"
        "EXPR":
                "VAL": "4"
    "DECLAFF":
        "TYPE": "int"
        "NAME": "b"
        "EXPR": [
                "VAL": "2"
                "+"
                "VAL": "3"
        ]
]
```

```
"DECLAFF": {
        "TYPE": "int"
        "NAME": "a"
        "EXPR":  {
                "VAL": "4"
        }
},
"DOBJT": null,
"DMAIN": null,
"DFUNC": null,
"DECL": null
```

Nettoyage

Parsing de l'AST




```python
class Declaration(Node):
    def __init__(self, data):
        Node.__init__(self, data)

        self.type = Type(data)
        self.name = Name(data, False)

        self.type.fill()
        self.name.fill()
```

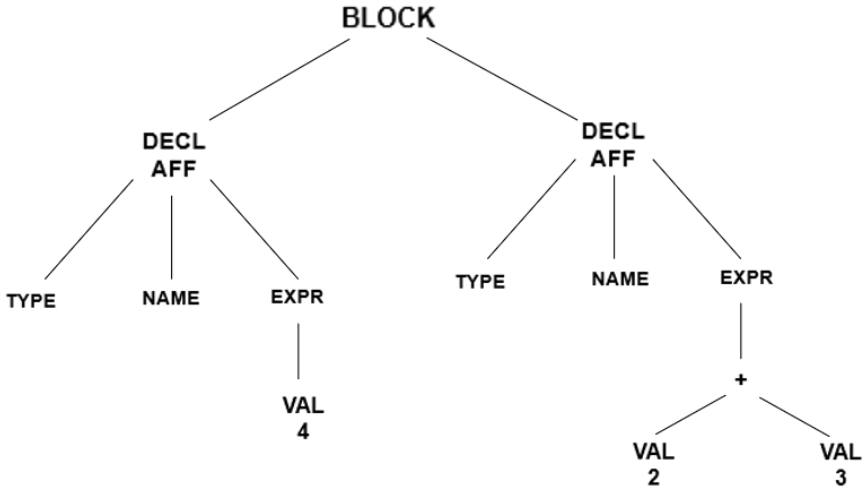
```python
class Declaffectation(Node):
    def __init__(self, data):
        Node.__init__(self, data)

        self.type = Type(data)
        self.name = Name(data, False)
        self.expr = Expression(data)

        self.type.fill()
        self.name.fill()
        self.expr.fill()
```

Analyse sémantique

Déclarations doubles
Utilisation de variables / fonctions inconnues
Types / classes inconnus
Incompatibilité de type (arguments, affectations)



Génération de code

self.__go__()

```
def __go__(self):
    return "var " + self.name.__go__() + " " + self.type.__go__()
Declaration.__go__ = __go__
```

Améliorations

- Objets
- Commentaires
- Lecture et ecriture au clavier
- Passage des arguments par valeur ou par référence
- Caractères speciaux
- Plus de types primitifs
- Surcouche
- Tests automatiques
- Héritage et polymorphisme, accesseurs et getters
- Import, package

Merci pour votre attention

Adèle BERTRAND-DALECHAMPS
Florian POPEK
Wei WEI