

---

## LOG6302A — Analyse d'applications et Cyber-sécurité

Laboratoire #2

Donné le : Feb 01 2024, 09 :30 AM

Échéance : Feb 15 2024, 09 :30 AM

---

- Une archive contenant le code et un rapport (PDF) doit être rendu sur Moodle avant la date limite.
  - Le rapport doit rendre compte de ce que vous avez fait et les problèmes rencontrés. Vous pouvez discuter de tout autre élément que vous jugeriez pertinent.
  - Chaque jour de retard entraîne une pénalité de 50%.
  - Si vous avez des questions, vous pouvez demander des clarifications sur Discord (#lab-question)
- 

## 1 Construction du CFG

Extrayez le CFG intra-procédural des exemples de code fournis à l'aide d'un visiteur. Le CFG devrait commencer par un noeud Entry et se terminer par un noeud Exit. Le graphe inter-procédural n'est pas demandé. Des exemples de CFG sont fournis sur la pages suivantes.  
- *dossier part1* -

## 2 Utilisation d'un CFG

### Détection de code mort

En utilisant les graphes fournis, créer un visiteur capable de trouver :

- du code mort intra-procedural :  
c'est à dire du code qui ne sera jamais executé sur une même procédure.
- du code mort au niveau inter-procedural :  
des fonctions definies mais jamais appelées

Essayer ensuite de trouver du code mort dans Wordpress dans les deux situations.

- *dossier part2/code\_mort* -

## Appel de fonction

```
1 mysql_query('SELECT *');
```

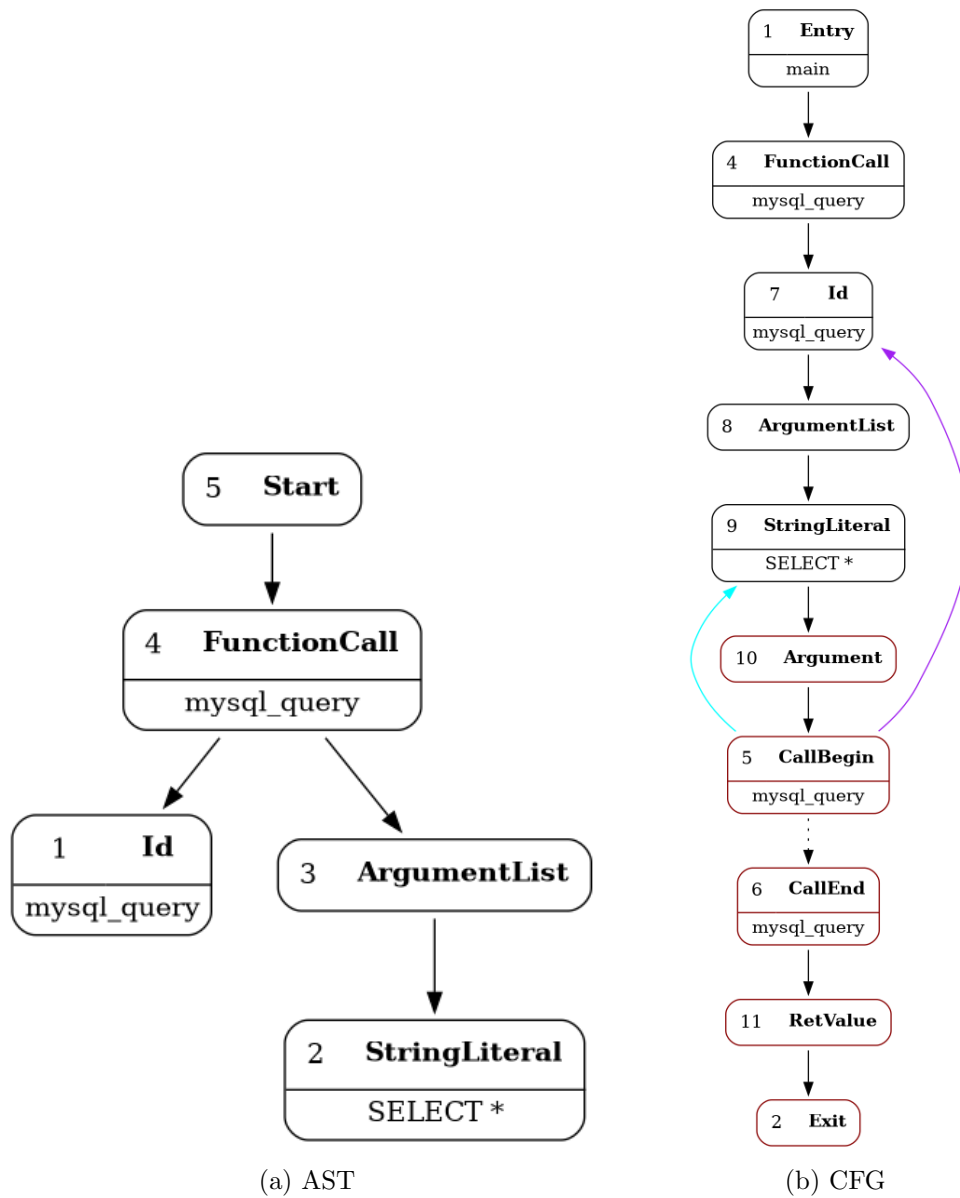


FIGURE 1 – FunctionCall

## Bloc if

```

1
2 $a = 10;
3 if($a < 5) {
4     echo "True";
5 }
6 else {
7     echo "False";
8 }

```

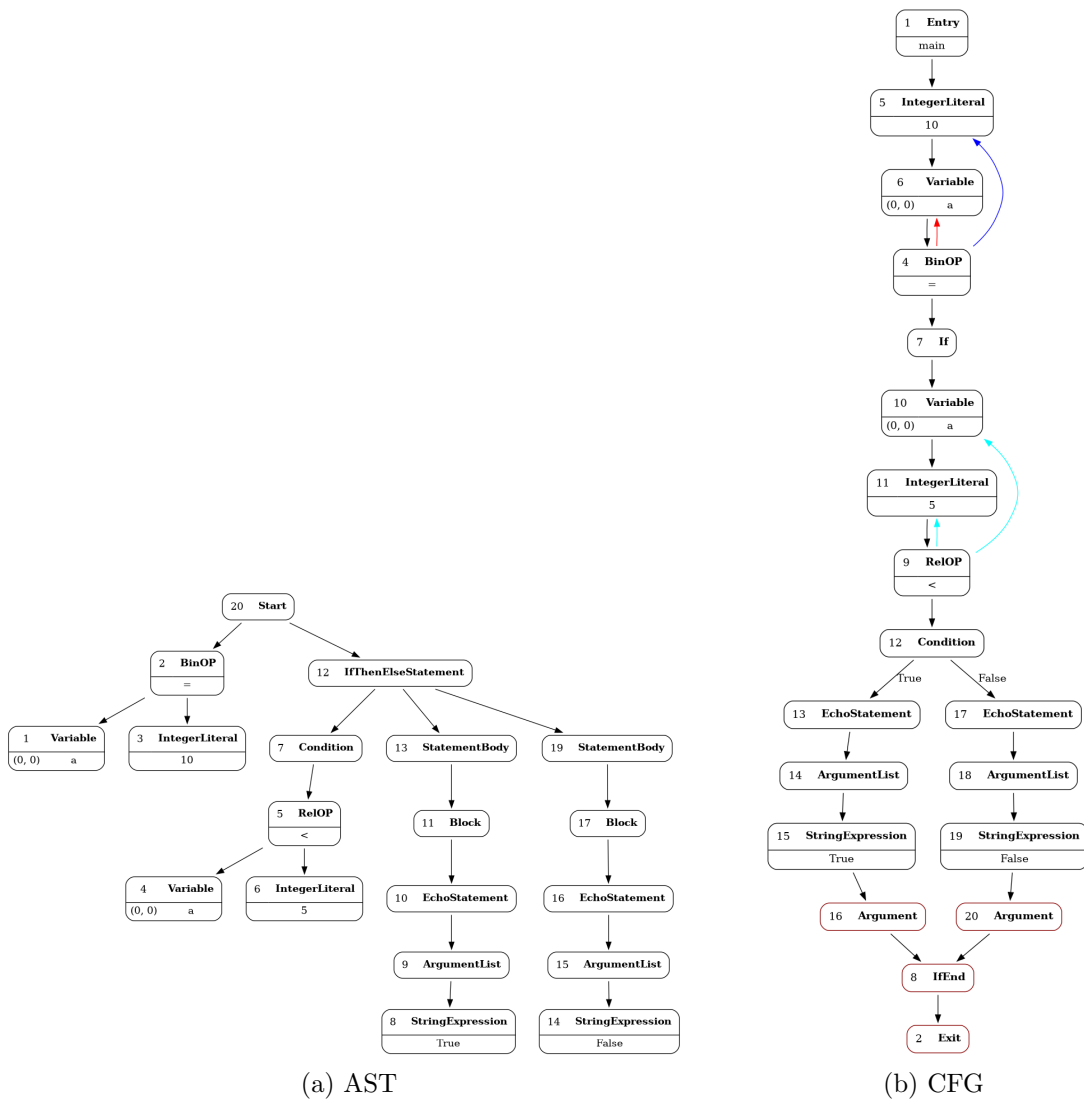


FIGURE 2 – If

## Bloc while

```

1 $i = 0;
2 while($i < 10) {
3     $i = $i + 1;
4 }
5 echo "Done";

```

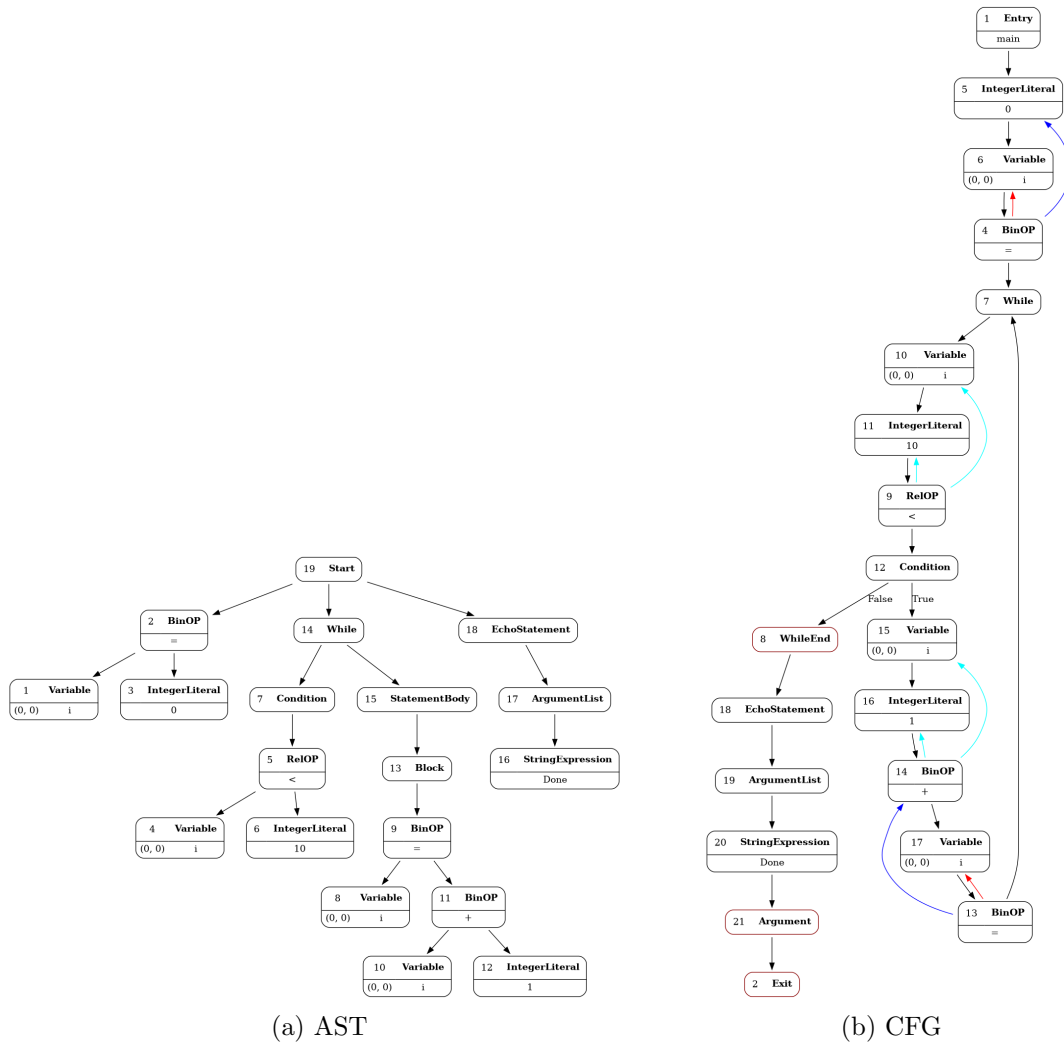


FIGURE 3 – While

## Break / Continue

Après avoir générer le CFG du *while*, ajoutez la gestion des instructions *break* et *continue* au sein des boucle While.