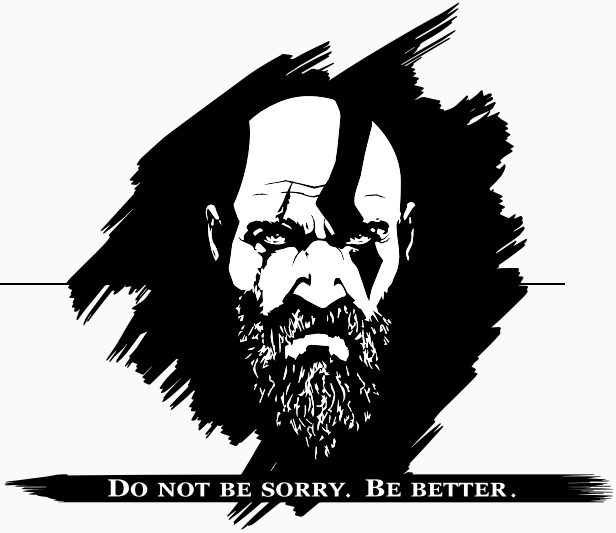
42sh - Presentation

ACU 2019 Team



Copyright

Rules

- You must have downloaded your copy from the Assistants' Intranet <https://intra.assistants.epita.fr>.
- This document is strictly personal and must **not** be passed on to someone else.
- Non-compliance with these rules can lead to severe sanctions.

General Tree Traversal

General Tree Reminder

```c
struct ast_node
{
    enum ast_node_type type;   // type of the node
    size_t nb_children;        // size of children
    struct ast_node *children; // array of children
};
```

General Tree Traversal

```c
void tree_traversal(struct ast_node *node)
{
    print_node(node);
    for (size_t index = 0; index < nb_children; index++)
        tree_traversal(node->children[index]);
}
```

General Tree Visitor

```c
int execute(struct ast_node *node)
{
    switch (node->type)
    {
        case NODE_IF:
            return execute_if_node(node);
        case NODE_COMPOUND_LIST:
            return execute_compound_list(node);
        /* ... */
    }
}
```

General Tree Node Executor

```c
int execute_if_node(struct ast_node *node)
{
    if (execute(node->children[0]))         // Condition is the first child
        return execute(node->children[1]);  // If Body is the second child
    if (node->size == 3)
        return execute(node->children[2]);  // Else Body is the third child if it exist
    return 0;
}
```

Object Oriented Tree Traversal

Object Oriented Tree Reminder

```c
struct ast_node_if
{
    struct ast_node_compound_list *condition; // the condition
    struct ast_node_compound_list *if_body;   // the body of the if
    struct ast_node_compound_list *else_body; // the body of the else, may be NULL
};

struct ast_node_shell_command
{
    enum shell_command_child_type type;
    union shell_command_child child;
};
```

Object Oriented Tree Traversal

```c
void ast_node_if_traversal(struct ast_node_if *node)
{
    print_if(node);
    ast_node_compound_list_traversal(node->condition);
    ast_node_compound_list_traversal(node->if_body);
    if (node->else_body != NULL)
        ast_node_compound_list_traversal(node->else_body);
}

void ast_node_shell_command_traversal(struct ast_node_shell_command *node)
{
    print_shell_command(node);
    switch (node->type)
    {
        case FOR:
            return ast_node_for_traversal(node->child);
        /* ... */
    }
}
```

Object Oriented Tree Node Executor

```c
int execute_if_node(struct ast_node_if *node)
{
    if (execute_compound_list(node->children[0]))         // Condition is the first child
        return execute_compound_list(node->children[1]);  // If Body is the second child
    if (node->size == 3)
        return execute_compound_list(node->children[2]);  // Else Body is the third child if it exist
    return 0;
}
```

Questions

Questions?