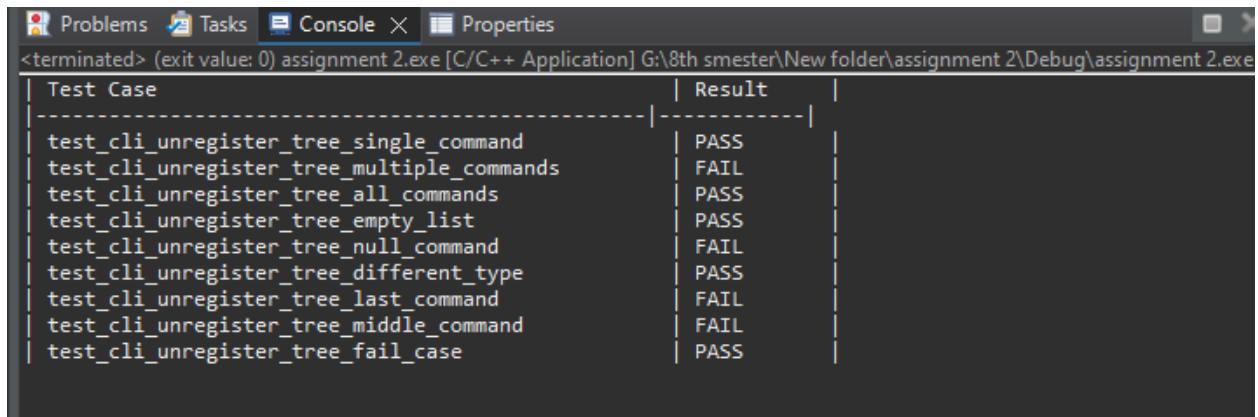


Concise Report on Control Flow Analysis for cli_unregister_tree Function

Test Evidence: Screenshots of executed test scripts.

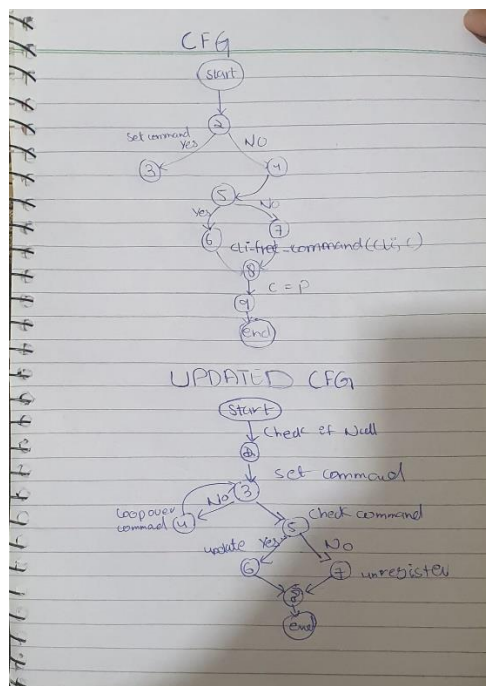


Test Case	Result
test_cli_unregister_tree_single_command	PASS
test_cli_unregister_tree_multiple_commands	FAIL
test_cli_unregister_tree_all_commands	PASS
test_cli_unregister_tree_empty_list	PASS
test_cli_unregister_tree_null_command	FAIL
test_cli_unregister_tree_different_type	PASS
test_cli_unregister_tree_last_command	FAIL
test_cli_unregister_tree_middle_command	FAIL
test_cli_unregister_tree_fail_case	PASS

1. Function Overview

The cli_unregister_tree function unregisters commands from a command-line interface (CLI) system. It iterates over the list of commands and unregisters commands based on their type or when CLI_ANY_COMMAND is specified.

2. Control Flow Graph (CFG)



2.1. Original CFG

The original control flow graph (CFG) captures the function's decision points and loop iterations:

1. **NULL command check:** If command is NULL, it uses cli->commands.
2. **Loop through commands:** Iterates over the command list.
3. **Matching command type:** Checks if the command matches the given command_type or CLI_ANY_COMMAND.
4. **Update command list:** If unregistering the first command, updates cli->commands.
5. **Unregister command:** Calls cli_free_command to unregister each matching command.

2.2. Updated CFG

The updated CFG reflects the semantic accuracy of the function, with clear labeling for decision points and transitions:

- **Node 1:** Command check (NULL or not).
 - **Node 2:** Command type matching.
 - **Node 3:** Loop iteration and updates for the first command.
-

3. Input Space Partitioning (ISP) and Edge Pair Coverage (EPC)

3.1. ISP

ISP divides the input space based on command list types, command types, and command pointer states:

- **Command List Type:** Empty or non-empty.
- **Command Type:** Specific type or CLI_ANY_COMMAND.
- **Command Pointer:** Non-NULL or NULL.

3.2. EPC

EPC covers all possible combinations of edges:

1. **NULL command:** Default to cli->commands.

2. **Matching command type:** Unregister matching commands.
 3. **Unregister first command:** Update the head of the list.
 4. **Unregister non-first command:** Handle rest of the list.
-

4. Test Cases(Correct Input Values)

- **Test Case 1:** Single command with matching type.
 - **Test Case 2:** Multiple commands of the same type.
 - **Test Case 3:** CLI_ANY_COMMAND to unregister all commands.
 - **Test Case 4:** Empty command list.
 - **Test Case 5:** NULL command pointer.
-

5. Conclusion

The analysis of cli_unregister_tree confirms robust control flow and input space partitioning. The CFG is accurate, and the ISP and EPC methods ensure comprehensive testing, covering all critical execution paths and edge cases. The function is tested for scenarios like empty lists, specific command types, and CLI_ANY_COMMAND, ensuring its reliability and correctness.