Haseeb Ahmed (FA21-BCS-042)

Compiler Construction Lab Terminal Q5

The **semantic analysis** functionality in this compiler is distributed across multiple methods. The constructor orchestrates the overall analysis by iterating over the file and invoking specific methods for different semantic checks.

Here is a detailed breakdown of how semantic analysis is performed:

**1. Constructor: Semantique(FILE\* SnailFile)**

* **Purpose:** Serves as the entry point for semantic analysis. It reads the file character by character, invoking specific methods based on the content.
* **Highlights:**
  + Reads characters using getc(SnailFile) until the end of the file (EOF).
  + Calls methods like checkOperator, checkCommenaireEtEnd, and checkprintf for targeted semantic rules.
  + Manages and builds a buffer manyC to store and process sequences of characters (potentially keywords, identifiers, or literals).

**2. checkOperator(char caractere)**

* **Purpose:** Likely validates operators in the source code.
* **Functionality:**
  + Checks if the current character represents a valid operator.
  + Ensures proper usage of operators in the source file.

**3. checkCommenaireEtEnd(char caractere, FILE\*& SnailFile)**

* **Purpose:** Handles comments and identifies end-of-file or other terminal markers.
* **Functionality:**
  + Detects comment delimiters (e.g., /\*, //) and skips over the comment content.
  + May check for syntax correctness of comments and ensure they terminate properly.

**4. checkprintf(char caractere, FILE\*& SnailFile)**

* **Purpose:** Validates the semantics of printf statements in the source code.
* **Functionality:**
  + Detects printf or similar keywords.
  + Ensures that its usage complies with expected syntax and rules (e.g., parentheses, format specifiers).

**5. check\_key(char\* key)**

* **Purpose:** Checks if the given sequence of characters matches a keyword.
* **Functionality:**
  + Matches key against a list of reserved keywords.
  + Ensures keywords are used in proper contexts (e.g., if, while).

**Flow of Semantic Analysis:**

1. **Character Reading:** The constructor reads characters from the file and determines their type (operators, keywords, or identifiers).
2. **Delegated Analysis:** Each identified type is passed to a corresponding method for detailed semantic checks.
3. **Buffer Management:** Characters are temporarily stored in buffers like manyC to assemble complete tokens for analysis.

**Semantic Analysis Method: Constructor as the Coordinator**

The constructor serves as the central function coordinating semantic analysis by:

* Parsing the file sequentially.
* Invoking specific methods for different semantic constructs.
* Managing state variables like intilg, TurnMe, and doit (likely for conditional processing or state tracking).