# Parsing Agent Specification

## Input Specifications

- Type: Entire SAS script.  
- Format: .sas file or plain text input string.  
- Encoding: UTF-8 (default), optionally configurable.

## Detailed Functionality Requirements

- Primary Task: Chunk the SAS code into logical blocks.  
- Sub-tasks:  
 - Identify block types: DATA, PROC, MACRO, %MACRO, INCLUDE, LIBNAME, OPTIONS, etc.  
 - Normalize indentation and spacing for consistency.  
 - Assign metadata tags to each block (e.g., type, line range, purpose).  
- Features:  
 - Skip empty lines or comment-only lines during parsing.  
 - Detect incomplete or nested blocks (e.g., macros containing data steps).

## Output Specifications

- Format: JSON list of blocks.  
- Structure:  
{  
 "block\_type": "DATA",  
 "block\_name": "data\_cleaning",  
 "lines": ["data data\_cleaning;", "...", "run;"],  
 "line\_range": [1, 25]  
}

## Methodology for Chunking

- Approach: Regex-driven line-by-line scanning.  
- Logic:  
 - Use regex patterns to match keywords (data, proc, run;, %macro, quit;, etc.).  
 - Maintain a block-level context stack to handle nested blocks.  
 - Collect lines between block initiators and terminators.  
- Algorithm: Finite State Machine (FSM) to switch between parsing states (e.g., IN\_DATA\_BLOCK, IN\_MACRO\_BLOCK, etc.).