

Especificação da Linguagem SimpleSCL: Uma Versão Simplificada da SCL (IEC 61131-3) para Fins Didáticos

1st Eliézer Carretin

Aluno da instituição

Fundação Hermínio Ometto, Uniararas
Araras, Brasil

eliezercarretin@alunos.fho.edu.br

2nd Iago Bernardino de Almeida

Aluno da instituição

Fundação Hermínio Ometto, Uniararas
Araras, Brasil

iagoalmeida@alunos.fho.edu.br

3rd Renato Luciano Cagnin

Docente da instituição

Fundação Hermínio Ometto, Uniararas
Araras, Brasil

renatocagnin@fho.edu.br

I. ESPECIFICAÇÃO DA LINGUAGEM SCL SIMPLIFICADA (INTERSCS)

A linguagem InterSCL, desenvolvida como parte deste trabalho, constitui uma simplificação da Structured Control Language (SCL) definida na norma IEC 61131-3, com foco em didática e portabilidade. O objetivo principal foi criar um ambiente de simulação leve e independente de plataformas industriais proprietárias, permitindo a execução e validação de lógicas de controle básicas.

A gramática da InterSCL foi concebida para cobrir os elementos essenciais da SCL para fins educacionais e de prototipagem, omitindo algumas das complexidades presentes na especificação completa da IEC 61131-3 que não são estritamente necessárias para a compreensão dos fundamentos da programação de CLPs.

A seguir, são detalhadas as principais construções sintáticas e semânticas suportadas pela InterSCL. Não-terminais são representados em **negrito**, enquanto terminais (tokens) são apresentados em *itálico*. O símbolo \$ denota uma produção vazia.

A. Estrutura Geral do Programa

Um programa InterSCL consiste em um bloco principal que encapsula declarações de variáveis e a lógica de execução.

- **Programa** → **BLOCO** **Corpo** **END_BLOCK**
- **Corpo** → **Declaracoes** **Comandos**

B. Declarações de Variáveis

A linguagem suporta expressões aritméticas, lógicas e relacionais, seguindo a precedência e associatividade padrão.

- **Declaracoes** → **Declaracoes** **SEMICOLON** **Declaracoes** | \$
- **Declaracao** → **Tipo** **ID**
- **Tipo** → **INT** | **REAL** | **BOOL**

C. Expressões

A linguagem suporta expressões aritméticas, lógicas e relacionais, seguindo a precedência e associatividade padrão.

- **Expressao** → **ExpressaoLogica** (**OR** **ExpressaoLogica**)
- **ExpressaoLogica** → **ExpressaoRelacional** (**AND** **ExpressaoRelacional**)
- **ExpressaoRelacional** → **ExpressaoAritmetica** (**EQ** | **NEQ**) | **LT** | **LE** | **GT** | **GE** | **ExpressaoAritmetica** | **NOT** **ExpressaoAritmetica**
- **ExpressaoRelacional** → **ExpressaoAritmetica** (**EQ** | **NEQ**) | **LT** | **LE** | **GT** | **GE** | **ExpressaoAritmetica** | **NOT** **ExpressaoAritmetica**
- **ExpressaoAritmetica** → **Termo** (**PLUS** | **MINUS** **Termo**)
- **Termo** → **Fator** (**MUL** | **DIV** **Fator**)
- **Fator** → **ID** | **NUMBER_LITERAL** | **BOOLEAN_LITERAL** | **LPAREN** **Expressao** **RPAREN**

D. Comandos de Controle de Fluxo e Atribuição

A InterSCL inclui comandos para atribuição de valores a variáveis e estruturas de controle de fluxo básicas.

- **Comandos** → **Comandos** **SEMICOLON** **Comandos** | \$
- **Comando** → **ID** **ASSIGN** **Expressao** | **ComandoBloco** | **PRINT** **LPAREN** **ID** **RPAREN**
- **ComandoBloco** → **IfStatement** | **WhileStatement** | **ForStatement**
- **IfStatement** → **IF** **Expressao** **THEN** **Comandos** **END_IF** | **IF** **Expressao** **THEN** **Comandos** **ELSE** **Comandos** **END_IF**
- **WhileStatement** → **WHILE** **Expressao** **DO** **Comandos** **END_FOR**

E. Tipos de Tokens (Terminais)

Os seguintes tokens são reconhecidos pelo analisador léxico da InterSCL:

- **BLOCO**, **END_BLOCK**, **IF**, **THEN**, **ELSE**, **END_IF**, **WHILE**, **DO**, **END_WHILE**, **FOR**, **TO**, **END_FOR**, **PRINT** (Palavras-chave)
- **INT**, **REAL**, **BOOL** (tipos de dados)
- **OR**, **AND**, **NOT** (Operadores lógicos)
- **ID** (Identificadores de variáveis)
- **NUMBER_LITERAL** (Números inteiros e reais)
- **BOOLEAN_LITERAL** (**TRUE**, **FALSE**)
- **ASSIGN** (**:=**)
- **EQ**, (**=**), **NEQ** (**<>**), **LT** (**<**), **LE** (**<=**), **GT** (**>**), **GE** (**>=**) (Operadores Relacionais)
- **PLUS** (**+**), **MINUS** (**-**), **MUL** (*****), **DIV** (**/**) (Operadores aritméticos)
- **LPAREN** (**(**), **RPAREN** (**)**)
- **SEMICOLON** (**;**)
- **COMMENT** (**//...** para comentários de linha)

F. Semântica Simplificada

A semântica da InterSCL foi implementada com as seguintes considerações:

- **Tipagem Estática Básica:** As variáveis devem ser declaradas com um tipo antes de serem utilizadas. O sistema realiza uma verificação básica de tipo durante a atribuição e avaliação de expressões para garantir compatibilidade. Conversões implícitas entre **INT** e **REAL** são permitidas quando semanticamente válidas (e.g., atribuição de INT para REAL).
- **Escopo Global:** Todas as variáveis declaradas são consideradas de escopo global, visíveis em todo o programa.
- **Ordem de Execução Sequencial:** Os comandos são executados sequencialmente, com o fluxo de controle sendo alterado apenas pelas estruturas **IF**, **WHILE** e **FOR**.
- **Ausência de Funções e Blocos Compostos:** Para manter a simplicidade, a InterSCL não suporta a definição ou chamada de funções, blocos de código nomeados ou passagem de parâmetros.
- **Sem Integração com Hardware:** O interpretador foca puramente na simulação da lógica, sem interface direta com Controladores Lógicos Programáveis (CLPs) reais ou simuladores industriais externos.
- **Sem Interface Gráfica:** A interação com o interpretador ocorre via console, com a entrada de código SCL como uma string e a saída da tabela de símbolos final após a execução.

Esta especificação demonstra a simplificação aplicada à SCL, tornando-a uma ferramenta didática eficaz para o aprendizado dos conceitos fundamentais de lógica de controle sem a sobrecarga da especificação completa da IEC 61131-3.