

APS — Linguagens de Programação

ChemMic: Linguagem para Simulação de Reações Químicas na MicrowaveVM

A linguagem ChemMic foi desenvolvida com o objetivo de simular reações químicas térmicas dentro da MicrowaveVM — uma máquina virtual minimalista que possui apenas dois registradores de escrita (TIME e POWER), dois sensores de leitura (TEMP e WEIGHT) e um modelo térmico que simula o aquecimento real de um micro-ondas.

A proposta desta APS é demonstrar como é possível criar uma linguagem de alto nível, definir sua gramática (EBNF), construir um compilador com Flex e Bison, e gerar código de baixo nível interpretável pela MicrowaveVM.

Objetivos

- Criar uma linguagem de domínio específico (DSL) voltada à simulação de reações químicas;
- Definir uma gramática formal (EBNF) e realizar a análise léxica e sintática via Flex e Bison;
- Gerar um código assembly (.mwasm) compatível com a MicrowaveVM;
- Validar o compilador executando programas da ChemMic na VM;
- Demonstrar a relação entre linguagem de alto nível e o código de máquina da VM.

Comandos

Comando	Descrição
reaction "Nome" at <temp> for <time>	Executa uma reação mantendo a temperatura alvo (TEMP) por um tempo específico (TIME).
heat to <temp>	Aumenta a potência gradualmente até atingir a temperatura desejada.
cool to <temp>	Aguarda resfriamento natural até que a temperatura fique abaixo do valor indicado.
stir <n>	Simula agitação alternando a potência entre níveis altos e baixos.
wait <t>	Pausa a execução (POWER=0) por um tempo determinado.

Comando	Descrição
measure	Exibe o estado atual dos registradores (TIME, POWER, TEMP).
repeat <n> { ... }	Repete um bloco de comandos n vezes.
halt	Finaliza o programa.

Implementação

A implementação foi feita em C, utilizando: Flex — para análise léxica (lexer.l), responsável por identificar tokens como REACTION, NUMBER, STRING, etc.; Bison — para análise sintática (parser.y), interpretando as regras da gramática e emitindo o código assembly da VM.

Exemplo

Código ChemMic (example.chem)

```
heat to 70
reaction "Evaporation" at 70 for 5
stir 3
measure
halt
```

Código Assembly Gerado (example.mwasm) ;

```
; Heat to 70 (heuristic power 35)
SET POWER 35
; Reaction: Evaporation
SET POWER 35
SET TIME 5
react_loop_1:
    DECJZ TIME react_loop_1_end
    GOTO react_loop_1
react_loop_1_end:
; Stir 3 times
SET TIME 3
stir_2:
    SET POWER 0
    INC TIME
    SET POWER 60
    INC TIME
    DECJZ TIME stir_end_3
    GOTO stir_2
stir_end_3:
PRINT
HALT
```

Saída da MicrowaveVM

```
Loaded program from: example.mwasm
TIME: 0
BEEEEEEP!
Final state: {'TIME': 0, 'POWER': 35}
Final readonly state: {'TEMP': 72, 'WEIGHT': 100}
Final stack: []
```

Conclusão

O projeto ChemMic demonstra a criação completa de uma linguagem de programação funcional, desde a concepção gramatical até a geração de código de máquina. Com apenas dois registradores (TIME e POWER), foi possível implementar estruturas de controle, loops e simulações térmicas, comprovando a Turing-completude da MicrowaveVM e a versatilidade do modelo de compilação Flex/Bison. Este trabalho cumpre todos os requisitos da APS, apresentando uma linguagem original, executável e bem documentada.