



SNOBOL4

TAREFA 02 – ESTRUTURA DE LINGUAGENS

Leonardo Sillas
Vinicius Santiago

INTRODUÇÃO

- SNOBOL 4 (String Oriented Symbolic Language number 4)
- Específica à manipulação de sequências de strings baseada nos princípios de casamento de padrões (pattern matching)
- Desenvolvida entre 1962 e 1967 por D. J. Farber, R. E. Griswold, e F. P. Polensky.

RELAÇÃO COM OUTRAS LINGUAGENS

FOI INFLUENCIADA

- COMIT
- MARKOV
- SCL
- SHADOW(Compilador)

INFLUENCIOU

- AWK
- SL5
- ICON

IMPERATIVA

- Usa declarações que alteram o estado de um programa. Da mesma forma que em linguagens naturais, expressa comandos, um programa imperativo consiste em comandos para o computador executar.
- No SNOBOL4 temos por exemplo:

N1 = 43

D = 17

OUTPUT = N1 + D

60

NÃO-ESTRUTURADA

- No SNOBOL4 vemos como exemplo o conceito de GOTO que altera a ordem de execução do programa através de labels, como no exemplo abaixo:

`N = 0`

`COPY OUTPUT = INPUT :F(DONE)`

`N = N + 1 :(COPY)`

`DONE OUTPUT = 'THERE WERE ' N ' LINES'`

`END`

INTERPRETADA

- A maioria de suas implementações executam instruções diretamente, sem antes compilar um programa em instruções de linguagem de máquina. O interpretador executa o programa diretamente, traduzindo cada declaração para uma sequência de uma ou mais sub-rotinas já compiladas em código de máquina.

DINÂMICA

- As variáveis não precisam ser pré-declaradas e seus valores podem mudar os tipos com facilidade. No exemplo abaixo, J começa como uma string, mas depois é alterado para que contenha um número inteiro; 42 é impresso.

J = "ABC"

A = 20

B = "22"

J = A + B

OUTPUT = J

PATTERN MATCHING (CASAMENTO DE PADRÕES)

- O casamento de padrões examina uma sequência de strings para alguma combinação de caracteres, chamado de “padrão” através da seguinte estrutura:

Label STRING PATTERN

- A string é o primeiro elemento de declaração após o campo de label (se houver). O padrão aparece em seguida, separado da string por espaço em branco.
- O casamento de padrão ocorre se o padrão for encontrado na sequência da string; caso contrário, ele falha.

COMPARAÇÃO DE CÓDIGOS REPRESENTATIVOS (PATTERN MATCHING)

SNOBOL4

```
1 PATTERN = ('TRUTURAS DE LINGUA') . MATCH
2 STRING = 'ESTRUTURAS DE LINGUAGENS'
3 STRING PATTERN
4 OUTPUT = MATCH
5 ENL
```

Program text: unnamed.sno

```
^ TRUTURAS DE LINGUA
```

PYTHON

C:\Windows\System32\cmd.exe - python

```
>>> pattern = "TRUTURAS DE LINGUA"
>>> string = "ESTRUTURAS DE LINGUAGENS"
>>> if pattern in string:
...     print(pattern)
...
TRUTURAS DE LINGUA
>>> █
```

CONTROLE DE FLUXO

A transferência de controle é realizada pelo comando GOTO. Ele interrompe a execução sequencial normal das instruções, informando qual declaração deve ser executada em seguida. O GOTO aparece no final da instrução, e por ser acompanhado por:

- 1 - dois pontos (:), para desvio incondicional;
- 2 - dois pontos + F (:F), para desvio somente se a condição avaliada retornar false (FAIL);
- 3 - dois pontos + S (:S), para desvio somente se a condição avaliada retornar true (SUCCESS).

```
* PROGRAMA QUE CALCULA A MEDIA E O SOMATÓRIO DE N NUMEROS
  I = 0
  SOMA = 0
  TOPO NUMERO = TRIM(INPUT)
  I = I + 1
  SOMA = SOMA + NUMERO
  FIM MEDIA = SOMA / I
  OUTPUT = 'SOMATORIO =' SOMA
  OUTPUT = 'MEDIA =' MEDIA
  END
```

1. :F (FIM)

2. : (TOPO)

No exemplo, a transferência “1.” ocorre apenas se houver falha na atribuição da variável ‘NUMERO’, caso contrário, a execução prossegue até o desvio incondicional “2.” onde retorna a execução do programa para a linha referenciada “TOPO”.

COMPARAÇÃO DE CÓDIGOS REPRESENTATIVOS (CONTROLE DE FLUXO)

SNOBOL4

```
TOPO n = input()
    DIFFER(n,'exit') :S(TOPO) F(FIM)
FIM    END
```

PYTHON

- while True:
- n = raw_input()
- if n.strip() == 'exit':
- break

A Variável 'n' está recebendo uma atribuição e enquanto ela não receber a palavra 'exit', ela continuará recebendo atribuições.

AVALIAÇÃO EM COMPARAÇÃO AO PYTHON

READABILITY

Python possui uma melhor Readability pois:

- Fica mais transparente à quem estiver lendo o que está sendo feito
- No primeiro caso a chamada do método "in" é mais natural
- A utilização do controle de fluxo e da função DIFFER não são tão naturais e óbvias de se entender quanto um loop com while true/break no python.
- O padrão de STRING PATTERN pode confundir quem está lendo sobre o que está acontecendo.

WRITABILITY

Nas duas linguagens verificamos a mesma facilidade e simplicidade para atingir o mesmo objetivo, portanto acreditamos que nesse quesito o writability das duas linguagens é o mesmo.

EXPRESSIVIDADE

- Se uma linguagem exige muito código para expressar uma abstração, ela é menos expressiva
- Em ambas as linguagens a exigência de código para representar as abstrações mencionadas é a mesma
- Portanto, a expressividade de ambas é equivalente.
- Por SNOBOL4 ser muito antiga suas principais características já foram incorporadas pelas linguagens mais modernas

CONCLUSÃO

- Por ser uma linguagem muito antiga, carece de suporte apropriado, seja da própria linguagem ou da comunidade (???). Apesar disso, SNOBOL4 ainda possui vantagens quando o quesito é o tratamento de strings, como por exemplo tipagem dinâmica que fora abordada anteriormente. Foi uma linguagem tão revolucionária na época que influenciou várias linguagens em gerações posteriores e ainda hoje, as linguagens mais modernas possuem algum traço de seu principal diferencial, que foi o pattern matching (casamento de padrões).

Bibliografia

- Slides: <https://github.com/vinivst/EDL/blob/master/slides/edl-01.pdf>
- Wikipedia:
 - <https://en.wikipedia.org/wiki/SNOBOL>
 - [https://en.wikipedia.org/wiki/Icon_\(programming_language\)](https://en.wikipedia.org/wiki/Icon_(programming_language))
 - <https://pt.wikipedia.org/wiki/AWK>
- Hopl:
 - <http://hopl.info/showlanguage.prx?exp=171>
 - <http://hopl.info/showlanguage.prx?exp=783>
- Site da linguagem:
 - <http://www.snobol4.org/>