Máquina Norma

Teoria da Computação

INF05501

A Máquina Norma

- Norma é um acrônimo para *Number theOretic Register MAchine*
- Possui como **memória** um **conjunto infinito de registradores naturais**
- Três **instruções** sobre cada registrador:
  - Adição de um ao valor armazenado (**incremento**)
  - Subtração de um do valor um armazenado (**decremento**)
  - **Teste se** o valor armazenado **é zero**

A Máquina Norma (cont.)

- $\mathbb{N}^\infty$ denota o **conjunto de todas as tuplas com infinitos (mas contáveis) componentes** sobre o conjunto dos números **naturais**

- Para evitar subscritos, as **componentes das tuplas são denotadas por letras maiúsculas** como A, B, X, Y, ..., as quais representam os **registradores da Máquina Norma**

Definição Formal da Máquina Norma

Supondo que K é um registrador, a Máquina Norma N é uma heptupla do tipo:

$$N = (\mathbb{N}^\infty, \mathbb{N}, \mathbb{N}, ent, sai, \{ad_K, sub_K\}, \{zero_K\})$$

onde:

- $\mathbb{N}^\infty$ *é o conjunto de valores de memória, sendo que cada um de seus elementos descreve uma configuração de um dos infinitos registradores, denotados por* $A, B, X, Y, ...$

- $ent : \mathbb{N} \rightarrow \mathbb{N}^\infty$ *é a função de entrada, a qual resulta no armazenamento do* ***valor de entrada no registrador*** *identificado por* X *e inicializa* ***todos os demais registradores com zero***

Definição Formal da Máquina Norma (cont.)

- $sai : \mathbb{N}^\infty \to \mathbb{N}$ *é a função de saída, a qual retorna o valor armazenado no registrador denotado por* Y

- $\{ad_K, sub_K\}$ *é conjunto de interpretações de operações, as quais são* ***indexadas por registradores****, de forma que, para cada registrador* $K \in \{A, B, X, Y, ...\}$*, tem-se que:*

  – $ad_K : \mathbb{N}^\infty \to \mathbb{N}^\infty$ ***incrementa a componente correspondente ao registrador*** K *sem alterar as demais (notação simplificada:* $K := K + 1$*)*

  – $sub_K : \mathbb{N}^\infty \to \mathbb{N}^\infty$ ***decrementa a componente correspondente ao registrador*** K *sem alterar as demais (notação simplificada:* $K := K - 1$*)*

Definição Formal da Máquina Norma (cont.)

- $\{zero_K\}$ *é* *conjunto de interpretações de testes*, *os quais são* ***indexados por registradores***, *de forma que, para cada registrador* $K \in \{A, B, X, Y, ...\}$, *tem-se que:*
  - $zero_K : \mathbb{N}^\infty \to \{\texttt{verdadeiro}, \texttt{falso}\}$ *resulta em* `verdadeiro` *se a componente correspondente ao registrador* K *for zero e* `falso`*, caso contrário (notação simplificada:* $K = 0?$*)*

Máquina Norma como Máquina Universal

- Apesar de ser uma **máquina extremamente simples**, seu poder computacional é, no mínimo, igual ao de qualquer computador moderno

- Reforça esta ideia a **possibilidade de simularem-se características de máquinas reais** usando a Máquina Norma

- Inclusive, **pode-se considerar apenas programa monolíticos**, pois **mecanismos como a recursão podem ser simulados usando fluxogramas na Máquina Norma**

- A seguir, veremos como simular algumas características usando a Máquina Norma

Simulação de Operações e Testes

- Atribuição de valor zero a um registrador A: Obtida através do seguinte **programa iterativo**:

```
até A = 0
faça (A := A − 1)
```

- Pode-se representar tal programa como uma **macro** $A := 0$

- A partir da definição desta macro, pode-se definir **operações de atribuição de qualquer valor** a um registrador

Simulação de Operações e Testes (cont.)

- Atribuição de um valor natural a um registrador: **Generalizando-se** a macro definida anteriormente, tem-se a macro $A := n$, a qual representa a **atribuição de um valor natural** n **a um registrador** A

- **Programa iterativo** para a aplicação da macro para $n = 3$:

$$\begin{array}{l} A := 0; \\ A := A + 1; \\ A := A + 1; \\ A := A + 1 \end{array}$$

Simulação de Operações e Testes (cont.)

- Adição de dois registradores: Podemos denotar a adição do valor de um registrador B a um registrador A através da macro $A := A + B$
- **Programa iterativo** correspondente à macro:

```
até B = 0
faça (A := A + 1; B := B - 1)
```

- Note: esta operação faz com que o registrador B seja zerado

Simulação de Operações e Testes (cont.)

- Adição de dois registradores, preservando B: Para preservarmos o valor B após a adição, precisamos usar um **registrador auxiliar** C, resultado na macro $A := A + B\ usando\ C$:

```
C := 0;
até B = 0
faça (A := A + 1; C := C + 1; B := B − 1);
até C = 0
faça (B := B + 1; C := C − 1)
```

- Note: o registrador C escolhido **não pode ser usado em outra parte do programa** que utiliza esta macro

Simulação de Operações e Testes (cont.)

- Atribuição de conteúdo de um registrador: A atribuição do valor armazenado em um registrador B para um registrador A também se utiliza de um registrador auxiliar, resultando na macro $A := B\ usando\ C$:

$$A := 0;$$
$$A := A + B \ \ usando \ \ C$$

Simulação de Operações e Testes (cont.)

- Multiplicação de dois registradores: Requer dois registradores auxiliares para a aplicação da macro $A := A \times B \ usando \ C, D$:

$C := 0;$
`até` $A = 0$
`faça` $(C := C + 1; A := A - 1);$
`até` $C = 0$
`faça` $(A := A + B \ \ usando \ \ D; C := C - 1)$

Simulação de Operações e Testes (cont.)

- Teste se valor de um registrador é primo: Utiliza-se a macro $teste\_primo(A)$ *usando* C:

```
(se A = 0
então falso
senão C := A;
      C := C − 1;
      (se C = 0
      então verdadeiro
      senão (até teste_mod(A, C)
            faça (C := C − 1));
            C := C − 1;
            (se C = 0
            então verdadeiro
            senão falso)))
```

Simulação de Operações e Testes (cont.)

- Atribuição do n-ésimo número primo a um registrador: Utiliza-se a macro $A := primo(B)\ usando\ D$:

$A := 1;$
$D := B;$
até $D = 0$
faça $(D := D - 1;$
　　$A := A + 1;$
　　até $teste\_primo(A)\ \ usando\ \ C$
　　faça $(A := A + 1))$

Representação de Números: Inteiros

- Um **número inteiro** m pode ser representado como um par ordenado (s,|m|), onde:

  - s define o **sinal** de m, sendo que, se $m < 0$, então $s = 1$ (negativo), senão $s = 0$ (positivo)
  - $|m|$ descreve o valor absoluto de m (**magnitude**)

- Note que **pares ordenados** podem ser representados em Norma conforme a **codificação de n-uplas de naturais** vista na aula passada **ou** usando **dois registradores**

Representação de Números: Inteiros (cont.)

- Representação de números inteiros com dois registradores: Supõe-se que o registrador inteiro A é representado pelo **par ordenado de registradores naturais** (A_1, A_2), onde A_1 armazena o sinal do valor armazenado em A e A_2, a sua magnitude

- Esta representação é chamada de sinal-magnitude

- A fim de trabalhar com esta representação e **executar operações inteiras**, precisamos de **três programas em Norma**, um para **cada operação elementar**

Representação de Números: Inteiros (cont.)

- Para $A := A + 1$, temos o seguinte programa iterativo:

```
(se A₁ = 0
então A₂ := A₂ + 1
senão A₂ := A₂ − 1;
        (se A₂ = 0
        então A₁ := A₁ − 1
        senão ✓))
```

Exercícios

1. Qual é o programa iterativo que corresponde à operação inteira $A := A - 1$ utilizando a representação de inteiros por dois registradores?

2. E para a operação inteira $A = 0$, qual o programa iterativo correspondente?

Representação de Números: Racionais

- Um **número racional** r pode ser representado como um par ordenado de naturais (a, b), tal que $b > 0$ e $r = a/b$

- Note que esta **representação não é unívoca**, pois um mesmo valor pode ter mais de um par que o represente

- Por exemplo, o número 0.75 pode ser representado, entre outros, por $(3, 4)$ ou $(6, 8)$

- Portanto, esta representação cria **classes de equivalência** de pares ordenados naturais

Representação de Números: Racionais

- Neste contexto, as operações de adição, subtração, multiplicação e divisão, bem como o teste de igualdade, podem ser definidos como segue:

  $(a, b) + (c, d) =$

  $(a, b) - (c, d) =$

  $(a, b) \times (c, d) =$

  $(a, b)/(c, d) =$

  $(a, b) = (c, d)$

Representação de Números: Racionais

- Neste contexto, as operações de adição, subtração, multiplicação e divisão, bem como o teste de igualdade, podem ser definidos como segue:

  $(a, b) + (c, d) = (a \bullet d + b \bullet c, b \bullet d)$

  $(a, b) - (c, d) =$

  $(a, b) \times (c, d) =$

  $(a, b)/(c, d) =$

  $(a, b) = (c, d)$

Representação de Números: Racionais

- Neste contexto, as operações de adição, subtração, multiplicação e divisão, bem como o teste de igualdade, podem ser definidos como segue:

  $(a, b) + (c, d) = (a \bullet d + b \bullet c, b \bullet d)$

  $(a, b) - (c, d) = (a \bullet d - b \bullet c, b \bullet d)$

  $(a, b) \times (c, d) =$

  $(a, b)/(c, d) =$

  $(a, b) = (c, d)$

Representação de Números: Racionais

- Neste contexto, as operações de adição, subtração, multiplicação e divisão, bem como o teste de igualdade, podem ser definidos como segue:

  $(a, b) + (c, d) = (a \bullet d + b \bullet c, b \bullet d)$

  $(a, b) - (c, d) = (a \bullet d - b \bullet c, b \bullet d)$

  $(a, b) \times (c, d) = (a \bullet c, b \bullet d)$

  $(a, b)/(c, d) =$

  $(a, b) = (c, d)$

Representação de Números: Racionais

- Neste contexto, as operações de adição, subtração, multiplicação e divisão, bem como o teste de igualdade, podem ser definidos como segue:

  $(a, b) + (c, d) = (a \bullet d + b \bullet c, b \bullet d)$

  $(a, b) - (c, d) = (a \bullet d - b \bullet c, b \bullet d)$

  $(a, b) \times (c, d) = (a \bullet c, b \bullet d)$

  $(a, b)/(c, d) = (a \bullet d, b \bullet c)$ para $c \neq 0$

  $(a, b) = (c, d)$

Representação de Números: Racionais

- Neste contexto, as operações de adição, subtração, multiplicação e divisão, bem como o teste de igualdade, podem ser definidos como segue:

  $(a, b) + (c, d) = (a \bullet d + b \bullet c, b \bullet d)$

  $(a, b) - (c, d) = (a \bullet d - b \bullet c, b \bullet d)$

  $(a, b) \times (c, d) = (a \bullet c, b \bullet d)$

  $(a, b)/(c, d) = (a \bullet d, b \bullet c)$ para $c \neq 0$

  $(a, b) = (c, d)$ sss $a \bullet d = b \bullet c$

Representação de Cadeias de Caracteres

- A Máquina Norma também não possui um tratamento predefinido para o tipo de dado **cadeia de caracteres** (string)

- O **tratamento da definição e da manipulação** de cadeias de caracteres será realizado através de uma **outra Máquina Universal**, denominada Máquina de Turing, a qual prova-se, é **equivalente à Norma**

Representação de Dados Estruturados

- Representação de arranjos unidimensionais: Uma estrutura do tipo **arranjo unidimensional** da forma $A(1), A(2), ...$ pode ser definida por **um único registrador** A usando a **codificação de n-uplas naturais**

- Note que **não é necessário haver um tamanho máximo predefinido**

- Em arranjos, podemos acessar cada posição através de **índices**

- A **indexação** de posições pode ser direta (número natural) ou indireta (conteúdo de um registrador)

Representação de Dados Estruturados (cont.)

- Em ambos os casos, precisamos **definir as operações de incremento, decremento e teste de valor zero** a fim de manter a coerência com a definição de Máquina Norma, tendo as seguintes definições iniciais:
  - Arranjo é implementado usando um registrador A
  - p_n denota o n-ésimo número primo
  - $teste\_div(A, C)$ é um teste que retorna **verdadeiro** se o conteúdo de C é um divisor do conteúdo de A e falso, caso contrário
  - $A := A/C$ denota uma macro de divisão de registradores
  - Por simplicidade, na referência a uma macro definida anteriormente, é omitida a referência aos registradores usados (Ex.: $A := A \times B\ usando\ C, D$ é abreviada por $A := A \times B$

Representação de Dados Estruturados (cont.)

- Indexação direta

  Programa iterativo $ad_A(n)$ *usando* C

$C := p_n;$
$A := A \times C$

  Programa iterativo $sub_A(n)$ *usando* C

$C := p_n;$
(se $teste\_div(A, C)$
então $A := A/C$
senão $\checkmark$)

Representação de Dados Estruturados (cont.)

- Indexação direta (cont.)

  Programa iterativo $zero_A(n)$ $usando$ C

$C := p_n;$
(se $teste\_div(A, C)$
então **falso**
senão **verdadeiro**)

Representação de Dados Estruturados (cont.)

- Indexação indireta

Programa iterativo $ad_A(B)$ $usando$ C

$C := primo(B);$
$A := A \times C$

Programa iterativo $sub_A(B)$ $usando$ C

$C := primo(B);$
`(se` $teste\_div(A, C)$
`então` $A := A/C$
`senão` $\checkmark$`)`

Representação de Dados Estruturados (cont.)

- Indexação indireta (cont.)

Programa iterativo $zero_A(B)\ usando\ C$

$C = primo(B);$
(se $teste\_div(A, C)$
então **falso**
senão **verdadeiro**)

Representação de Dados Estruturados (cont.)

- Arranjo unidimensional $\times$ Norma com 2 registradores
  - Usando-se **indexação direta**, pode-se mostrar que os **registradores** X **e** Y **são suficientes para realizar qualquer processamento**
  - É **suficiente usar-se** X **para armazenar um arranjo unidimensional** onde cada posição corresponde a um registrador
  - Assim, para um determinado registrador K, as operações e teste de Norma ad_K, sub_K e $zero_K$ podem ser simulados pelas operações e teste indexados: $ad_X(k)$ $usando$ Y, $sub_X(k)$ $usando$ Y e $zero_X(k)$ $usando$ Y, onde $X(k)$ denota a k-ésima posição do arranjo em X

Representação de Dados Estruturados (cont.)

- Exemplo: Pilha
  - Base é **fixa** e representa a **primeira posição**
  - Topo é variável e determina a **última posição ocupada**
  - **Operações** são:
    - $empilha$: **Adiciona** conteúdo de um registrador no topo da pilha
    - $desempilha$: **Retira** valor do topo da pilha e o armazena em um registrador
  - Pilha pode ser **simulada usando-se um arranjo e um registrador de índice** (indexação indireta) que indica o topo da pilha

Endereçamento Indireto e Recursão

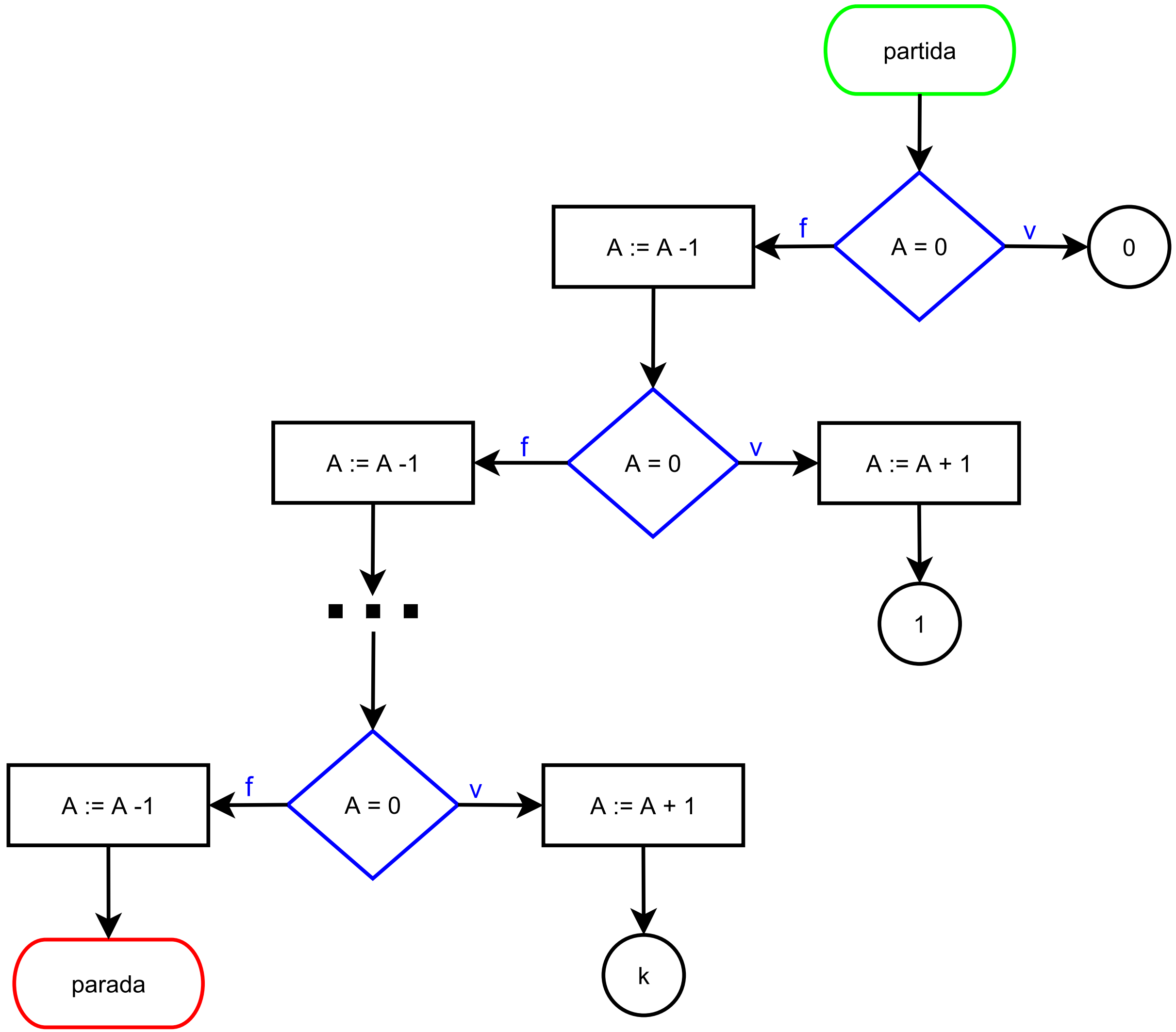
- Considere uma operação com **endereçamento indireto** da seguinte forma, onde A é um registrador:

```
r:   faça F vá_para A
```

- Tal operação pode ser definida pelo **programa monolítico**

```
r:   faça F vá_para End_A
```

onde a macro End_A trata o endereçamento indireto de A e é representada pelo fluxograma a seguir



partida
A = 0
f
v
A := A -1
0
A = 0
f
v
A := A -1
A := A + 1
1
A = 0
f
v
A := A -1
A := A + 1
parada
k

Endereçamento Indireto e Recursão (cont.)

- De forma análoga, é possível definir-se um teste com endereçamento indireto como segue, onde A e B são registradores:

r: `se` T `então vá_para` A `senão vá_para` B

Tarefa

Utilizando o simulador de Máquina Norma disponível no Moodle, crie os programas abaixo (submissão via Moodle até 26/04):

1. Fazer um programa $par(n)$ em Norma que verifica se um número n é par, preservando os valores dos registradores

2. Crie um programa $mult\_int(v1, v2)$ em Norma, o qual realiza a multiplicação dos valores inteiros $v1$ e $v2$

3. Apresente um programa $soma_A(k)$ em Norma que retorne a soma dos k primeiros valores de um arranjo unidimensional A