Matemática Discreta para Ciência da Computação

P. Blauth Menezes

blauth@inf.ufrgs.br

Departamento de Informática Teórica

Instituto de Informática / UFRGS

Matemática Discreta para Ciência da Computação

P. Blauth Menezes

1 Introdução e Conceitos Básicos
2 Lógica e Técnicas de Demonstração
3 Álgebra de Conjuntos
4 Relações
5 Funções Parciais e Totais
6 Endorrelações, Ordenação e Equivalência
7 Cardinalidade de Conjuntos
8 Indução e Recursão
9 Álgebras e Homomorfismos
10 Reticulados e Álgebra Booleana
11 Conclusões

8 – Indução e Recursão

8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
8.8 Recursão
8.9 Leitura Complementar: Funções Recursivas Parciais

8 Indução e Recursão

8.1 Princípio da Indução Matemática

◆ Princípio da Indução Matemática

- técnica para lidar com tipos de dados
- que têm uma relação de boa-ordem

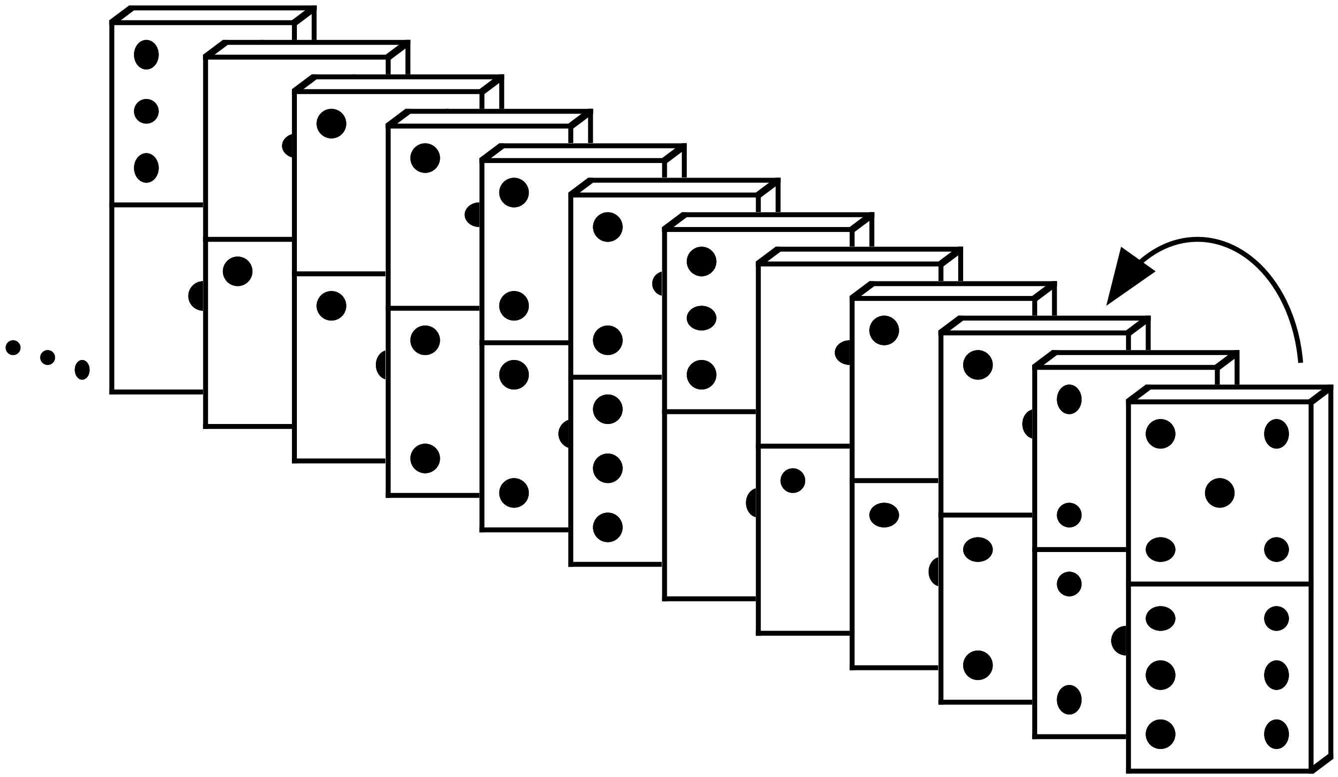
◆ Relação de Boa-Ordem

- todo conjunto não-vazio de elementos do tipo de dado
- tem um elemento mínimo segundo esta relação de ordem
- exemplo: **N**

- **Dada uma boa ordem**
  - pode-se aplicar indução
  - para provar propriedades que valem para todo elemento
- **Por simplicidade**
  - tipo de dados considerado: $\mathbf{N}$
  - ou qq outro conjunto isomorfo a $\mathbf{N}$
- **Exemplo simples que ilustra o Princípio da Indução Matemática**
  - *efeito dominó*

◆ Efeito Dominó

* ao derrrubar a primeira peça
* todas as demais peças serão derrubadas em cadeia



Para estar certo que ocorrerá

- primeira peça é derrubada (direção das demais) (1)
- se qq peça está suficientemente próxima da seguinte, então, ao ser derrubada, fará com que a seguinte também seja derrubada (2)

Então

- por (1), a primeira peça é derrubada
- por (2), a segunda peça é derrubada
- por (2), a terceira peça é derrubada
- e assim sucessivamente…

Portanto, para i tão grande quanto se queira

- i-ésima peça é derrubada

Logo, para qq n

- n-ésima peça é derrubada

◆ Princípio da Indução Matemática

- pode ser resumido como

se o início é correto e se coisa alguma pode dar errada,
então sempre será correto

Def: Princípio da Indução Matemática

$p(n)$ uma proposição sobre $M = \{n \in \mathbf{N} \mid n \geq m \text{ e } m \in \mathbf{N}\}$

Princípio da Indução Matemática

- $p(m)$ é verdadeira
- para qq $k \in M$, $p(k) \Rightarrow p(k+1)$
- então, para qq $n \in \mathbf{N}$, $p(n)$ é verdadeira

Base de indução

- $p(m)$

Hipótese de indução

- $p(k)$

Passo de indução

- $p(k) \Rightarrow p(k+1)$

◆ Na definição

- forma mais tradicional
- denominada de Primeiro Princípio da Indução Matemática
- formulações alternativas: adiante

◆ Duas aplicações da Indução Matemática destacam-se

- Prova Indutiva ou Prova por Indução
  - ⁎ técnica de demonstração comum na CC & Informática
  - ⁎ *não* é do domínio da lógica pura
- Definição Indutiva ou Definição Recursiva
  - ⁎ comum na CC & Informática
  - ⁎ já usada anteriormente de forma informal

◆ Recursão

- conceito próximo ao de indução e presente na grande maioria das linguagens de programação é o de recursão.

◆ Exemplos de construções baseadas em indução e recursão e de especial interesse para CC e Informática

- Computações de um autômato finito
- Gramáticas de Chomsky e a Forma de Backus Naur (ou BNF)
- Expressões Regulares
- Funções Recursivas Parciais ou Funções Recursivas de Kleene

8 – Indução e Recursão

8.1 Princípio da Indução Matemática

8.2 Prova Indutiva

8.3 Segundo Princípio da Indução Matemática

8.4 Definição Indutiva

8.5 Expressões Regulares

8.6 Computações de um Autômato Finito

8.7 Leitura Complementar: Gramática e BNF

8.8 Recursão

8.9 Leitura Complementar: Funções Recursivas Parciais

8.2 Prova Indutiva

◆ Prova Indutiva ou Prova por Indução

- técnica de demonstração
  - baseada no Princípio da Indução Matemática
  - *não* é do domínio da lógica pura
- limita-se a confirmar se $p(n)$ é correta

◆ Demonstração por indução

- demonstrar a base de indução $p(m)$
- fixado um k
  - supor verdadeira a hipótese de indução $p(k)$
  - demonstrar o passo de indução $p(k) \Rightarrow p(k+1)$

Exp: Prova por Indução - p(n): $n < 2^n$

para qq $n \in \mathbb{N}$, *tem-se que* $n < 2^n$

Base de Indução. $k = 0$

$$0 < 1 = 2^0$$

Hipótese de Indução. Suponha que, para $k \in \mathbb{N}$

$$p(k): k < 2^k \text{ é verdadeira}$$

Passo de Indução. Prova para $p(k+1): k+1 < 2^{k+1}$

- $k + 1 <$ hipótese de indução
- $2^k + 1 \leq 2^k + 2^k = 2 * 2^k = 2^{k+1}$

Logo, para qq $n \in \mathbb{N}$, tem-se que $n < n^2$

Exp: Prova por Indução - $p(n)$: $2^n < n!$

para qualquer $n \in \mathbf{N}$, se $n > 3$, então $2^n < n!$

Base de Indução. $k = 4$

$$4^2 = 16 < 24 = 4!$$

Hipótese de Indução. Suponha para $k > 3$

$p(k)$: $2^k < k!$ é verdadeira

Passo de Indução. Prova para $p(k+1)$: $2^{k+1} < (k+1)!$

- $2^{k+1} = 2 * 2^k <$ hipótese de indução
- $2 * k! < (k+1) * k! = (k+1)!$

Logo, para qq $n \in \mathbf{N}$, se $n > 3$, então $2^n < n!$

Exp: Prova por Indução - $1 + 2 + \ldots + n = (n^2 + n)/2$

para qualquer $n \in \mathbf{N}$, tem-se que $1 + 2 + \ldots + n = (n^2 + n)/2$

Base de Indução. $k = 0$

$$(0^2 + 0)/2 = (0 + 0)/2 = 0/2 = 0$$

Hipótese de Indução. Suponha para $k \in \mathbf{N}$

$$p(k)\colon 1 + 2 + \ldots + k = (k^2 + k)/2 \text{ é verdadeira}$$

Passo de Indução. $p(k+1)\colon 1+2+\ldots+k+(k+1) = ((k+1)^2+k+1)/2$

- $1 + 2 + \ldots + k + (k + 1) =$
- $(1 + 2 + \ldots + k) + (k + 1) =$ pela hipótese de indução
- $(k^2 + k)/2 + (k + 1) =$
- $(k^2 + k)/2 + (2k + 2)/2 =$
- $(k^2 + k + 2k + 2)/2 =$
- $((k^2 + 2k + 1) + (k + 1))/2 =$
- $((k + 1)^2 + (k + 1))/2$

Exp: Prova por Indução - $\#2^A = 2^{\#A}$

Foi afirmado que

- se o cardinal de um conjunto A é n
- então o cardinal de 2^A é 2^n, o que justifica a notação 2^A.

Teorema (versão limitada a conjuntos finitos)

para qq conjunto finito A, se $\#A = n$, então tem-se que $\#2^A = 2^n$

Considere o seguinte exemplo motivacional:

- $\mathbf{P}(\varnothing) = \{ \varnothing \}$
- $\mathbf{P}(\{ a \}) = \{ \varnothing, \{ a \} \}$
- $\mathbf{P}(\{ a, b \}) = \{ \varnothing, \{ a \}, \{ b \}, \{ a, b \} \}$
- $\mathbf{P}(\{ a, b, c \}) =$
  $\{ \varnothing, \{ a \}, \{ b \}, \{ c \}, \{ a, b \}, \{ a, c \}, \{ b, c \}, \{ a, b, c \} \}$

Observe $\mathbf{P}(\{ a, b \})$ e $\mathbf{P}(\{ a, b, c \})$ (por exemplo)

- metade dos elementos de $\mathbf{P}(\{ a, b, c \})$ corresponde a $\mathbf{P}(\{ a, b \})$

$$\mathbf{P}(\{ a, b, c \}) = \mathbf{P}(\{ a, b \}) \cup \{ \{ c \}, \{ a, c \}, \{ b, c \}, \{ a, b, c \} \}$$

- outra metade de $\mathbf{P}(\{ a, b, c \})$ corresponde a $\mathbf{P}(\{ a, b \})$, adicionando-se c a cada conjunto

$$\mathbf{P}(\{ a, b, c \}) = \mathbf{P}(\{ a, b \}) \cup$$
$$\{ \varnothing \cup \{ c \}, \{ a \} \cup \{ c \}, \{ b \} \cup \{ c \}, \{ a, b \} \cup \{ c \} \}$$

Ou seja, $\#\mathbf{P}(\{ a, b, c \})$ possui o dobro de elementos de $\#\mathbf{P}(\{ a, b \})$

cada elemento adicionado a um conjunto duplica o cardinal do correspondente conjunto das partes

Base de Indução. Seja $k = 0$ ($\#A = 0$). Então $A = \varnothing$

$$\mathbf{P}(\varnothing) = \{ \varnothing \}$$

- $p(0)$ é verdadeira pois

$$\#\mathbf{P}(\varnothing) = 1 = 2^0 = 2^{\#\varnothing}$$

Hipótese de Indução. Suponha que, para algum $k \in \mathbf{N}$ ($\#A = k$)

$$p(k)\text{: } \#\mathbf{P}(A) = 2^{\#A} \text{ é verdadeira}$$

Passo de Indução. $p(k + 1)$. Sem perda de generalidade, suponha os seguintes conjuntos (conjuntos com o mesmo cardinal são isomorfos)

- $A = \{ 1, 2, 3, \ldots, k \}$
- $B = \{ 1, 2, 3, \ldots, k, k + 1 \}$

Por hipótese de indução $\#\mathbf{P}(A) = 2^k$

$\mathbf{P}(A)$ são todos subconjuntos de B que *não* possuem o elemento $k + 1$

- demais subconjuntos B são aqueles que contém o elemento $k + 1$
- a união de cada conjunto de $\mathbf{P}(A)$ com $\{ k + 1 \}$
- resultando em outros 2^k conjuntos

$$\#\mathbf{P}(B) = 2^k + 2^k = 2^{k+1}$$

Logo, para qq conjunto finito A, se $\#A = n$, então tem-se que $\#2^A = 2^n$

8 – Indução e Recursão

8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
8.8 Recursão
8.9 Leitura Complementar: Funções Recursivas Parciais

8.3 Segundo Princípio da Indução Matemática

- **Freqüentemente é conveniente trabalhar com outras formulações do Princípio da Indução Matemática**
  - Primeiro Princípio da Indução Matemática
    - ⁎ princípio introduzido
  - Segundo Princípio da Indução Matemática
    - ⁎ formulação especialmente importante
    - ⁎ considera não apenas o resultado anterior $p(k)$ mas todos os anteriores para concluir $p(k+1)$

◆ Lembrando...

Def: (Primeiro) Princípio da Indução Matemática

$p(n)$ uma proposição sobre $M = \{ n \in \mathbf{N} \mid n \geq m \text{ e } m \in \mathbf{N} \}$

Princípio da Indução Matemática

- $p(m)$ é verdadeira
- para qq $k \in M$, $p(k) \Rightarrow p(k+1)$
- então, para qq $n \in \mathbf{N}$, $p(n)$ é verdadeira

Def: Segundo Princípio da Indução Matemática

Seja $p(n)$ proposição sobre $M = \{n \in \mathbf{N} \mid n \geq m \text{ e } m \in \mathbf{N}\}$.

Primeira Versão.

- $p(m)$ é verdadeira
- para qq $k \in M$: $p(m) \land p(m+1) \land \ldots \land p(k) \Rightarrow p(k+1)$
- então, para qq $n \in \mathbf{N}$, $p(n)$ é verdadeira

Segunda Versão. Suponha $t \in \mathbf{N}$

- $p(m), p(m+1), \ldots, p(m+t)$, são verdadeiras;
- para qq $k \in M$ tq $k \geq m+t$: $p(m) \land p(m+1) \land \ldots \land p(k) \Rightarrow p(k+1)$
- então, para qq $n \in \mathbf{N}$, $p(n)$ é verdadeira

Segunda versão: prova os t primeiros casos em separado para verificar a base de indução

◆ Aplicação usual do Segundo Princípio

- definição e prova de propriedades de
  - ∗ expressões
  - ∗ fórmulas
  - ∗ árvores, etc.

- razão pela qual freqüentemente é denominado de
  - ∗ indução em estrutura
  - ∗ indução estruturada

Exp: Segundo Princípio: Proposição Lógica

Suponha que p é uma proposição lógica a qual contém exclusivamente os conetivos lógicos $\land$, $\lor$ e $\rightarrow$. Se o valor verdade de todos os átomos de p é V, então o valor verdade de p é V

Uma prova por indução (no número de átomos, primeira versão):

Base de Indução. Seja $k = 1$

- p é um átomo
- portanto, por hipótese, p é V

Hipótese de Indução. Suponha que, para algum $k \in \mathbf{N}$, e para qq $u \in \mathbf{N}$ tal que $u \leq k$

- se o número de átomos de p é u, então o valor verdade de p é V

Passo de Indução. Seja p uma proposição com $k + 1$ átomos

- p pode ser reescrita (q e r possuem individualmente no máximo k átomos e, conjuntamente, $k + 1$ átomos):
  - $q \wedge r$
  - $q \vee r$
  - $q \rightarrow r$
- por hipótese de indução q e r são V
  - portanto, em qualquer dos casos, p é V

Exp: Segundo Princípio: Postagem

Qualquer valor de postagem igual ou maior que 12 reais pode ser formado usando exclusivamente selos de 4 e de 5 reais

Uma prova por indução (no número de selos, segunda versão):

Base de Indução. Seja $k \in \{ 12, 13, 14, 15 \}$

- 12 reais: 3 selos de 4
- 13 reais: 2 selos de 4 e 1 selo de 5
- 14 reais: 1 selo de 4 e 2 selos de 5
- 15 reais: 3 selos de 5

Hipótese de Indução. Suponha que, para algum $k \in \mathbf{N}$, e para qq $u \in \mathbf{N}$ tal que $15 \leq u \leq k$

- se o valor é u, então pode ser formado usando selos de 4 e 5

Passo de Indução. Seja uma postagem cujo valor é $k + 1$ reais

- tal postagem pode ser formada usando
  - uma postagem de $k - 3$ reais
  - mais um selo de 4 reais

8 – Indução e Recursão

8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
8.8 Recursão
8.9 Leitura Complementar: Funções Recursivas Parciais

8.4 Definição Indutiva

- **Princípio da indução Matemática**
  - pode ser usado em definições
- **Definição Indutiva ou Definição Recursiva**
  - Base de indução
    - explicita os casos elementares (os mais simples)
  - Passo de indução/recursão
    - demais casos são definidos em termos dos anteriores
- **Definição Indutiva ja foi usado informalmente**
  - Fecho transitivo

Exp: Definição Indutiva – Fecho Transitivo

Endorrelação $R: A \to A$. Fecho Transitivo

- Se $\langle a, b\rangle \in R$, então $\langle a, b\rangle \in R^+$ (1)
- Se $\langle a, b\rangle \in R^+$ e $\langle b, c\rangle \in R^+$, então $\langle a, c\rangle \in R^+$ (2)
- Os únicos elementos de R^+ são os construídos como acima (3)

(1) base de indução

(2) passo de indução (hipótese?)

(3) garante que, de fato, é definição indutiva

- (3) pode ser omitido (subentendido)

Exp: Def. Indutiva – Conjunto de Todas as Palavras

Alfabeto $\Sigma = \{ a, b \}$

$$\Sigma^* = \{ \varepsilon, a, b, aa, ab, ba, bb, aaa,\ldots \}$$

Base de Indução.

- $\varepsilon \in \Sigma^*$
- qq $x \in \Sigma$, tem-se que $x \in \Sigma^*$

Passo de Indução.

- se u e v são palavras de Σ^*,
- então a concatenação $u\,v$ é palavra de Σ^*

Exp: Definição Indutiva – Fórmula Lógica

Simplificada, *não* incluindo quantificadores, variáveis, etc.

Base de Indução

- qq proposição atômica (incluindos V e F) é fórmula

Passo de Indução. Se q e r são fórmulas então também são fórmulas

- $(\neg q)$
- $(q \wedge r)$
- $(q \vee r)$
- $(q \rightarrow r)$
- $(q \leftrightarrow r)$

Exp: Definição Indutiva × Princípio da Indução Mat

Ilustração da relação entre o Princípio da Indução e a definição indutiva

- definição indutiva (no número de conetivos) de fórmula lógica usando o Segundo Princípio

Base de Indução. Seja $k = 0$

- qq proposição atômica (incluindo V e F) é uma fórmula;

Hipótese de Indução. Suponha que, para algum $k \in \mathbf{N}$, e para qq $i \in \mathbf{N}$ tal que $i \leq k$

- se p é uma fórmula com i conetivos, então p é uma fórmula lógica

Passo de Indução. Seja p uma fórmula com $k + 1$ conetivos

- p pode ser reescrita
- q e r são fórmulas e possuem conjuntamente k conetivos
  - $(\neg q)$
  - $(q \wedge r)$
  - $(q \vee r)$
  - $(q \rightarrow r)$
  - $(q \leftrightarrow r)$

8 – Indução e Recursão

8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
8.8 Recursão
8.9 Leitura Complementar: Funções Recursivas Parciais

8.5 Expressões Regulares

◆ Autômato Finito

- já introduzido
- é especialmente importante
  - ∗ define a classe das linguagens regulares
  - ∗ possui diversas aplicações na Computação e Informática

◆ Expressões Regulares

- alternativa para definir linguagens regulares
- definição indutiva que segue
  - ∗ Segundo Princípio da Indução Matemática (segunda versão)

Def: Expressão Regular (ER) sobre um alfabeto Σ

Base de Indução.

- $\varnothing$ é ER e denota a linguagem vazia
- ε é ER e denota a linguagem $\{ \varepsilon \}$
- qq símbolo $x \in \Sigma$ é ER e denota a linguagem $\{ x \}$

Passo de Indução. Se r e s são ER e denotam as ling. R e S, então

- *União*. $(r + s)$ é ER e denota a linguagem $R \cup S$

- *Concatenação*. (rs) é ER e denota a linguagem

$$R S = \{ uv \mid u \in R \text{ e } v \in S \}$$

- *Concatenação Sucessiva*. (r^*) é ER e denota a linguagem R^*

Exp: Expressão Regular

Omissão de parênteses em uma ER é usual

- concatenação sucessiva: precedência sobre concatenação e união
- concatenação: precedência sobre união

ER	Linguagem Representada
aa	somente a palavra aa
ba*	todas palavras iniciam por b seguido por 0 ou mais a
$(a + b)^*$	todas as palavras sobre { a, b }
$(a + b)^*aa(a + b)^*$	todas as palavras contendo aa como subpalavra
$a^*ba^*ba^*$	todas as palavras contendo exatamente dois b
$(a + b)^*(aa + bb)$	todas as palavras que terminam com aa ou bb
$(a + \varepsilon)(b + ba)^*$	todas as palavras sem dois a consecutivos

8 – Indução e Recursão

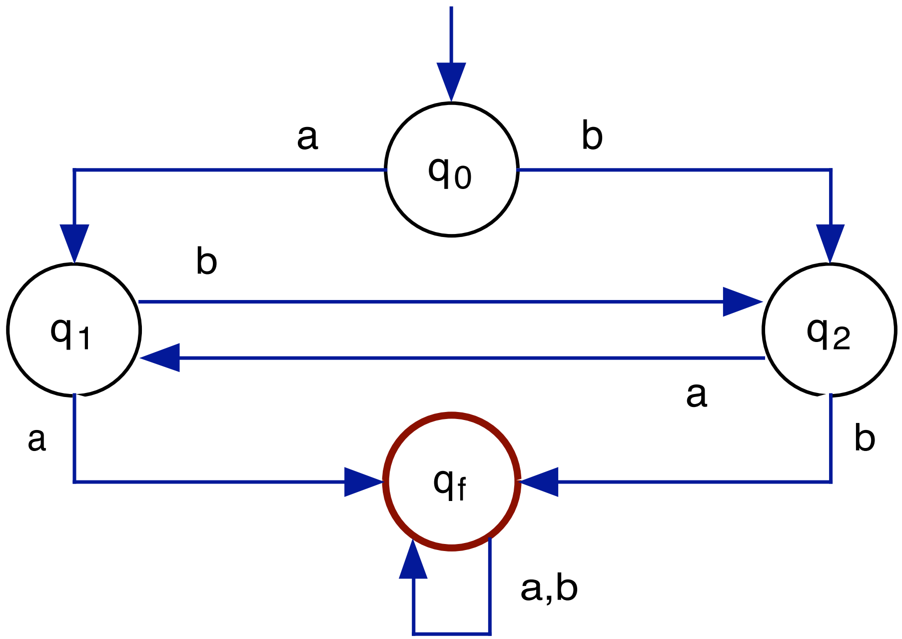
8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
8.8 Recursão
8.9 Leitura Complementar: Funções Recursivas Parciais

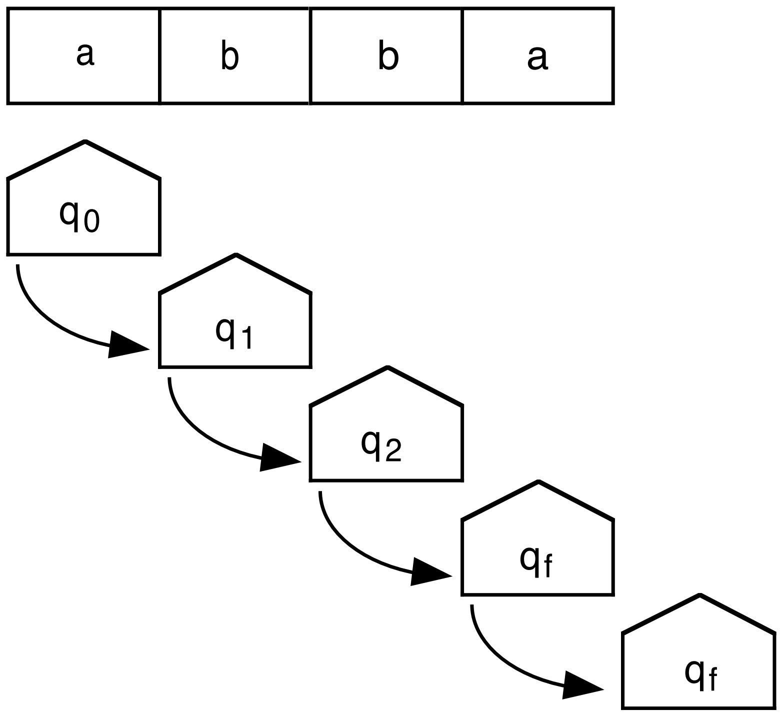
8.6 Computações de um Autômato Finito

- **Exemplos de modelos computacionais introduzidos**
  - Rede de Petri, definida como uma relação
  - Autômato Finito, definido como uma função parcial
- **Definidos sintaticamente**
  - definições formais de sua *forma*
- **Semântica ??**
  - interpretação do funcionamento ou processamento
  - introduzida textualmente, de forma informal

◆ Autômato Finito

processamento de um Autômato Finito é a sucessiva aplicação de computações atômicas (transições)





◆ Programa de um Autômato Finito: função parcial

$$\delta: Q \times \Sigma \to Q$$

- $\delta(\langle q, a \rangle) = p$
  - ∗ no estado q, ao ler o símbolo a, assume o estado p

◆ Para dar semântica à sintaxe de um Autômato Finito

- *estender* a definição da função programa
  - ∗ argumento: um estado e uma *palavra*
- definida indutivamente

◆ Função Programa Estendida

- *base de indução*: entrada vazia (palavra vazia)
  - permanece no mesmo estado

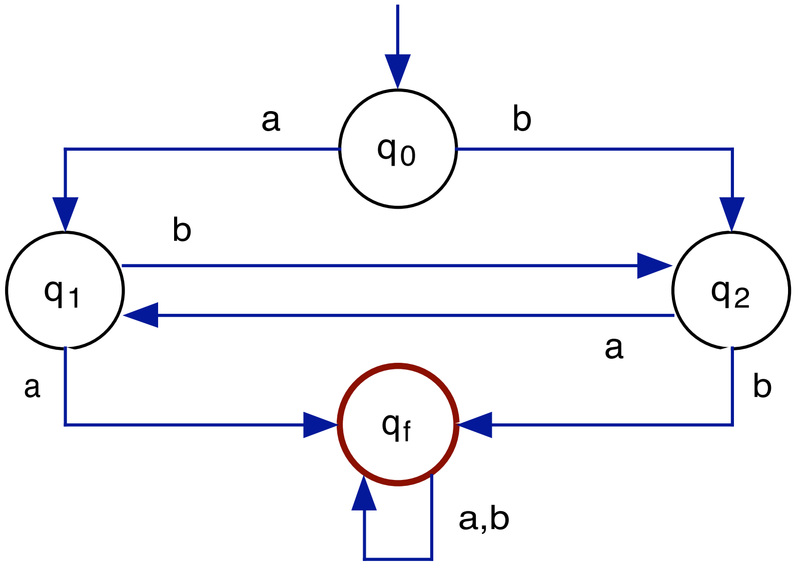
- *passo de indução*: entrada *não*-vazia (palavra não-vazia)
  - processa o primeiro símbolo da entrada, usando δ
  - mesmo raciocínio sucessivamente ao resto da palavra

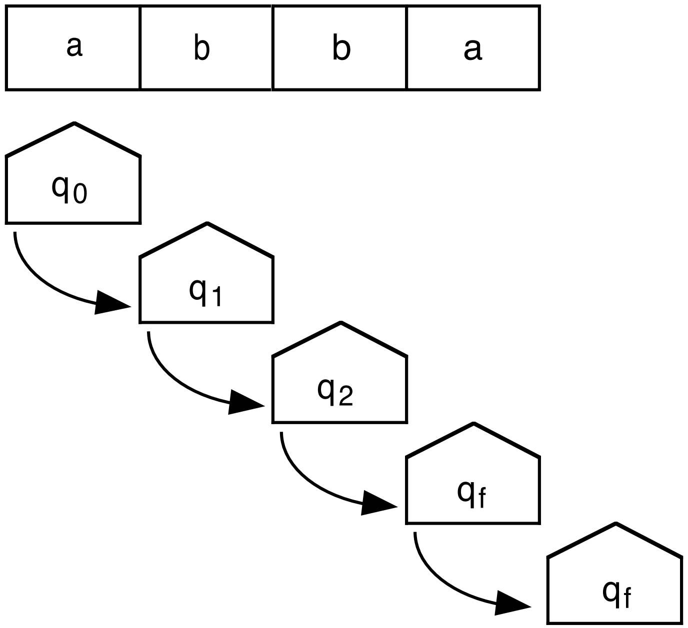
◆ Função Programa Estendida – definição formal

$$\underline{\delta}: Q \times \Sigma^* \to Q$$

- $\underline{\delta}(q, \varepsilon) = q$
- $\underline{\delta}(q, aw) = \underline{\delta}(\delta(q, a), w)$

Exp: Função Programa Estendida





- $\underline{\delta}(q_0, abba) = \underline{\delta}(\delta(q_0, a), bba) =$
- $\underline{\delta}(q_1, bba) = \underline{\delta}(\delta(q_1, b), ba) =$
- $\underline{\delta}(q_2, ba) = \underline{\delta}(\delta(q_2, b), a) =$
- $\underline{\delta}(q_f, a) = \underline{\delta}(\delta(q_f, a), \varepsilon) =$
- $\underline{\delta}(q_f, \varepsilon) = q_f$

8 – Indução e Recursão

8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
8.8 Recursão
8.9 Leitura Complementar: Funções Recursivas Parciais

8.7 Gramática e BNF

◆ Gramáticas de Chomsky

- formalismo universalmente aceito e usado em Computação e Informática e outras áreas

- Linguagens Formais
  - definição e estudo das propriedades das linguagens
- Teoria da Computação
  - limites da solucionabilidade de problemas e propriedades
- Linguagens Naturais
  - linguagens como o português
- Sistemas Biológicos
  - simulação do desenvolvimento de sistemas vivos

- **Prova-se: gramática é equivalente a Máq. de Turing**
  - poder computacional
- **Hipótese de Church**
  - qq função computável pode ser especificada por uma gramática
  - gramática: aceita como definição formal de algoritmo
- **Gramáticas em Computação e Informática**
  - especialmente importantes para definir a sintaxe de linguagens
    - Pascal, C, …
    - definidas usando gramáticas

8 – Indução e Recursão

8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
  8.7.1 Gramática
  8.7.2 BNF
8.8 Recursão
8.9 Leitura Compl.: Funções Recursivas Parciais

8.7.1 Gramática

- **Gramática é, basicamente**
  - conjunto *finito* de regras
  - aplicadas *sucessivamente*, geram palavras
- **Geração de palavra**
  - indutivamente definida
- **Linguagem**
  - conjunto de todas as palavras geradas
  - forma *finita* de expressar sintaxe de linguagens eventualmente *infinitas*

◆ Gramáticas para linguagens naturais ???

- mesmas

◆ Definição de Semântica ???

- *podem* ser usadas gramáticas
- em geral, são usados outros tipos de formalismos

Def: Gramática

Gramática de Chomsky ou simplesmente Gramática

- V - conjunto *finito* de símbolos variáveis ou não-terminais
- T - conjunto *finito* de símbolos terminais disjunto de V
- P: $(V \cup T)^+ \rightarrow (V \cup T)^*$
  - relação *finita* — P é finito
  - regra de produção: par da relação
- S - elemento *distinguido* de V
  - símbolo inicial ou variável inicial

◆ Notação

- regra de produção $\langle \alpha, \beta \rangle$

$$\alpha \to \beta$$

- grupo de regras da forma $\alpha \to \beta_1$, $\alpha \to \beta_2$, ..., $\alpha \to \beta_n$

$$\alpha \to \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$$

◆ Regras de produção

- definem as condições de geração das palavras da linguagem

◆ Derivação

- aplicação de uma regra de produção
- par de uma relação

Def: Relação de Derivação

G gramática tq P: $(V \cup T)^+ \to (V \cup T)^*$ e S é o símbolo inicial

Derivação é um par da Relação de Derivação

- $\Rightarrow$: $(V \cup T)^+ \to (V \cup T)^*$
- $\langle \alpha, \beta \rangle$ da relação $\Rightarrow$ é denotado de forma infixada

$$\alpha \Rightarrow \beta$$

Relação $\Rightarrow$ é *indutivamente definida*

- qq regra $S \to \beta$ de P, então $S \Rightarrow \beta$ é derivação

- qq derivação $\eta \Rightarrow \rho\alpha\sigma$
  - ∗ se $\alpha \to \beta$ é regra de P
  - ∗ então $\eta \Rightarrow \rho\beta\sigma$

Def: Linguagem Gerada (por uma Gramática)

G gramática tq S é símbolo inicial

Linguagem Gerada pela gramática G

$$\text{Linguagem}(G) = \{ w \in T^* \mid S \Rightarrow^+ w \}$$

- $\Rightarrow^+$ denota o fecho transitivo da relação de derivação $\Rightarrow$
- portanto: palavra de uma linguagem *não* pode conter variáveis

Exp: Gramática e Derivação - Números Naturais

Gramática G

- $V = \{ N, D \}$, sendo que N é o símbolo inicial
- $T = \{ 0, 1, 2, \ldots, 9 \}$
- P contém as regras
  - $N \rightarrow D$
  - $N \rightarrow DN$
  - $D \rightarrow 0 \mid 1 \mid \ldots \mid 9$

Derivação de 243 (existe mais alguma geração?)

$$N \Rightarrow DN \Rightarrow 2N \Rightarrow 2DN \Rightarrow 24N \Rightarrow 24D \Rightarrow 243$$

Distingue zeros à esquerda: $123 \neq 0123$

◆ Interpretação indutiva

- *Base de Indução*
  - ⁎ todo dígito é um natural
- *Passo de Indução*
  - ⁎ Se n é um natural
  - ⁎ então a concatenação de n com qq dígito também é natural

Exp: Gramática e Derivação - Expressão Aritmética

Gramática G

- $V = \{ E \}$, sendo que E é o símbolo inicial
- $T = \{ +, *, (,), x \}$
- P contém as regras (interpretação indutiva?)
  - $E \to E+E$
  - $E \to E*E$
  - $E \to (E)$
  - $E \to x$

$(x + x)*x \in \text{Linguagem}(G)$???

$$E \Rightarrow E * E \Rightarrow (E) * E \Rightarrow (E + E) * E \Rightarrow$$

$$(x + E) * E \Rightarrow (x + x) * E \Rightarrow (x + x) * x$$

8 – Indução e Recursão

8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
  8.7.1 Gramática
  8.7.2 BNF
8.8 Recursão
8.9 Leitura Compl.: Funções Recursivas Parciais

8.7.2 BNF

Forma de Backus Naur ou simplesmente BNF

- (do inglês, Backus Naur Form)

◆ BNF

- variáveis são delimitas por $\langle$ e por $\rangle$
- palavras *não*-delimitas: símbolos terminais
- representação de uma regra de produção $\langle\alpha, \beta\rangle$

$$\alpha ::= \beta$$

Exp: BNF - Identificador em Pascal

BNF - ⟨ident⟩ é o símbolo inicial

- ⟨ident⟩ ::= ⟨letra⟩ | ⟨ident⟩⟨letra⟩ | ⟨ident⟩⟨dígito⟩
- ⟨letra⟩ ::= a | b | … | z
- ⟨dígito⟩ ::= 0 | 1 | … | 9

Base de Indução

- toda letra é um identificador

Passo de Indução

- se S é identificador
- então a concatenação de S com qq letra/dígito é identificador

8 – Indução e Recursão

8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
8.8 Recursão
8.9 Leitura Complementar: Funções Recursivas Parciais

8.8 Recursão

- **Recursão**
  - conceito próximo ao de indução
  - presente na grande maioria das linguagens de programação
- **Inspirado nas Funções Recursivas de Kleene**
  - equivalente: Máquina de Turing e Gramática de Chomsky
  - Hipótese de Church
    - * qq algoritmo pode ser expresso usando recursividade

◆ Indução × Recursão

- qq definição indutiva pode ser simulada por recursão
- *nem* toda recursão possui uma correspondente definição indutiva
  - *não* necessariamente respeita a boa ordem da indução
- recomendável programar recursão respeitando os princípios da definição indutiva
  - garante que o programa pára e atinge o resultado esperado

◆ Exemplo importante para o entendimento da recursão

Exp: Definição Indutiva – Fatorial

Fatorial $n!$

- $n! = 1$, se $n = 0$
- $n! = n * (n - 1) * (n - 2) \ldots * 1$, se $n > 0$

Para $n > 0$

$n * (n - 1)!$ sendo que $(n - 1)! = (n - 1) * (n - 2) \ldots * 1$

Da mesma forma, $(n - 1)!$

$(n - 1) * (n - 2)!$, sendo que $(n - 2)! = (n - 2) * (n - 3) \ldots * 1$

e assim sucessivamente…

Base de Indução

- $0! = 1$

Passo de Indução

- $n! = n * (n - 1)!$

Fatorial de 4

- $4! = 4 * (4 - 1)! = 4 * 3! =$ passo de indução
- $4 * 3 * (3 - 1)! = 4 * 3 * 2! =$ passo de indução
- $4 * 3 * 2 * (2 - 1)! = 4 * 3 * 2 * 1! =$ passo de indução
- $4 * 3 * 2 * 1 * (1 - 1)! = 4 * 3 * 2 * 1 * 0! =$ base de indução
- $4 * 3 * 2 * 1 * 1 = 24$

Exp: Função Recursiva em Pascal– Fatorial

```
function fatorial(n: integer): integer;
   begin
   if n = 0
   then fatorial := 1
   else fatorial := n * fatorial(n – 1)
   end
```

Exp: Função recursiva em Haskell– Fatorial

```
fatorial(0) = 1
fatorial(n) = n * fatorial(n – 1)
```

❏

- **Nem toda programação recursiva corresponde a uma definição indutiva**

Exp: Função Recursiva

```
function ciclo_infinito(x: real): real;
   begin
   ciclo_infinito := x * ciclo_infinito(x)
   end
```

- *não* corresponde a noção de boa ordem
- *não* caracteriza definição indutiva

Obs: Eficiência da Recursão

Processamento de recursão em linguagens imperativas: C, Pascal…

- considerável demanda do recurso memória

- recursão
  - expressa certos programas de forma mais elegante
  - mas *não* é muito recomendado

- recursão escrita de forma iterativa (**`while`**, **`for`**…)
  - em geral, *mais eficiente*

Obs: Eficiência da Recursão

Recursão em Linguagens funcionais, como Haskell

- técnica padrão de aplicar uma operação sucessivamente
- base da maioria dos programas funcionais

Otimizações não-usadas em linguagens imperativas

- estudo de tais técnica *não* é detalhado

8 – Indução e Recursão

8.1 Princípio da Indução Matemática
8.2 Prova Indutiva
8.3 Segundo Princípio da Indução Matemática
8.4 Definição Indutiva
8.5 Expressões Regulares
8.6 Computações de um Autômato Finito
8.7 Leitura Complementar: Gramática e BNF
8.8 Recursão
8.9 Leitura Complementar: Funções Recursivas Parciais

Matemática Discreta para Ciência da Computação

P. Blauth Menezes

1 Introdução e Conceitos Básicos
2 Lógica e Técnicas de Demonstração
3 Álgebra de Conjuntos
4 Relações
5 Funções Parciais e Totais
6 Endorrelações, Ordenação e Equivalência
7 Cardinalidade de Conjuntos
8 Indução e Recursão
9 Álgebras e Homomorfismos
10 Reticulados e Álgebra Booleana
11 Conclusões

Matemática Discreta para Ciência da Computação

P. Blauth Menezes

blauth@inf.ufrgs.br

Departamento de Informática Teórica

Instituto de Informática / UFRGS