



Engenharia de Computação

Estrutura de Dados 2

Aula 18 – Árvore B+

Prof. Muriel de Souza Godoi
muriel@utfpr.edu.br

Tipos de Acesso a Arquivos

- Alternativas (até o momento)
 - acesso indexado
 - arquivo pode ser visto como um conjunto de registros que são indexados por uma chave
 - acesso sequencial
 - arquivo pode ser acessado sequencialmente (i.e., registros fisicamente contínuos)
- Objetivo
 - arquivos devem suportar acesso indexado eficiente, e também acesso sequencial

Exemplo

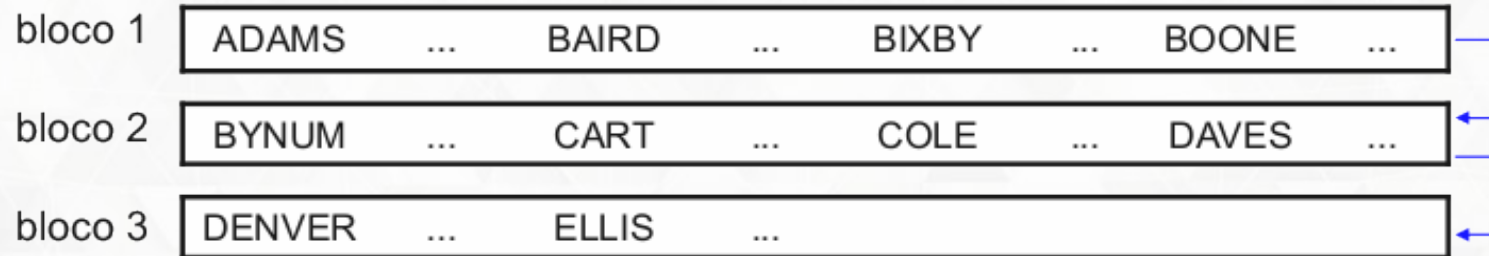
- Arquivo indexado por um índice Árvore B
 - acesso indexado pela chave: desempenho excelente :)
 - acesso sequencial aos registros ordenado pela chave: desempenho péssimo :(
- Arquivo com registros ordenados pela chave
 - processamento sequencial (acessar todos registros): apropriado :) (bufferização)
 - processamento randômico: inapropriado :(- (logarítmico (ordem 2 – busca binária)

Meta

- Organizar um arquivo de modo que seja eficiente tanto para processamento sequencial quanto aleatório
- Solução?
 - Modelo Híbrido!

Modelo Híbrido

- Arquivo de dados é organizado em **blocos de tamanho fixo**, de registros sequenciais, ordenados pelas chaves, e encadeados (*Sequence Set*)
 - Privilegiando o acesso sequencial

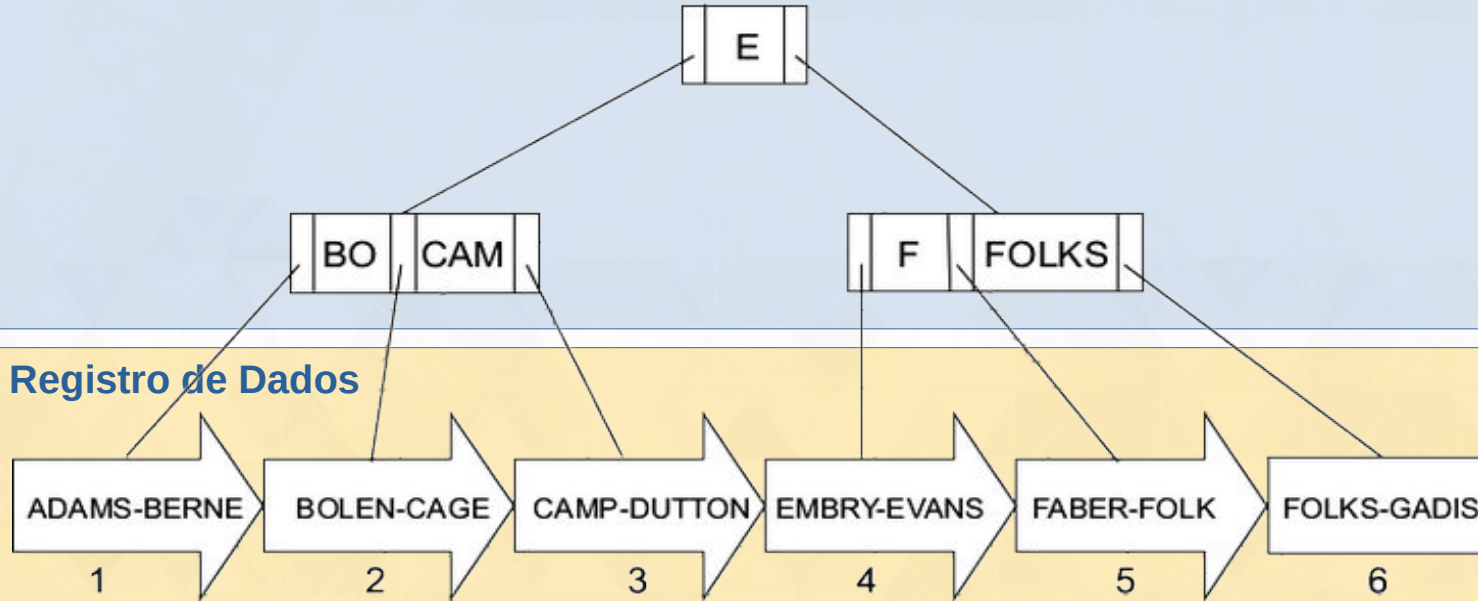


Modelo Híbrido

- Arquivo de índices é organizado como uma Árvore-B, cujas folhas são os blocos de registros sequenciais
 - Privilegiando busca aleatória
- Páginas não folhas
 - contêm chaves ou partes de chaves separadoras para os filhos

Árvore B+ de Prefixo Simples

Árvore B



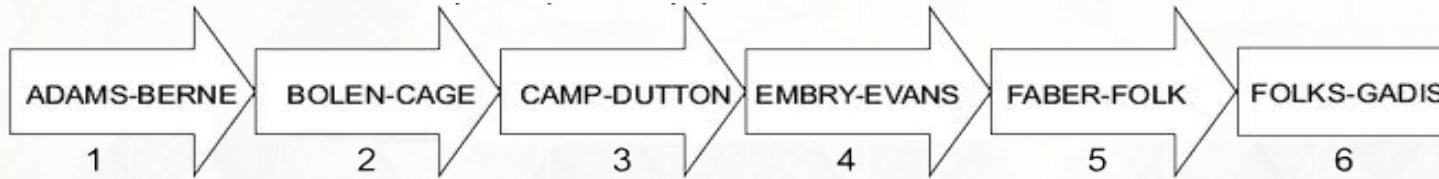
Blocos de Registro de Dados

Árvores B+

- Todas as **chaves** estão apenas nas **páginas folhas**;
- As páginas **não folhas** abrigam **apenas separadores** de chaves
 - prefixos mínimos de tamanhos variáveis;
 - Economia de espaço;
- As páginas folhas consistem de blocos sequenciais e ordenados dos registros de dados
- As páginas folhas **são encadeadas** de modo que possam ser acessadas independentemente da árvore

Índice Simples (Tabela)

- Se todos os índices couberem na RAM, uma tabela poderia substituir a Árvore B:
 - Busca binária (adaptada) para encontrar a chave

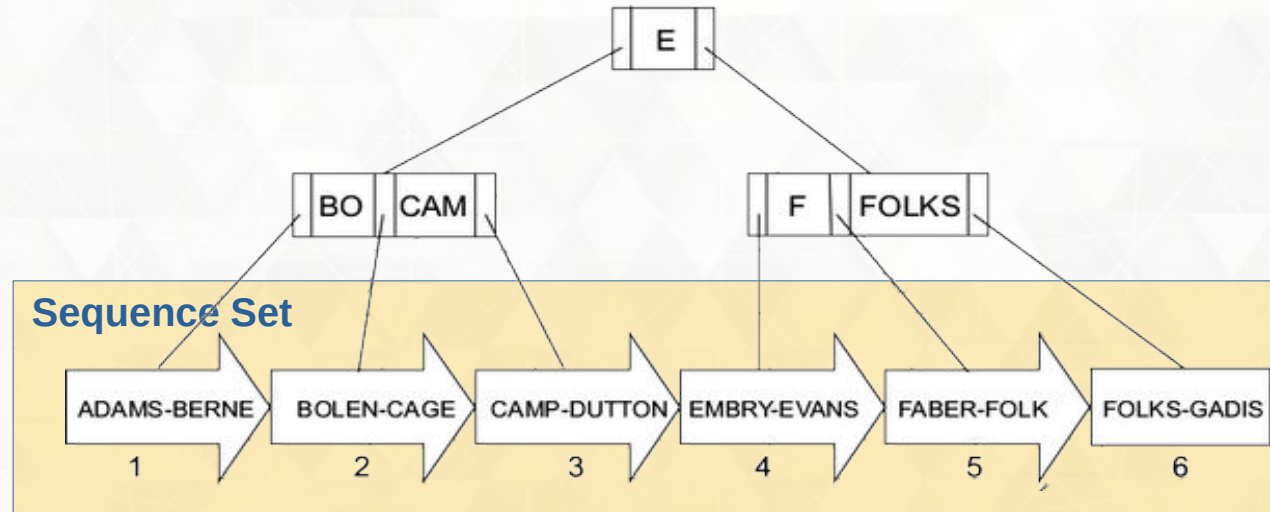


Chave	Bloco
BERNE	1
CAGE	2
DUTTON	3
EVANS	4
FOLK	5
GADIS	6

- **Índice de Nível 1**
 - Registros de tamanho fixo
 - Contém a chave do último registro do bloco

Sequence Set

- Registros ordenados fisicamente pela chave (*sequence set*)
- Organizados como blocos de registros
 - um bloco consiste na unidade básica de entrada e saída e deve ter seu tamanho determinado pelo tamanho do *buffer-pool*



Uso de blocos - Características

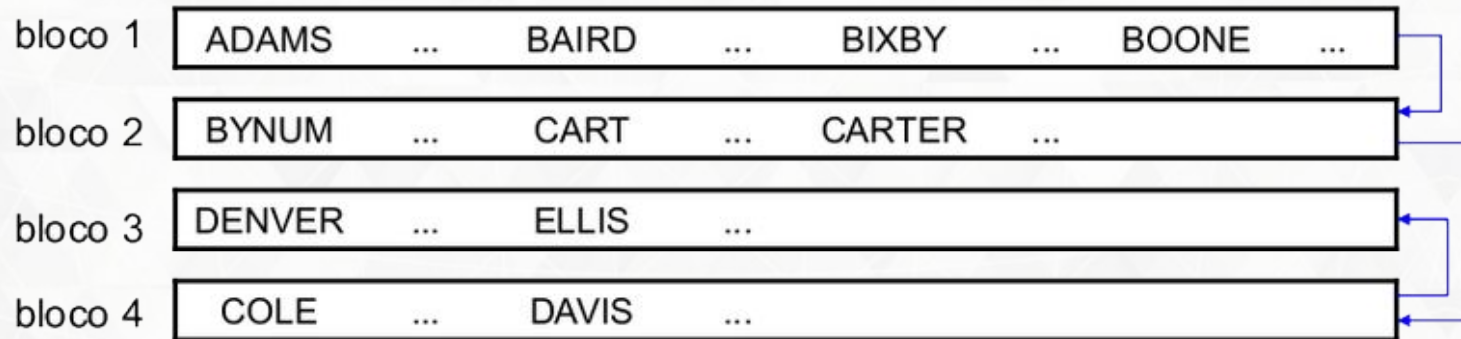
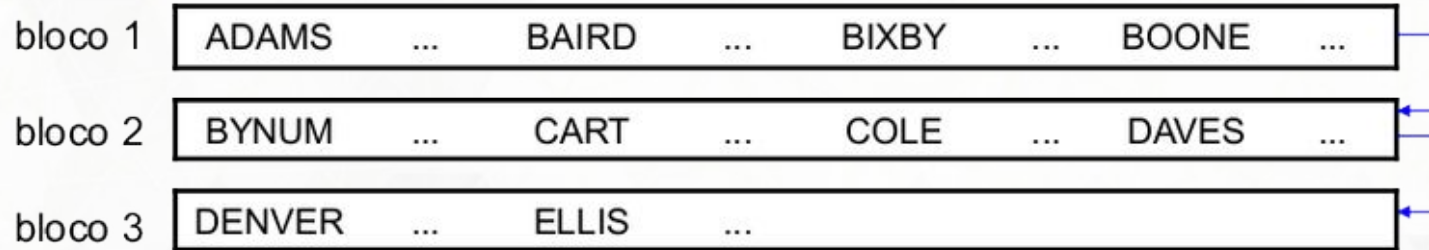
- Conteúdo de cada bloco está ordenado, e pode ser recuperado em um acesso
- cada bloco mantém um 'ponteiro' para o bloco antecessor e um 'ponteiro' para o bloco sucessor
 - Lista duplamente encadeada
- Blocos logicamente adjacentes não estão (necessariamente) fisicamente adjacentes
- **Garante acesso sequencial ao arquivo!**

Inserção com Overflow

- Inserção de registros pode provocar *overflow* em um bloco
- Solução
 - dividir o bloco, em um processo análogo ao realizado em árvores-B (*split*)
- Passos
 - divide os registros entre os dois blocos
 - rearranja os ponteiros
- Não existe promoção!

Inserção com Overflow

- Exemplo: Inserção de **CARTER**



Remoção com Underflow

- Remoção de registros pode provocar *underflow* em um bloco
- Solução
 - **concatenar** o bloco com o seu antecessor ou sucessor na sequência lógica; ou
 - **redistribuir** os registros, movendo-os entre blocos logicamente adjacentes

Remoção com Underflow

- Exemplo: Remoção de **DAVIS**

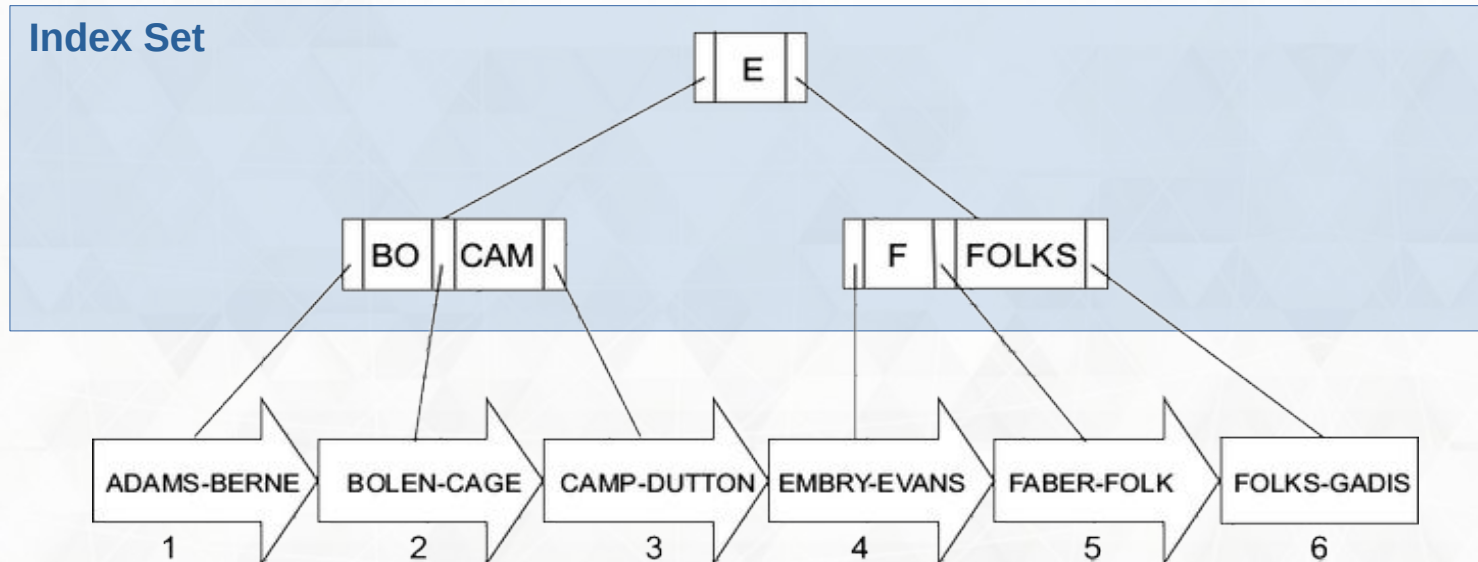


Uso de Blocos – Custos Associados

- Devido à fragmentação gerada pelas inserções, o arquivo pode ocupar mais espaço do que um arquivo ordenado comum
 - melhorias incluem redistribuição antes do particionamento, split 2-to-3, etc.
- A ordem física dos registros não é necessariamente sequencial ao longo do arquivo

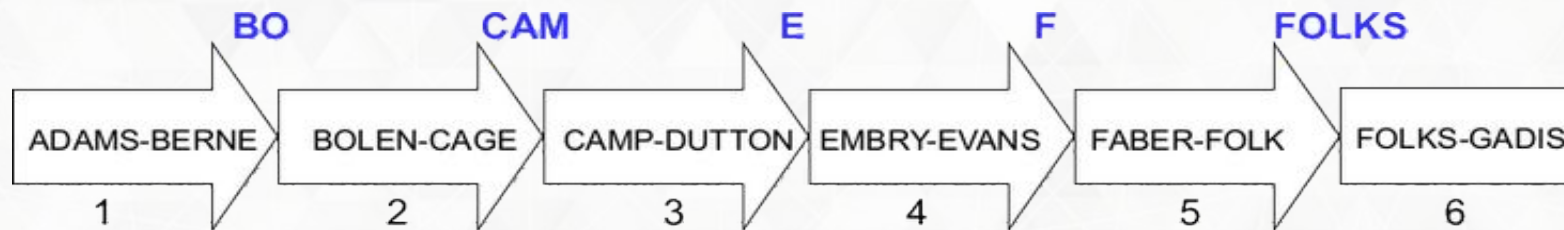
Index Set (Árvore-B)

- Para localizar eficientemente um bloco com um registro particular, dada a chave do registro
- Ao invés de chaves, prefixos de chaves (de tamanho variável) compõem as páginas não folhas das árvore-B



Separadores

- Características
 - Mantidos no índice, ao invés das chaves de busca
 - Possuem tamanho variável
- Desafio
 - Escolher o menor separador para utilizar no índice
 - Tabela de decisão:
 - **Chave < Separador?** Procure a **esquerda**
 - **Chave >= Separador?** Procure à **direita**

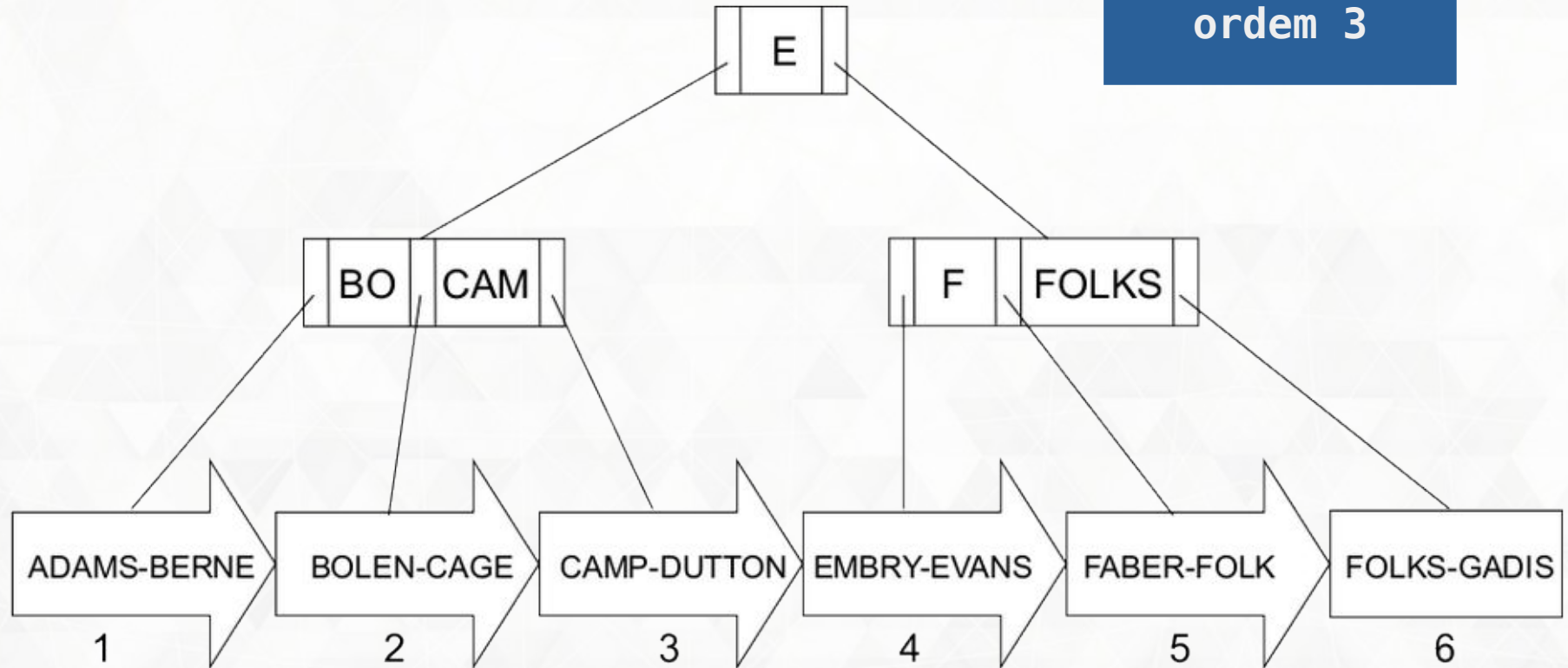


Árvore B+

- Estrutura híbrida
 - chaves
 - organizadas como árvore-B (i.e., index set)
 - nós folhas
 - consistem em blocos de sequence set
- Árvores B+ de prefixos simples
 - armazena na árvore as cadeias separadoras mínimas entre cada par de blocos
 - usar separadores mínimos faz com que os nós possam ser maiores

Árvores B+ de prefixos simples

Árvore B de
ordem 3



Árvore B+ - Manutenção

- Cenários

- inserção
- remoção
- overflow
- underflow



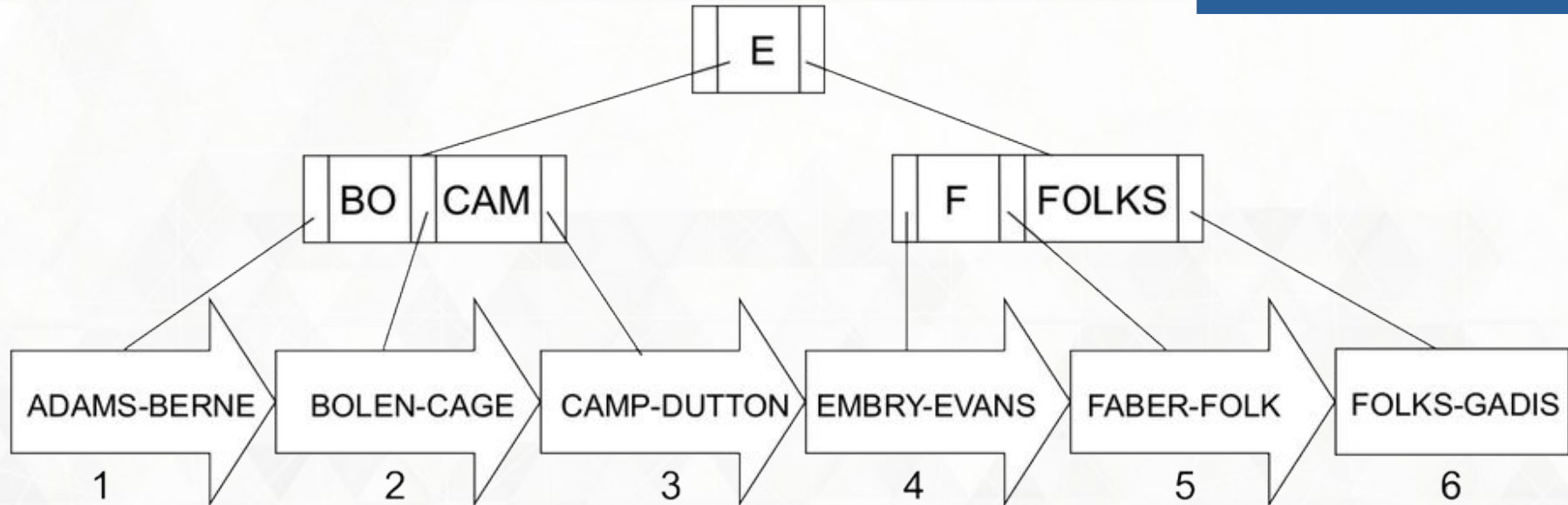
- Efeitos colaterais

- sequence set
- index set

Manipulando a Árvore B+

- Remoção de **EMBRY**

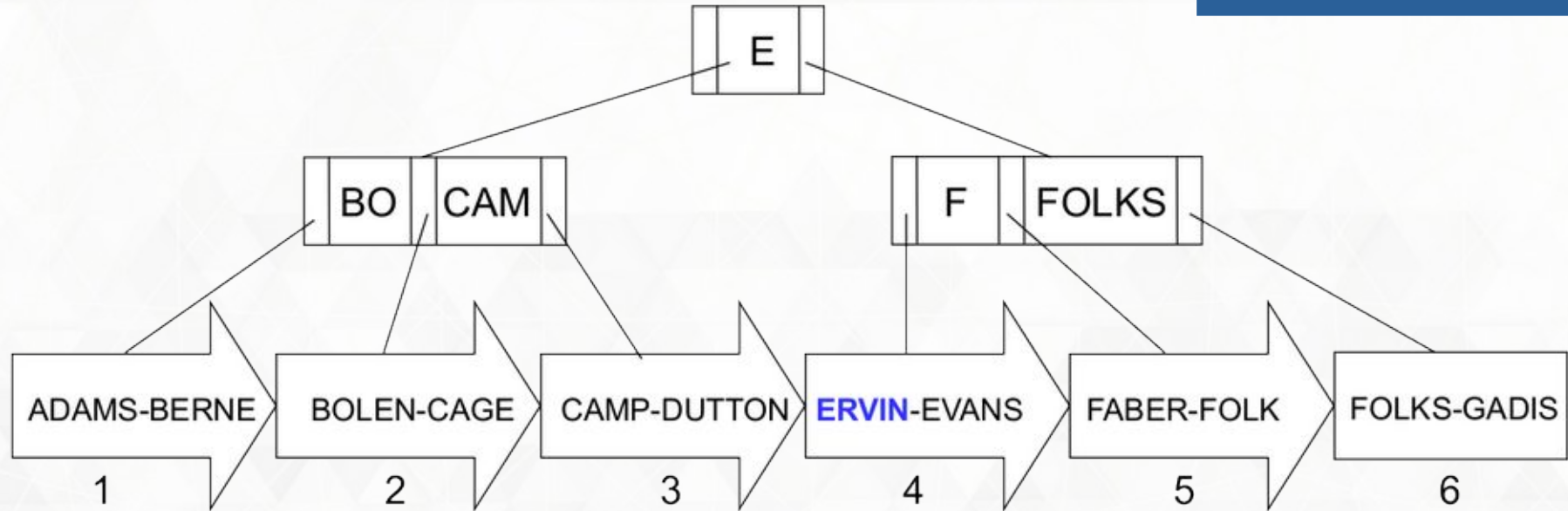
Sem redistribuição
ou concatenação



Manipulando a Árvore B+

- Remoção de **EMBRY**

Sem redistribuição
ou concatenação

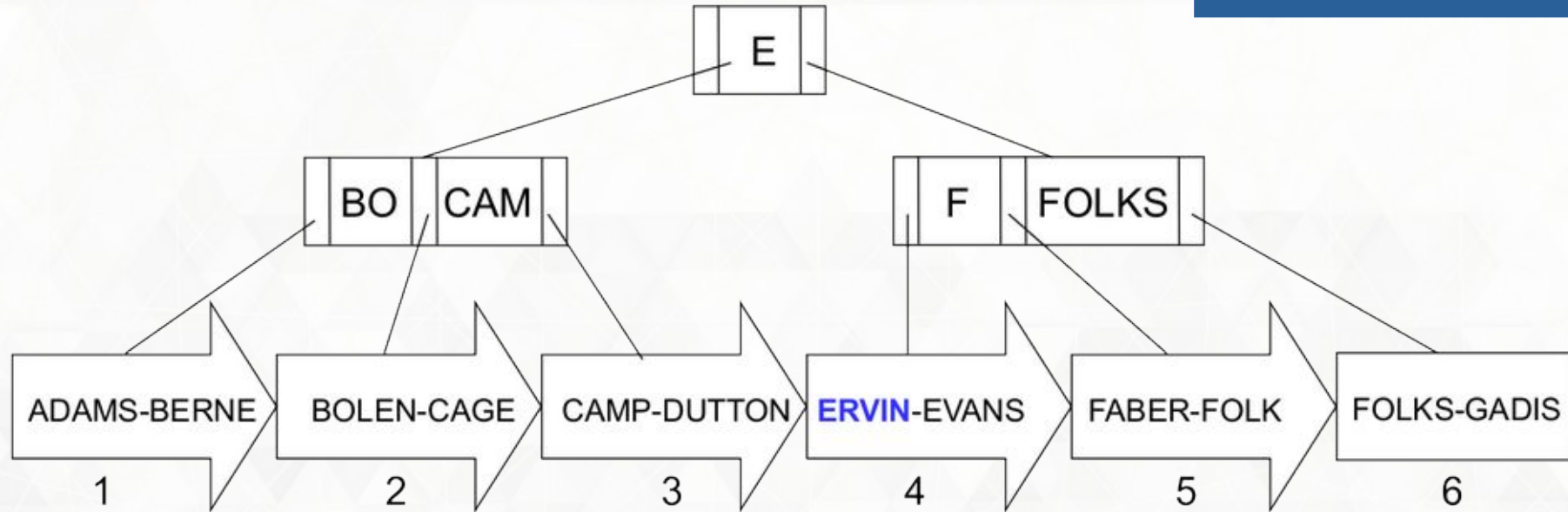


- Efeito no *sequence set*
 - limitado a alterações no bloco 4

Manipulando a Árvore B+

- Remoção de **EMBRY**

Sem redistribuição
ou concatenação



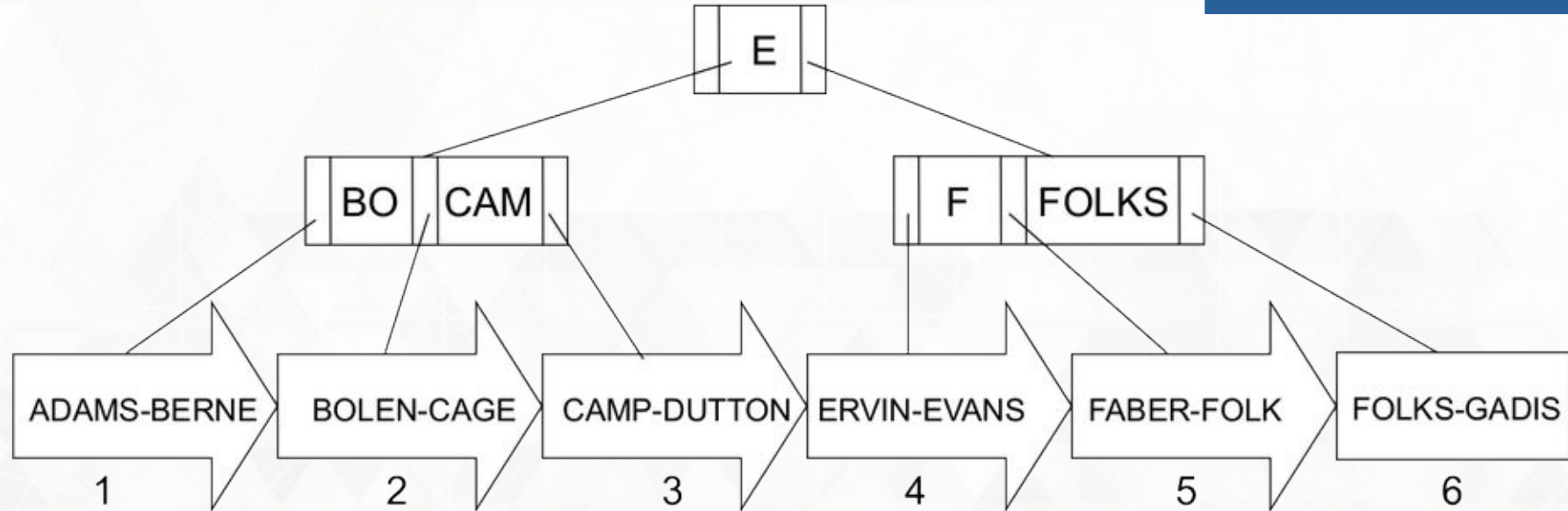
- Efeito na **Árvore B+**

- **nenhum**: 'E' é uma boa chave separadora

Manipulando a Árvore B+

- Remoção de **FOLKS**

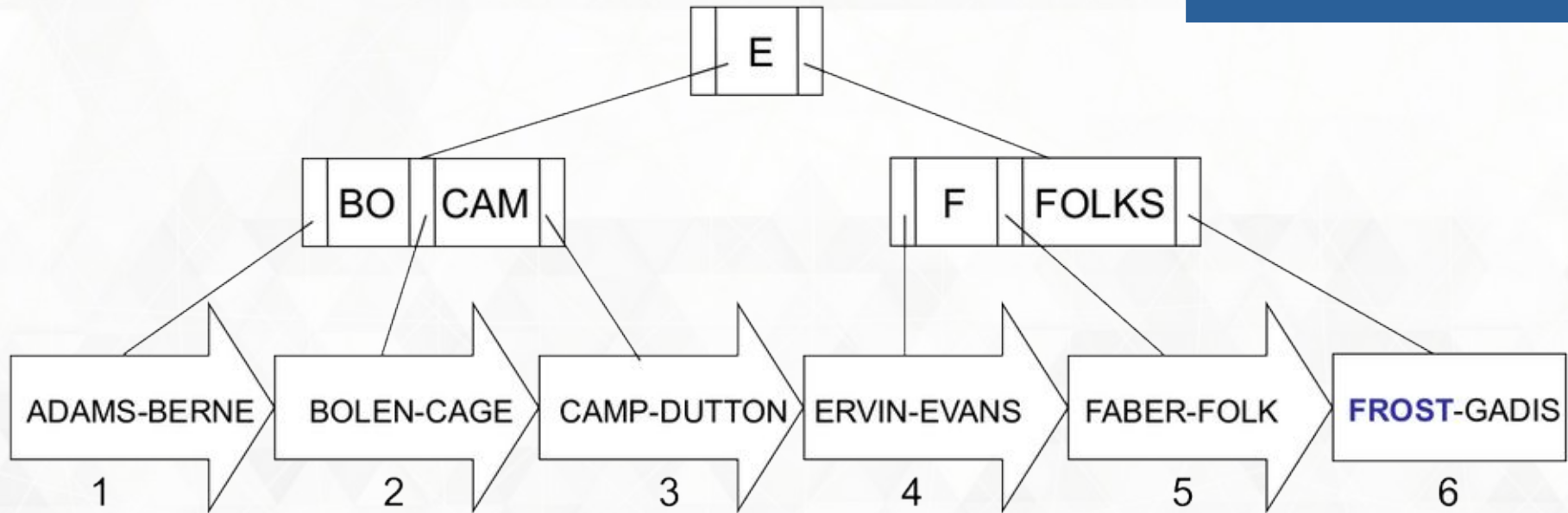
Sem redistribuição
ou concatenação



Manipulando a Árvore B+

- Remoção de **FOLKS**

Sem redistribuição
ou concatenação

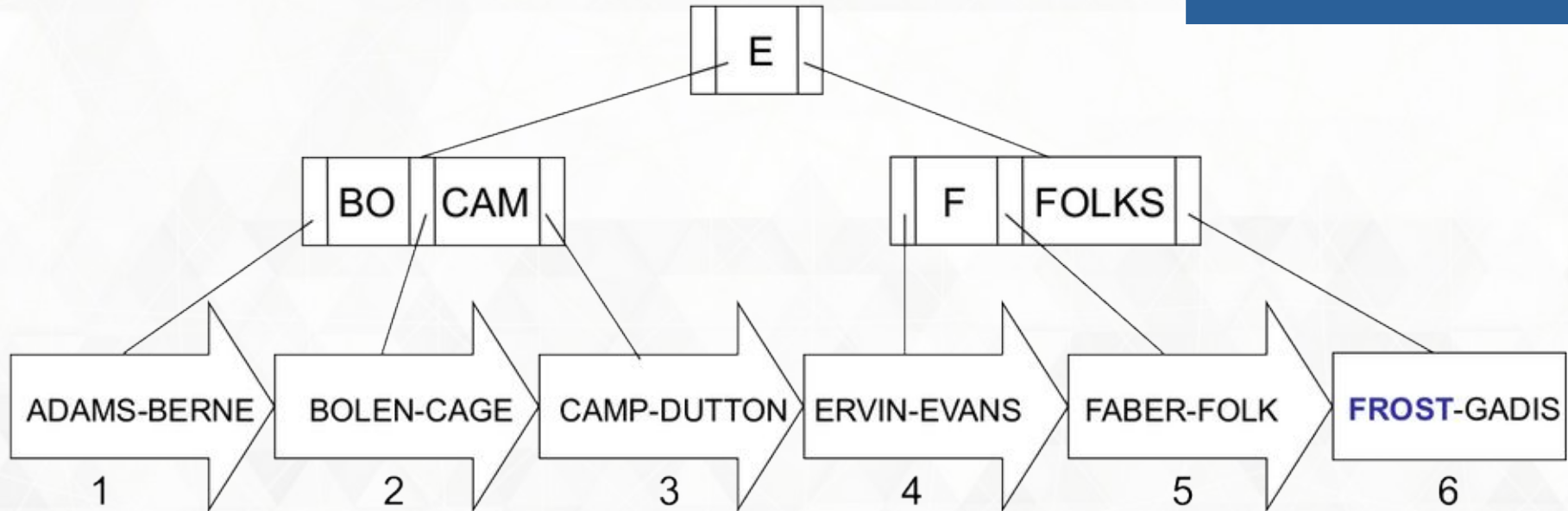


- Efeito no *sequence set*
 - limitado a alterações no bloco 6

Manipulando a Árvore B+

- Remoção de **FOLKS**

Sem redistribuição
ou concatenação



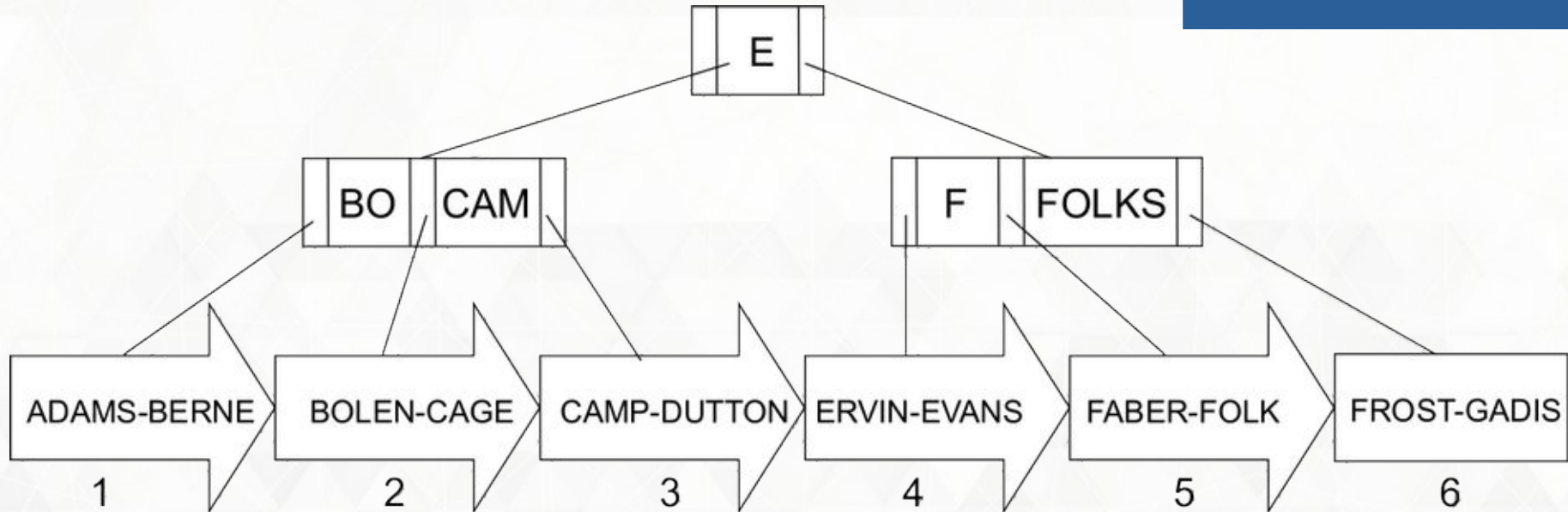
- Efeito na **Árvore B+**

- **nenhum**: custo elevado se fosse arrumar1

Manipulando a Árvore B+

- Inserção de **EATON**

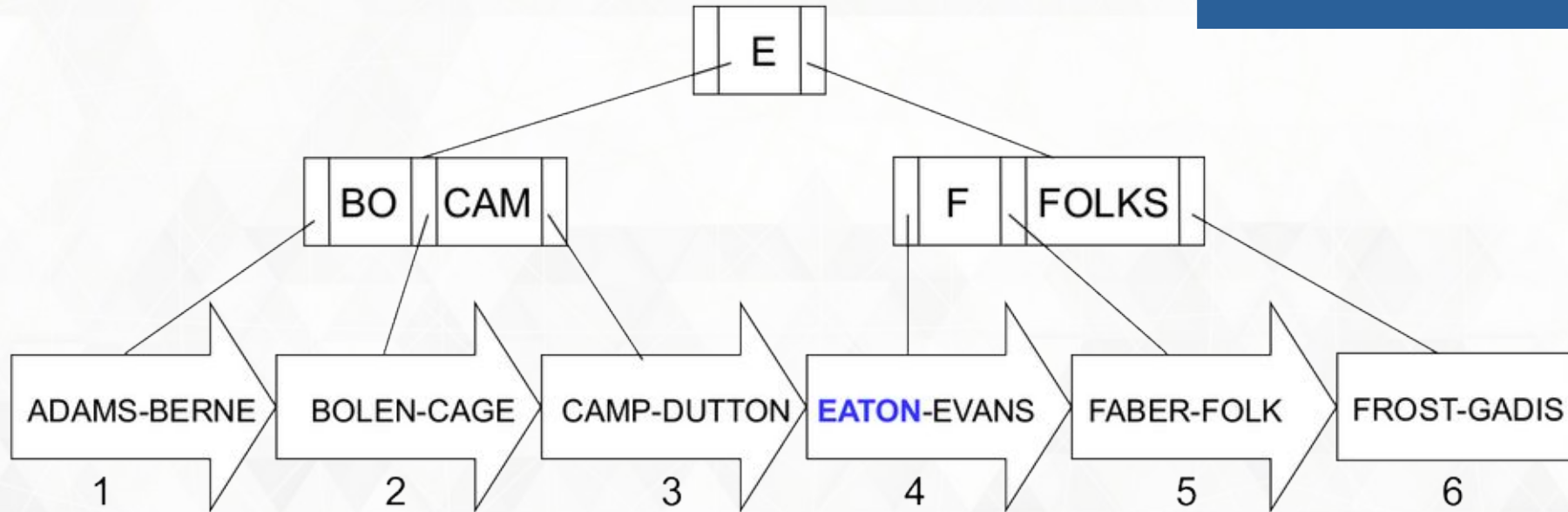
Espaço disponível
no bloco



Manipulando a Árvore B+

- Inserção de **EATON**

Espaço disponível
no bloco

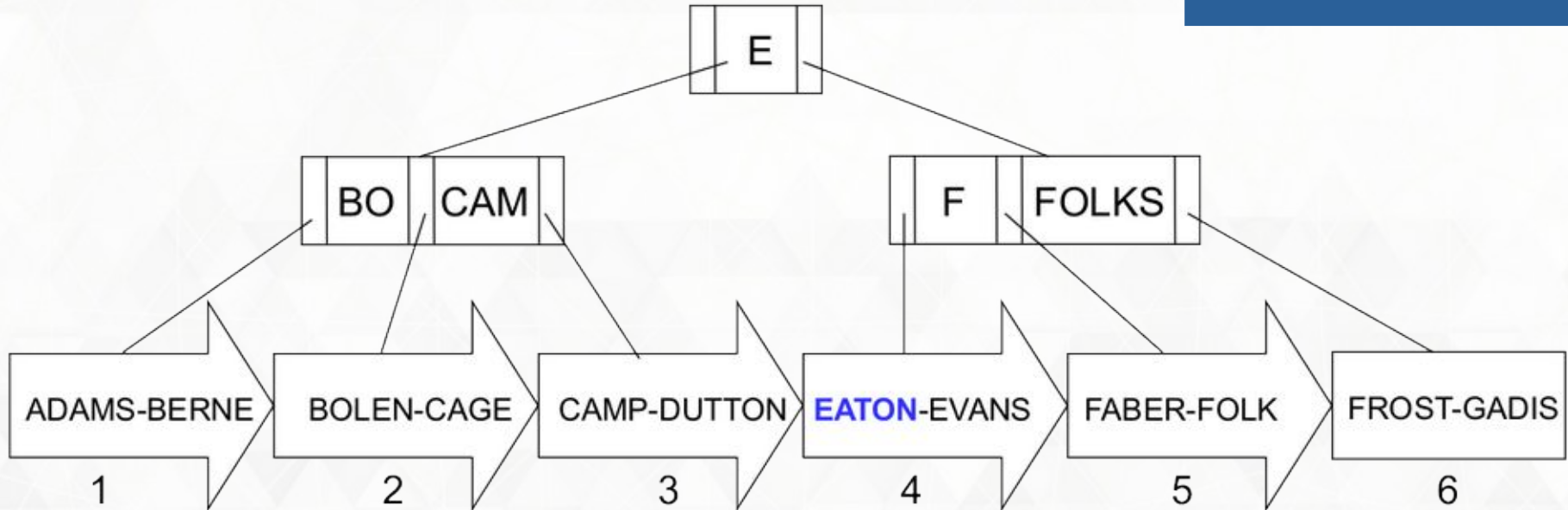


- Efeito no *sequence set*
 - limitado a alterações no bloco 4

Manipulando a Árvore B+

- Inserção de **EATON**

Espaço disponível
no bloco



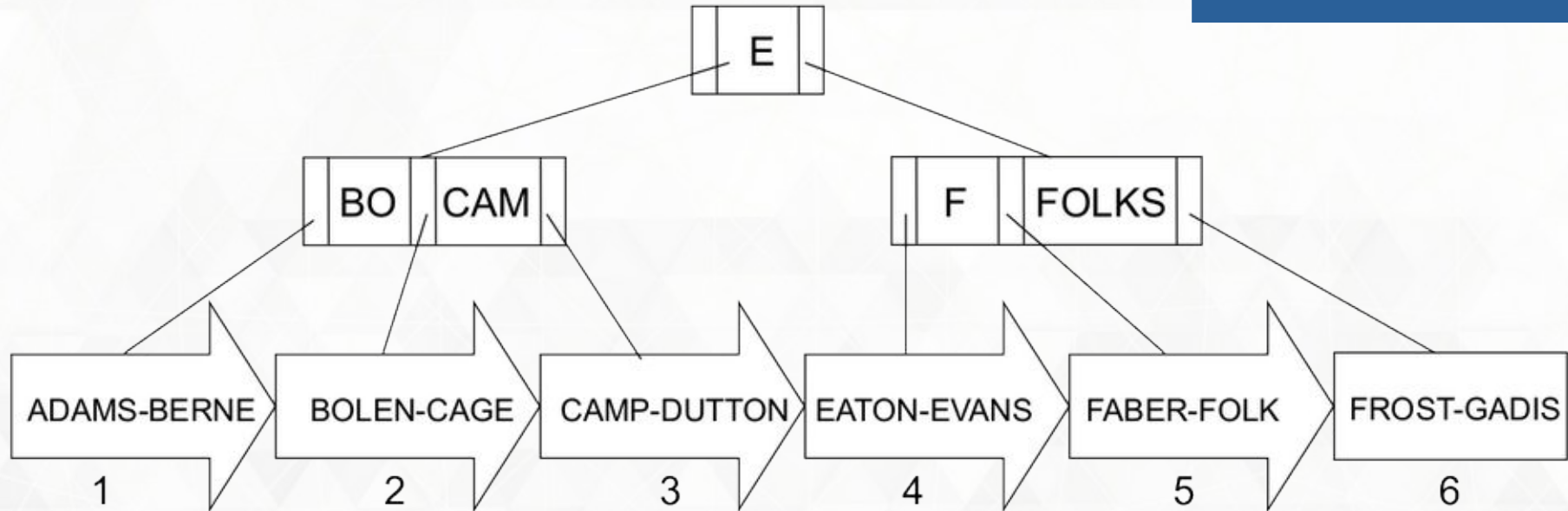
- Efeito na **Árvore B+**

- **nenhum**: 'E' ainda é uma boa chave separadora

Manipulando a Árvore B+

- Inserção de **AVERY**

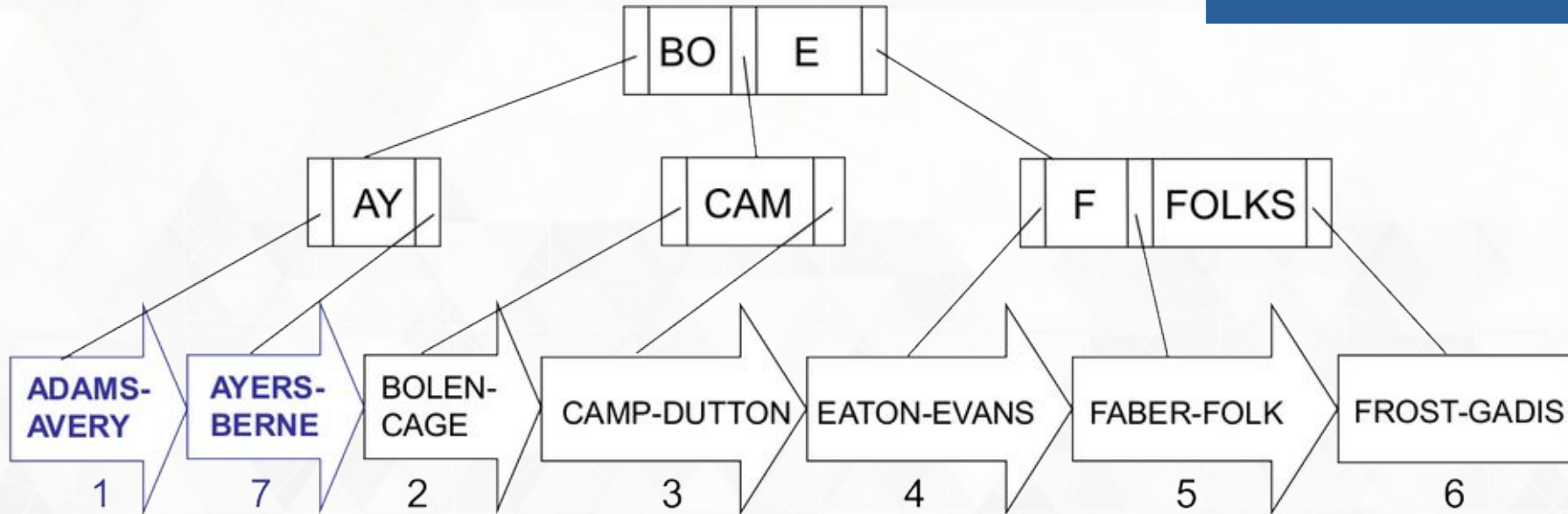
*Split no sequence
set 1*



Manipulando a Árvore B+

- Inserção de **AVERY**

Split no sequence set 1



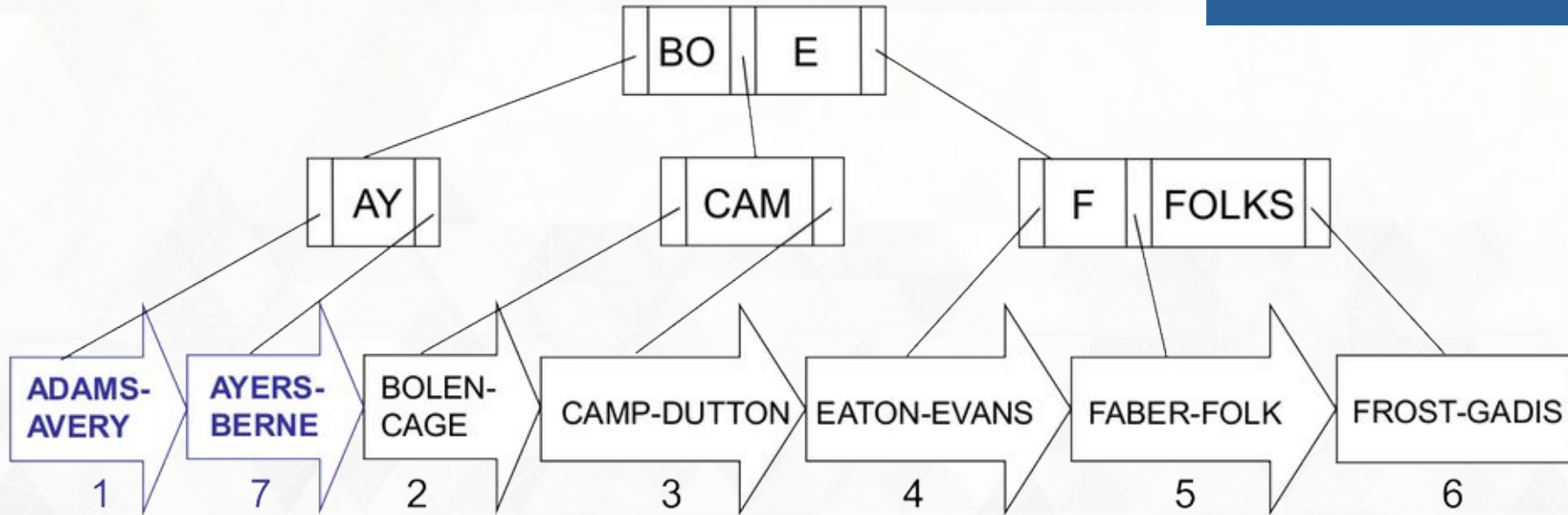
- Efeito no *sequence set*

- dados do bloco 1 + **AVERY** distribuídos entre os blocos 1 e 7

Manipulando a Árvore B+

- Inserção de **AVERY**

*Split no sequence
set 1*



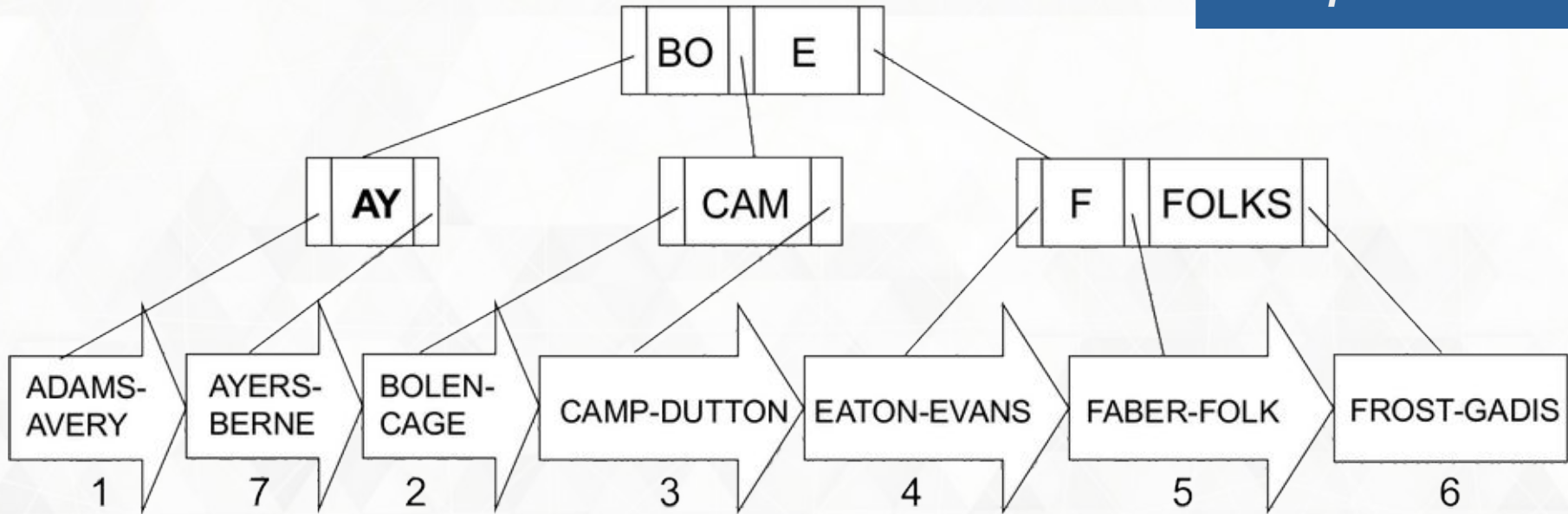
- Efeito na **Árvore B+**

- Criação e promoção de separador adicional **AY**;
- Overflow; splitting e promoção de **BO**

Manipulando a Árvore B+

- Remoção de **CAEL**

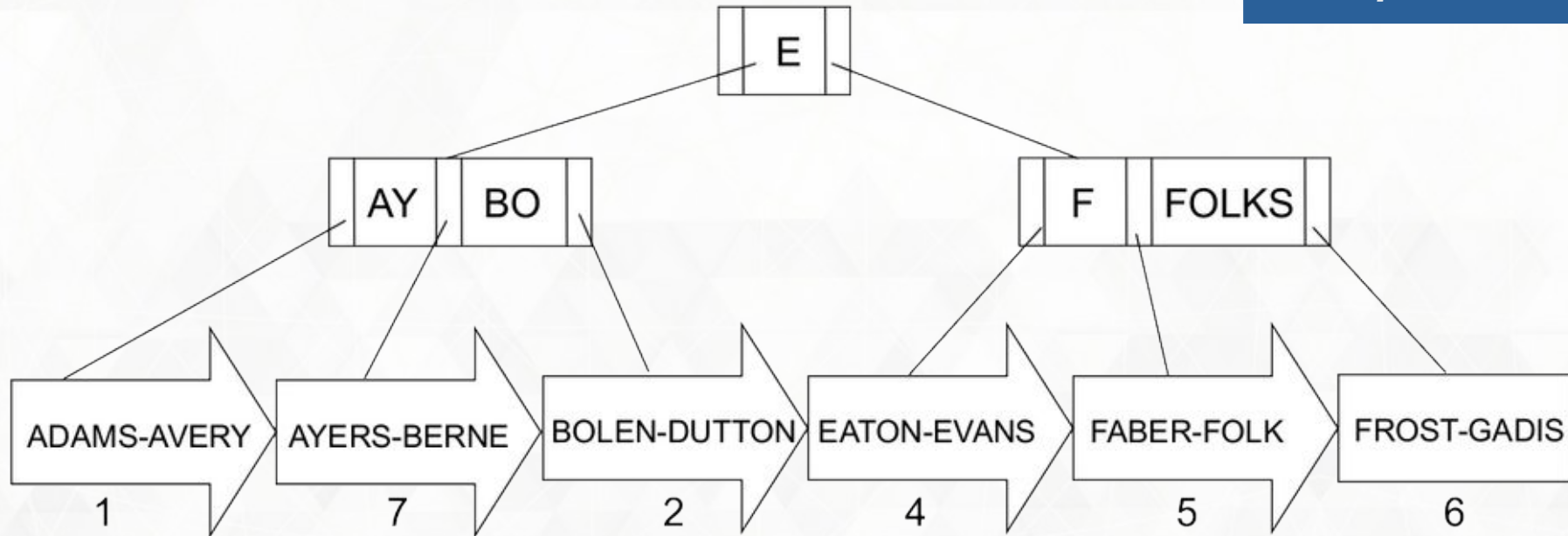
*Underflow no
sequence set 2*



Manipulando a Árvore B+

- Remoção de CAEL

*Underflow no
sequence set 2*

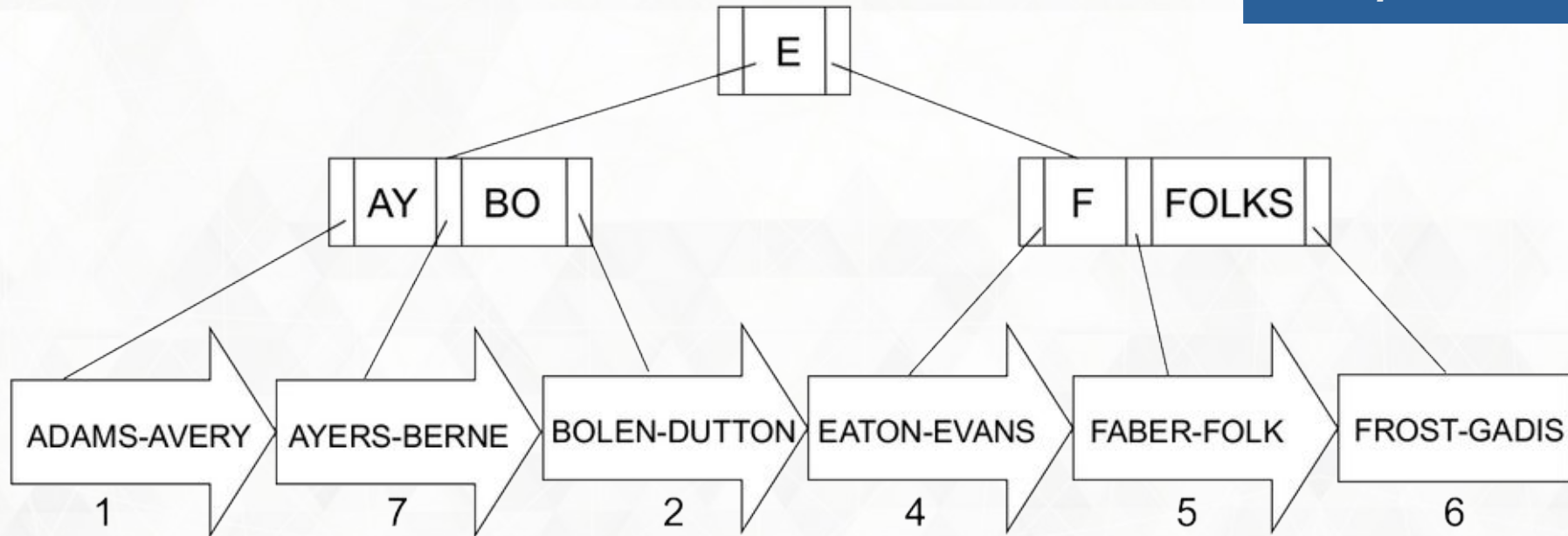


- Efeito no *sequence set*
 - concatenação dos blocos 2 e 3

Manipulando a Árvore B+

- Remoção de **CAEL**

*Underflow no
sequence set 2*



- Efeito na **Árvore B+**

- remoção de **CAM** e concatenação de nós

Inserção e Remoção

- Primeiro passo: *Sequence Set*
 - inserir ou remover o dado
- Tratar, caso necessário
 - split
 - concatenação
 - redistribuição



As alterações são sempre realizadas a partir
do **arquivo de dados**

Inserção e Remoção

- Segundo passo: Árvore-B +
 - **se split** no sequence set
inserir um **novo separador** no índice
 - **se concatenação** no sequence set
remover um separador do índice
 - **se distribuição** no sequence set
alterar o valor do separador no índice

Observações Adicionais

Tamanho físico de
um **nó no índice**
(i.e., árvore-B +)

=

Tamanho físico de um
**bloco no sequence
set**

- Escolha direcionada pelos mesmos quesitos
 - tamanho do bloco
 - características do disco
 - quantidade de memória disponível
- Facilidade para a implementação da árvore-B + virtual (várias páginas em RAM)
- Uso de um mesmo arquivo para armazenar os blocos do índice e os blocos do sequence set
 - evita *seeks* entre dois arquivos separados

Exercícios de Árvore B+

- **Considere:**
- Árvore-B + (index set)
 - Ordem: 3
- Blocos (sequence set)
 - número máximo de registros: 4
 - número mínimo de registros: 2
 - underflow: 1 registro

Exercícios de Árvore B+

- 1. Quais os separadores dos *sequence sets*?



- 2. Construa a *Árvore B+*
- 3. Realize as seguintes operações
 - a) inserção de CARTER
 - b) inserção de DRAG
 - c) remoção de BIXBY
 - d) remoção de COLE