

Árvores-B⁺

SCC-203 – Algoritmos e Estruturas de Dados II

Graça Nunes

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

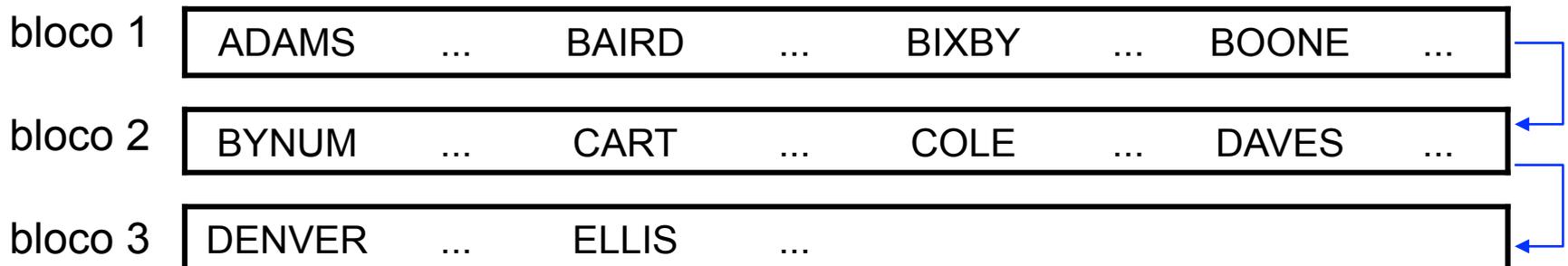
- Arquivo indexado por um índice **árvore-B**
 - acesso **indexado pela chave**: desempenho excelente 😊 - ordem logarítmica
 - acesso **sequencial** aos registros ordenados pela chave: desempenho péssimo ☹️ - ordem **linear**
- Arquivo com **registros ordenados** pela chave
 - processamento **sequencial** (acessar todos registros): **apropriado** 😊 (bufferização)
 - processamento **randômico**: **inapropriado** ☹️ - **logarítmico** (ordem 2)

Um Modelo Híbrido

- Objetivo:
- Organizar um arquivo de modo que seja eficiente tanto para processamento sequencial quanto aleatório

Um Modelo Híbrido

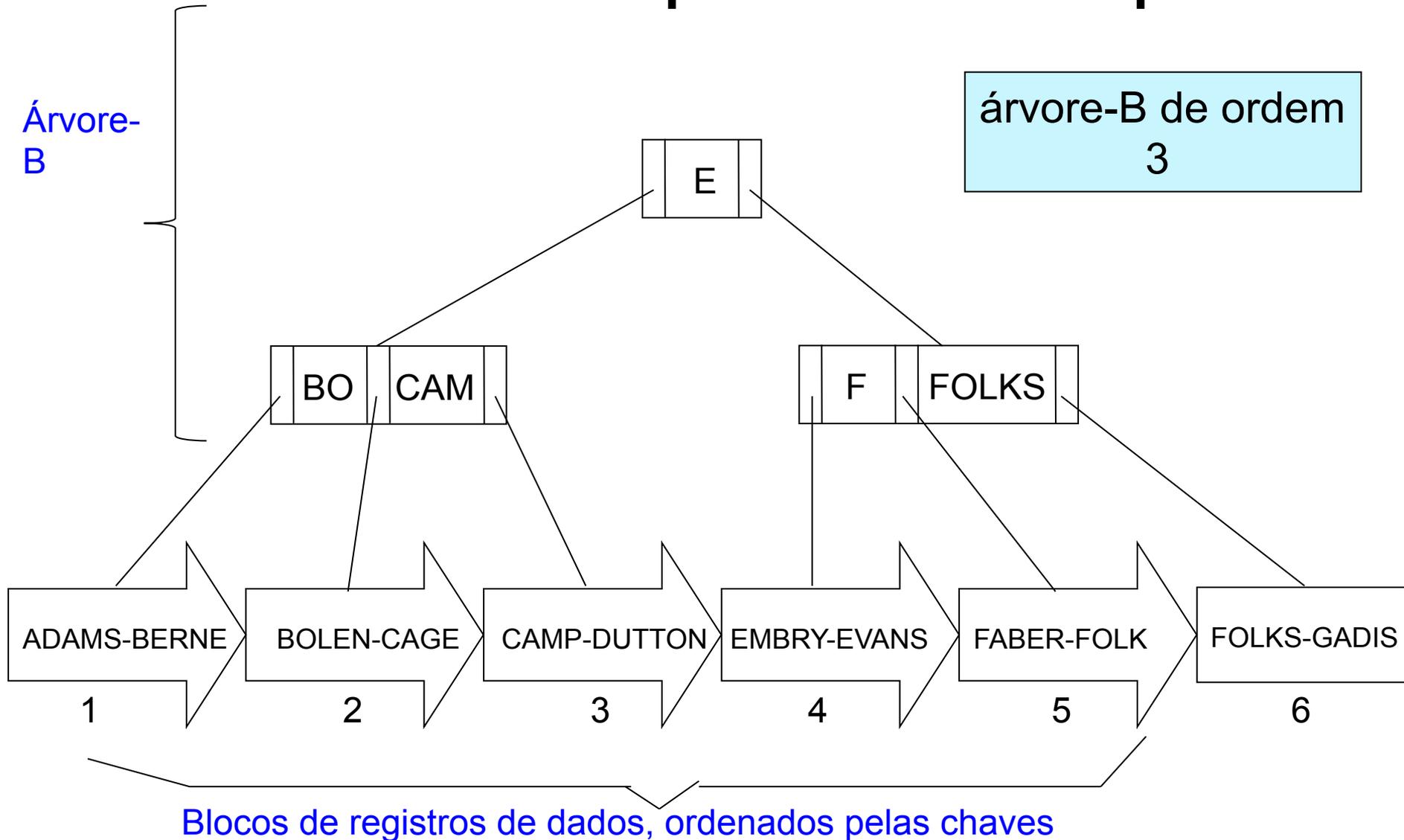
- Arquivo de dados é organizado em blocos de tamanho fixo, de registros sequenciais, ordenados pelas chaves, e encadeados (*Sequence Set*)
 - Privilegiando o acesso sequencial



Um 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 prefixos simples

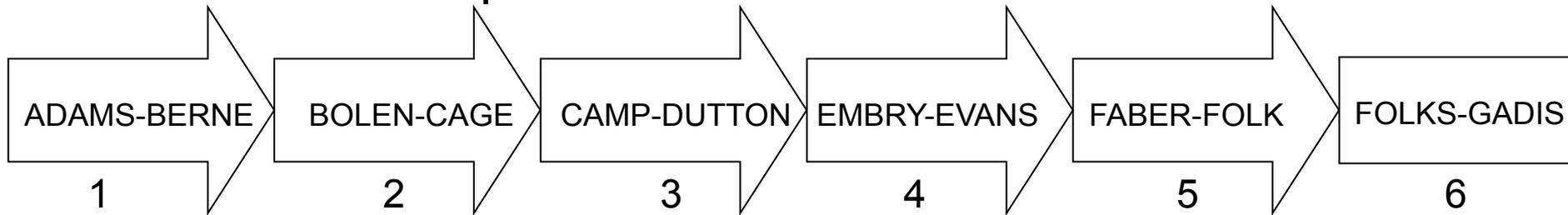


Árvores-B⁺

- Todas as chaves estão apenas nas páginas folhas;
- As páginas não folhas abrigam apenas separadores de chaves (prefixos mínimos de tamanhos variáveis, para economizar 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 para encontrar a chave



chave	bloco
BERNE	1
CAGE	2
DUTTON	3
EVANS	4
FOLK	5
GADIS	6

Índice de 1 nível

- registros de tamanho fixo
- contém a chave para o último registro no 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**

- o 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**
- blocos **logicamente adjacentes** não estão (necessariamente) fisicamente adjacentes

- Garante acesso sequencial ao arquivo

Problema 1

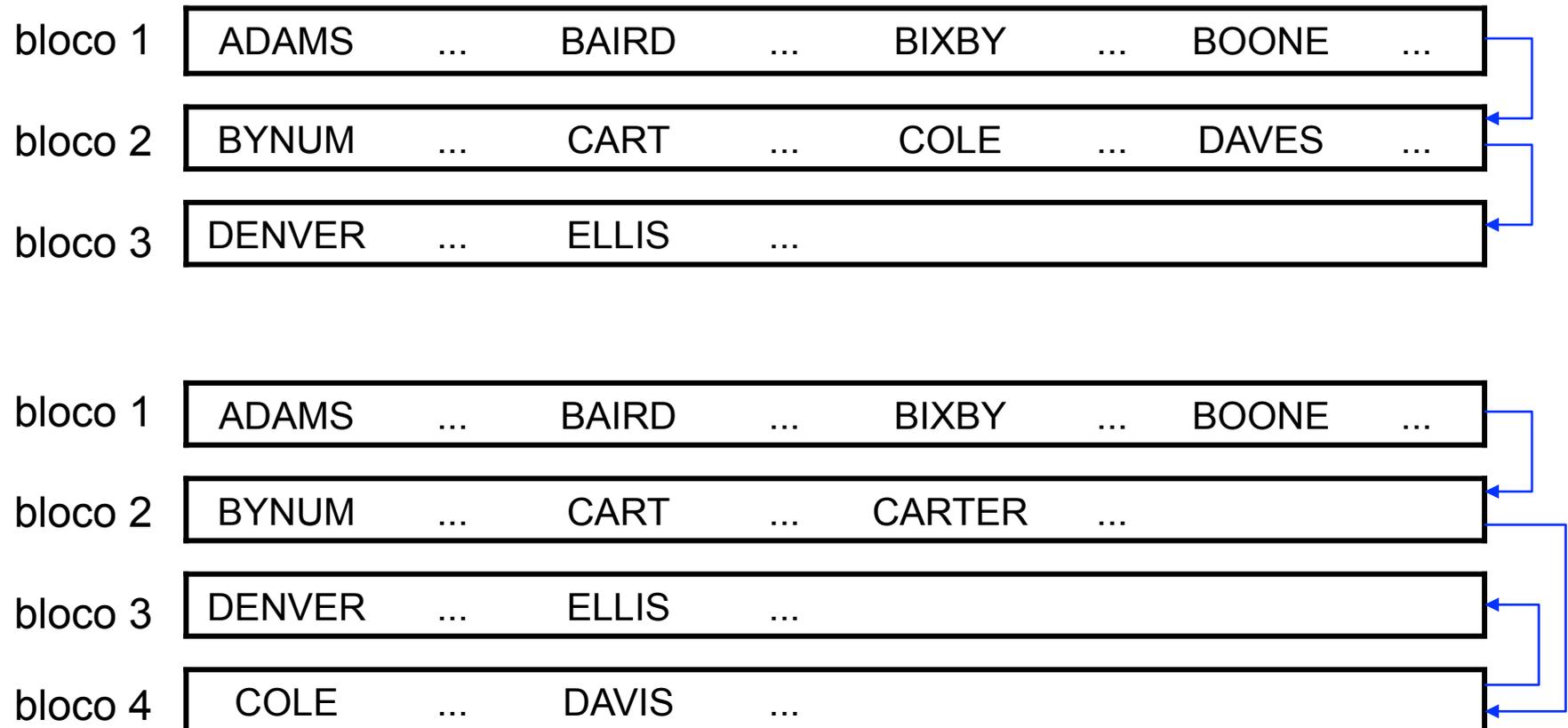
- Inserção de registros pode provocar *overflow* em um bloco
- Solução
 - dividir o bloco, em um processo análogo ao realizado em árvores-B
 - passos
 - divide os registros entre os dois blocos
 - rearranja os ponteiros

não existe
promoção !

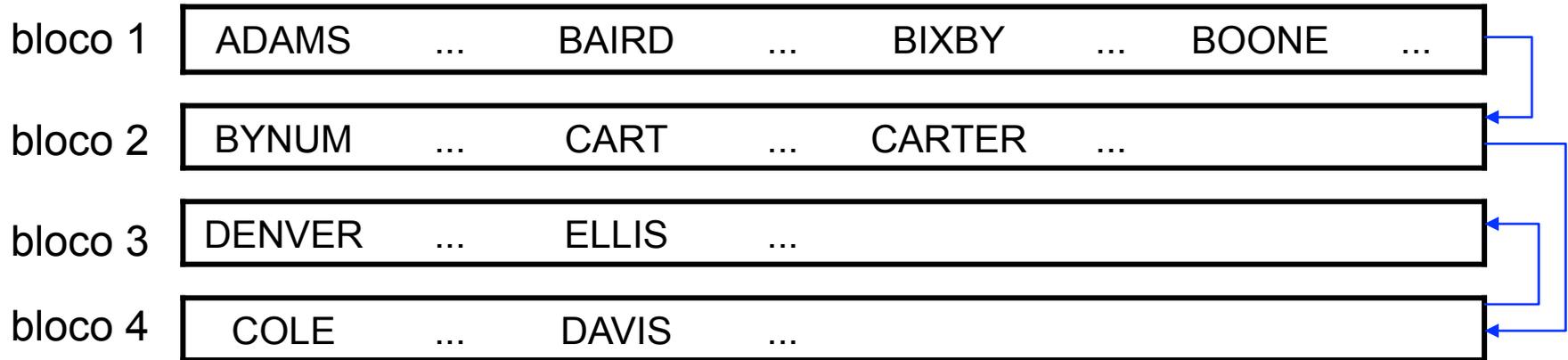
Problema 2

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

Exemplo: Inserção de CARTER



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)

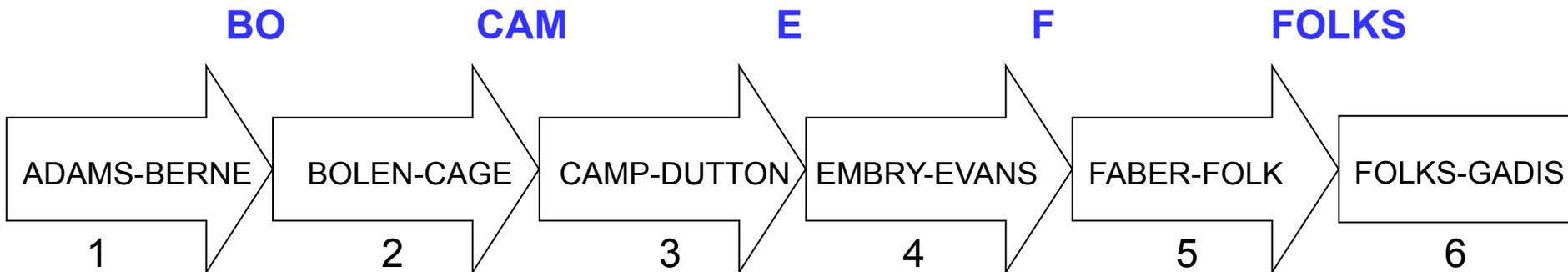
- Para localizar eficientemente um bloco com um registro particular, dada a chave do registro
- Ao invés de chaves, prefixos de chaves (de tamanho variável) compõem as páginas não folhas das árvore-B

Separadores

- **Características**

- são mantidos no índice, ao invés das chaves de busca
- possuem tamanho variável

- **Exemplo**



Separadores

- **Desafio**
 - escolher o menor separador para utilizar no índice
- Tabela de decisão

chave de busca x separador	decisão
chave < separador	procure à esquerda
chave = separador	procure à direita
chave > separador	procure à direita

Separadores

- Programa para gerar separadores mínimos

```
find_sep(char key1[], char key2[], char sep[]) {  
    while ( (*sep++ = *key2++) == *key1++);  
    *sep='\0';  
}
```

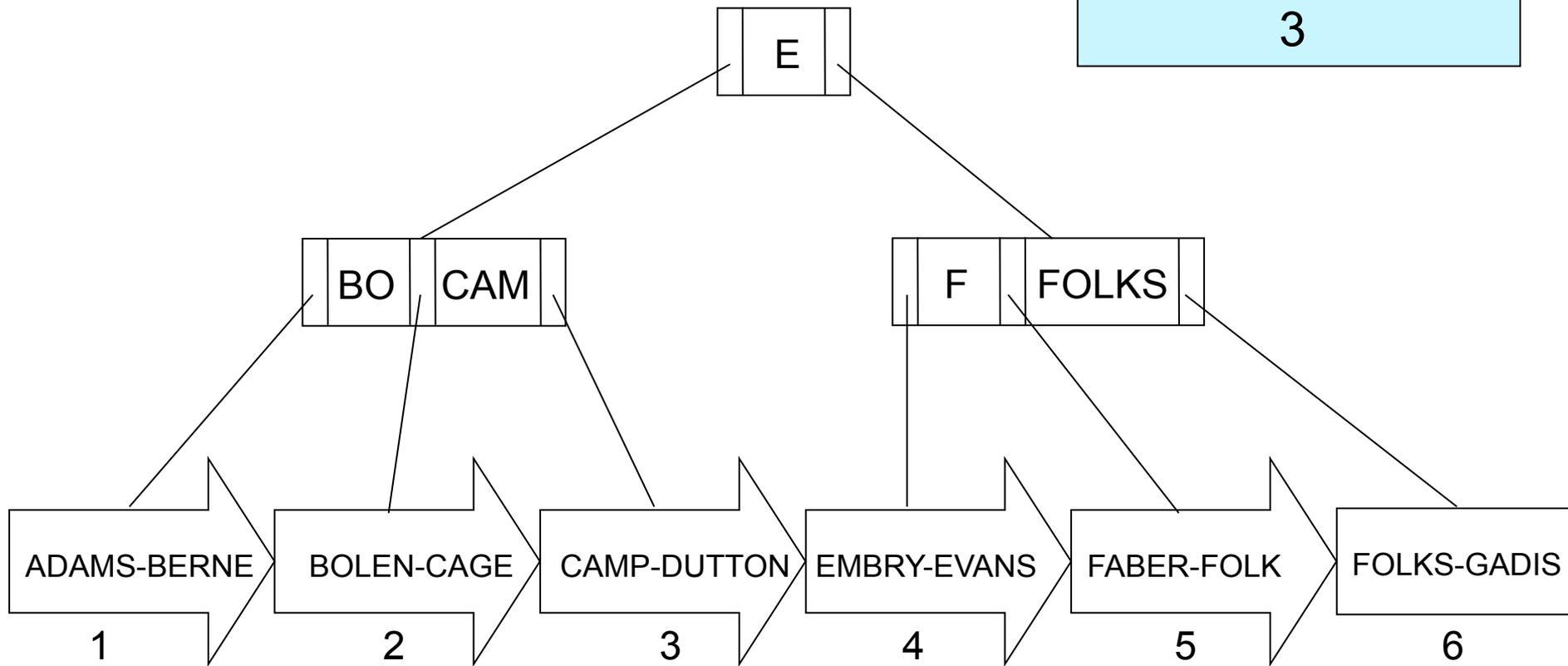
- Pode acontecer de o separador mínimo ser uma chave inteira

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
 - Necessidade de **maior controle** do tamanho do nó e de onde começa e termina cada cadeia separadora

Árvore-B+ de prefixos simples

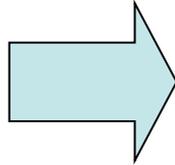
árvore-B de ordem
3



Manutenção

- Cenários

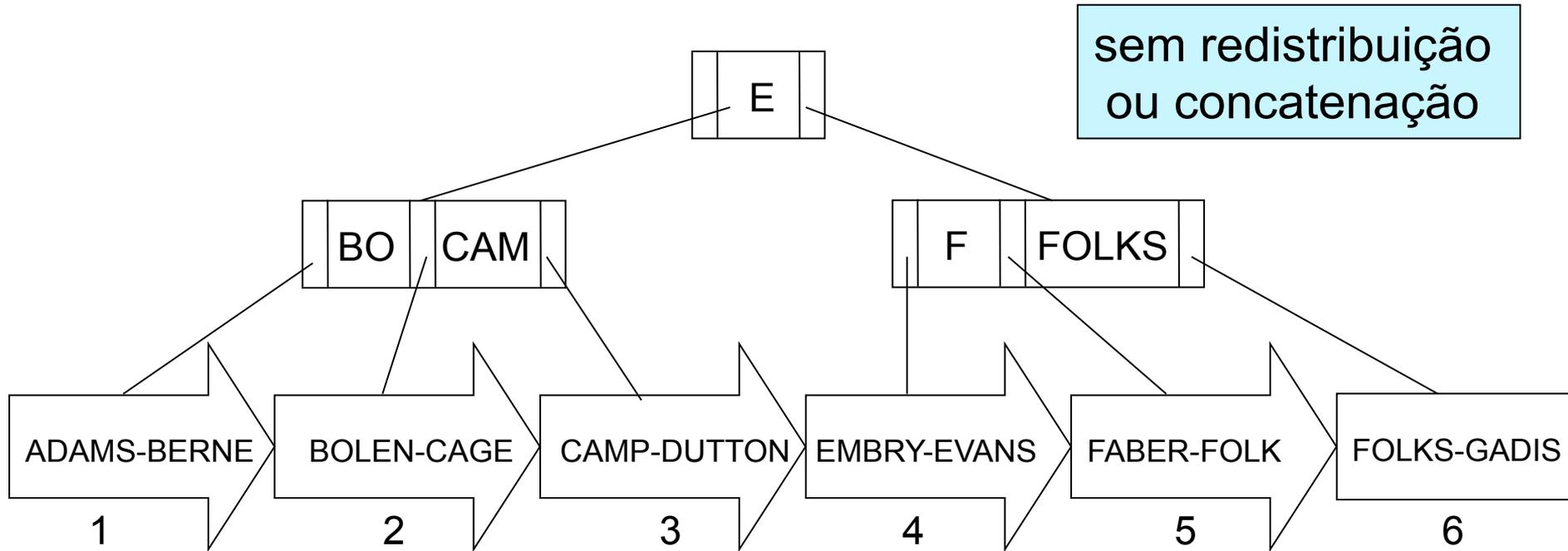
- inserção
- remoção
- *overflow*
- *underflow*



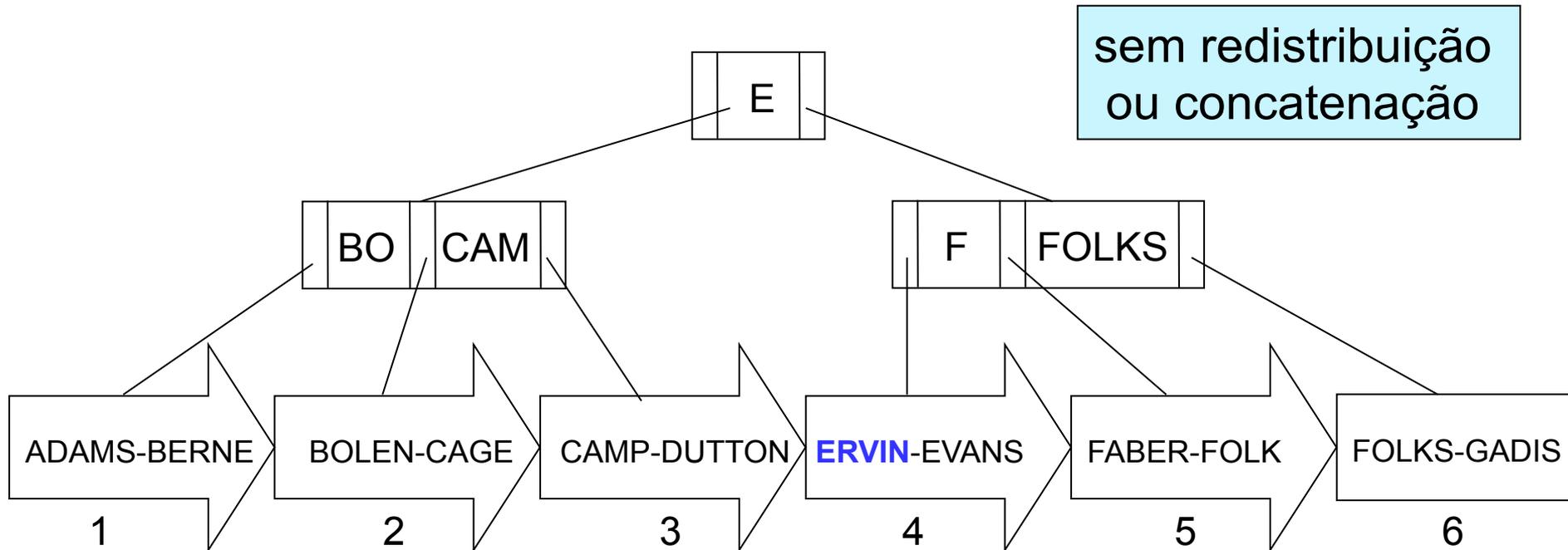
- Efeitos colaterais

- *sequence set*
- árvore-B+

Remoção de EMBRY

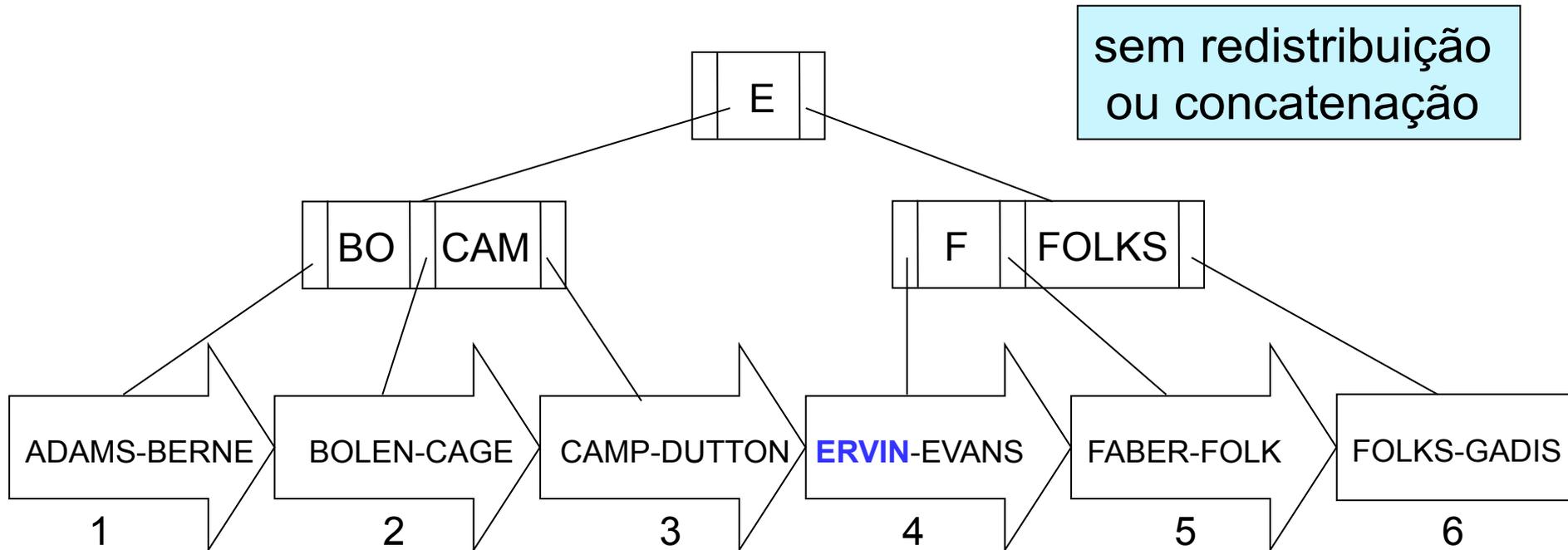


Remoção de EMBRY



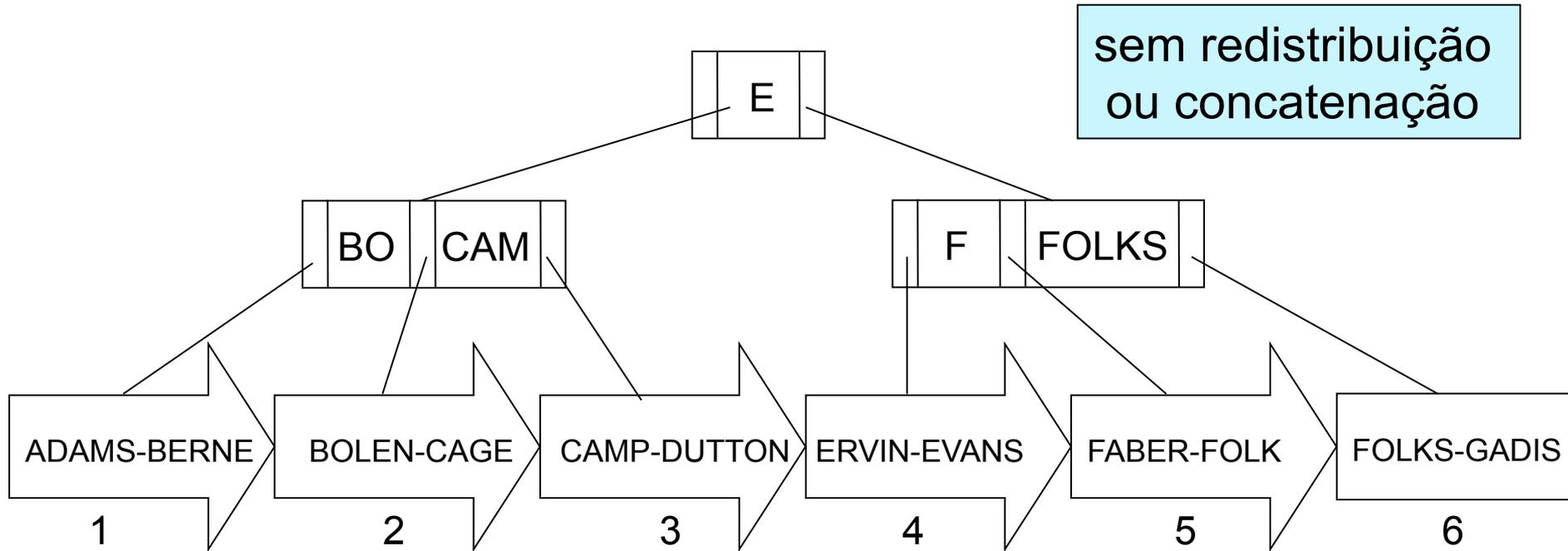
- Efeito no *sequence set*
 - limitado a alterações no bloco 4

Remoção de EMBRY

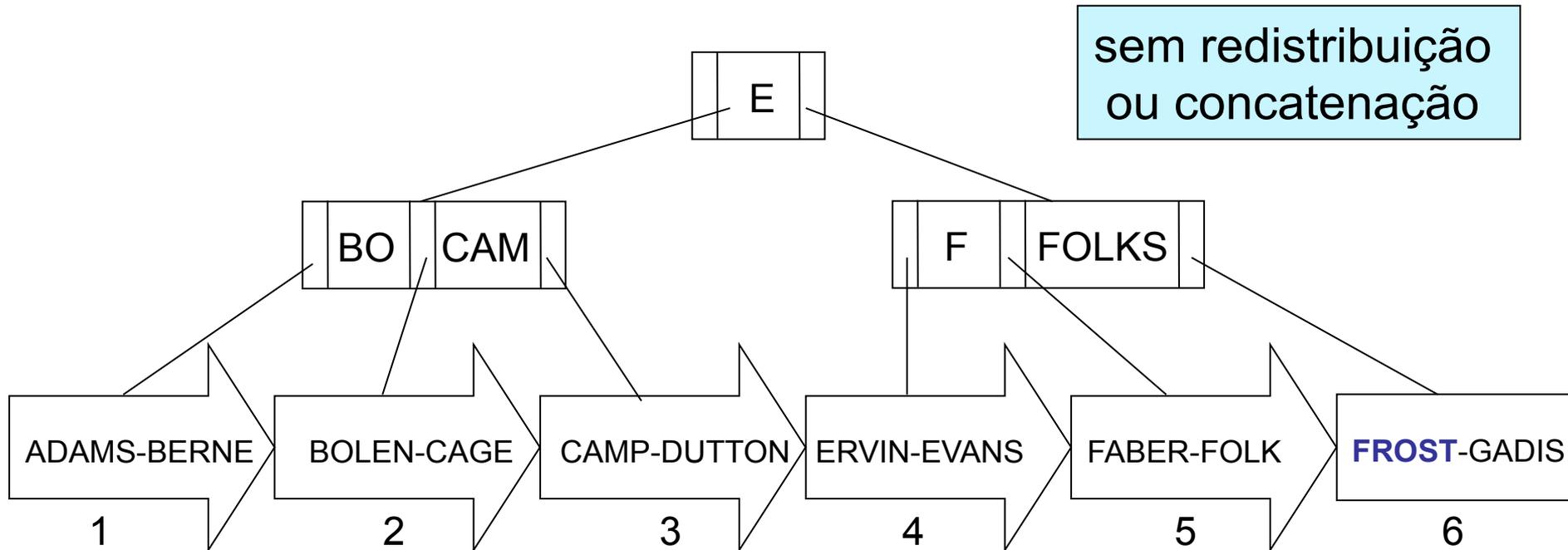


- Efeito na **árvore-B+**
 - nenhum: E é uma boa chave separadora

Remoção de FOLKS

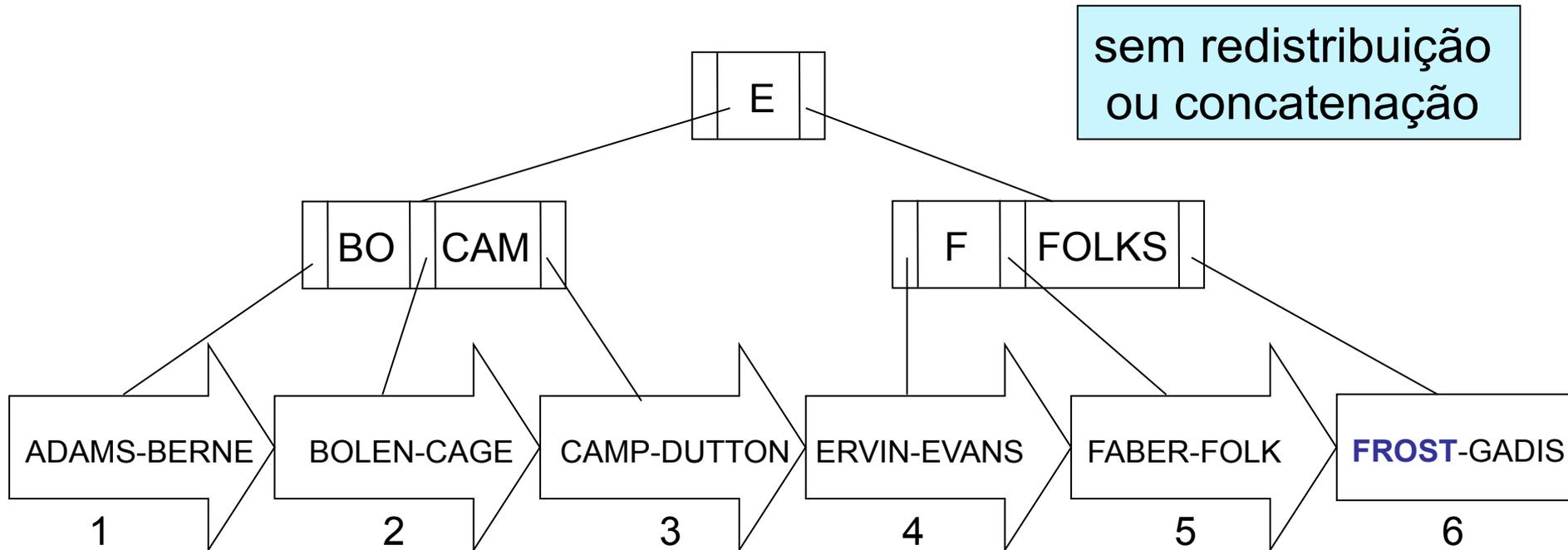


Remoção de FOLKS



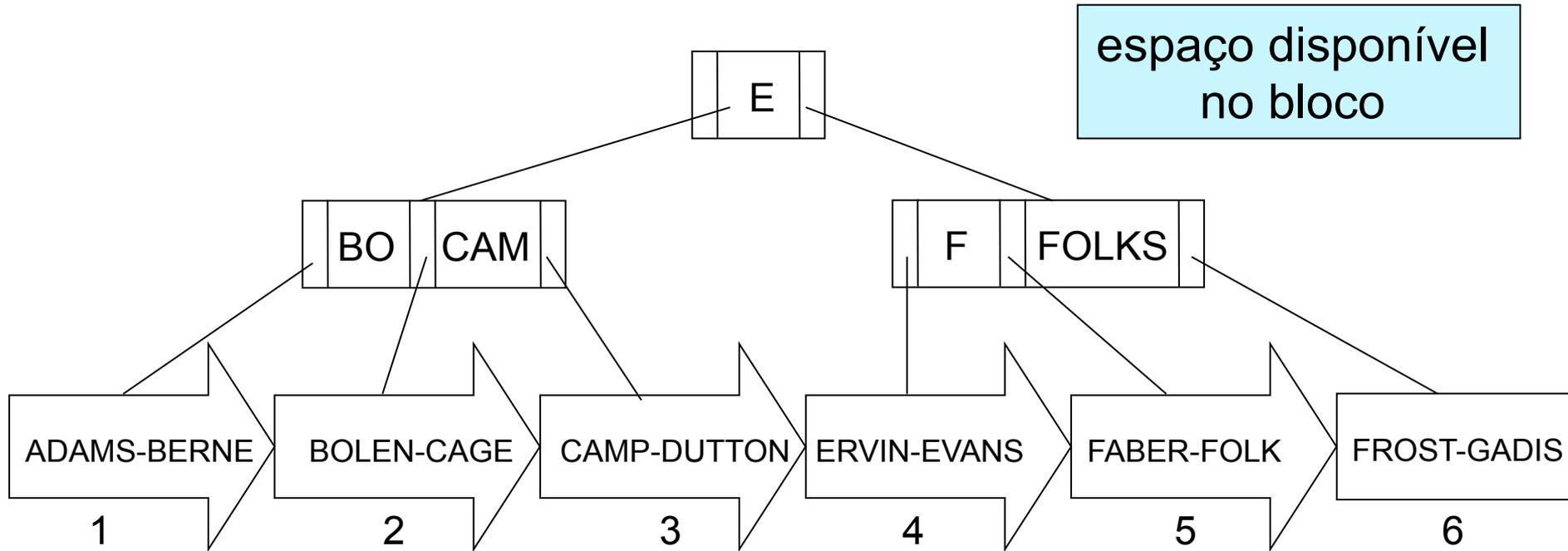
- Efeito no *sequence set*
 - limitado a alterações no bloco 6

Remoção de FOLKS

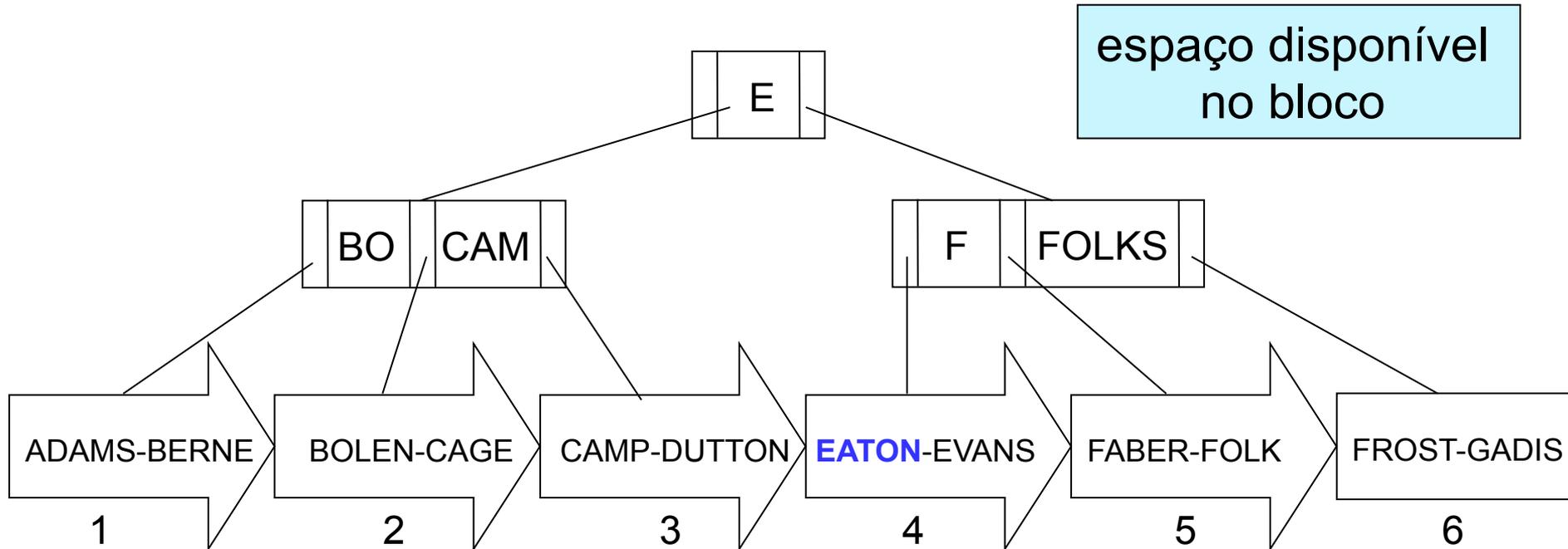


- Efeito na **árvore-B+**
 - nenhum: custos elevados se fosse arrumar

Inserção de EATON

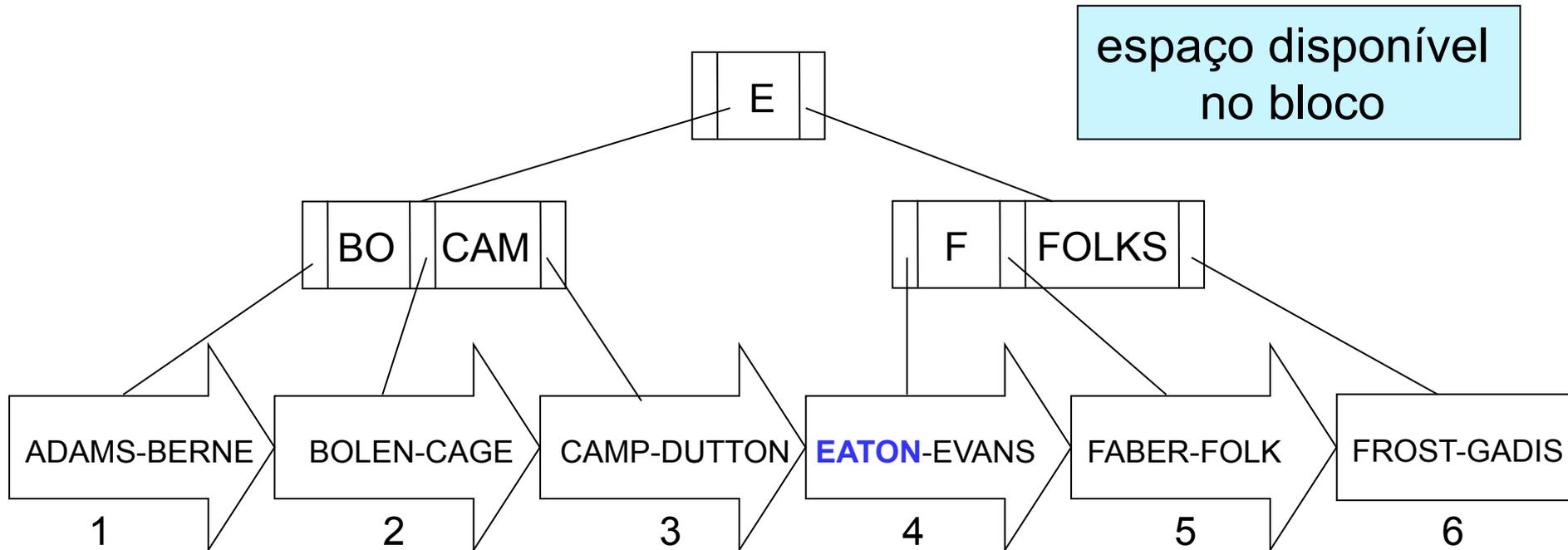


Inserção de EATON



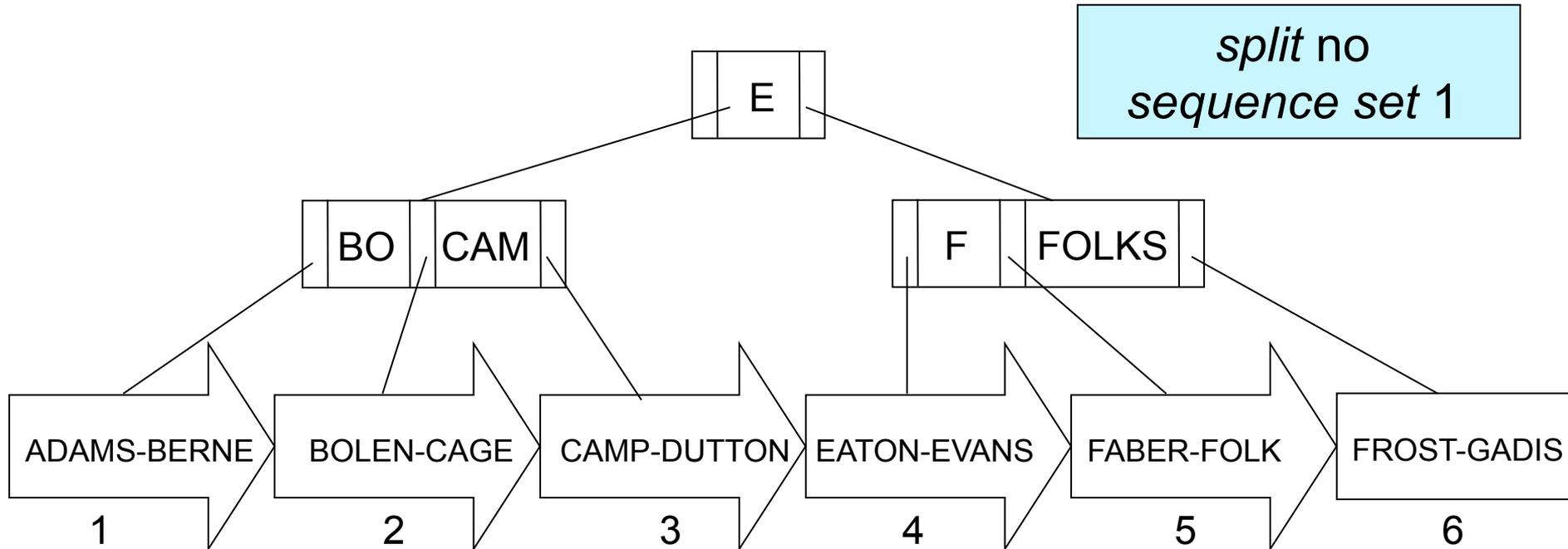
- Efeito no *sequence set*
 - limitado a alterações no bloco 4

Inserção de EATON

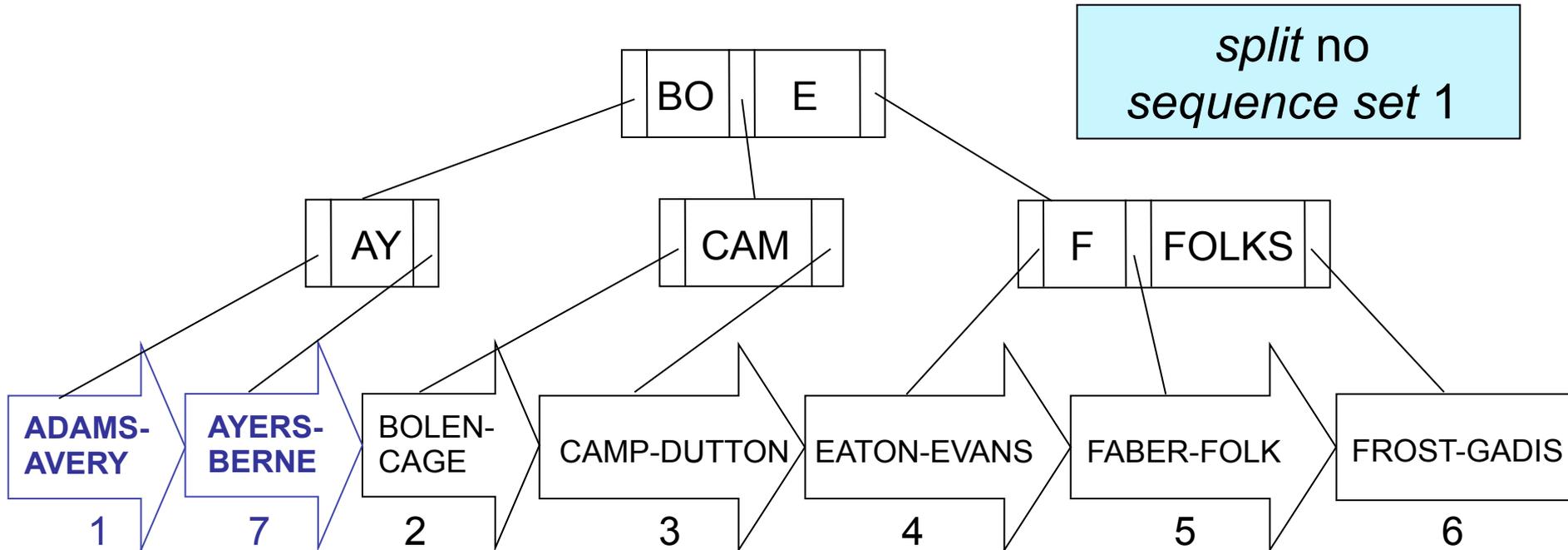


- Efeito na **árvore-B+**
 - nenhum: E é uma boa chave separadora

Inserção de AVERY

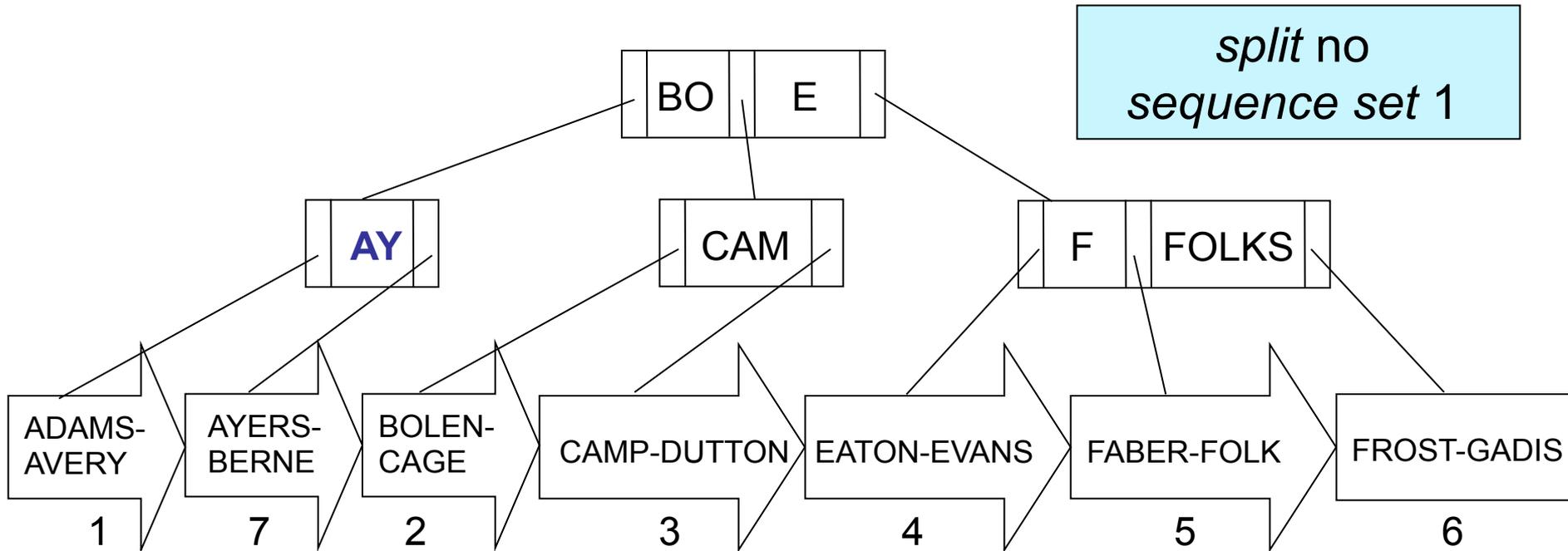


Inserção de AVERY



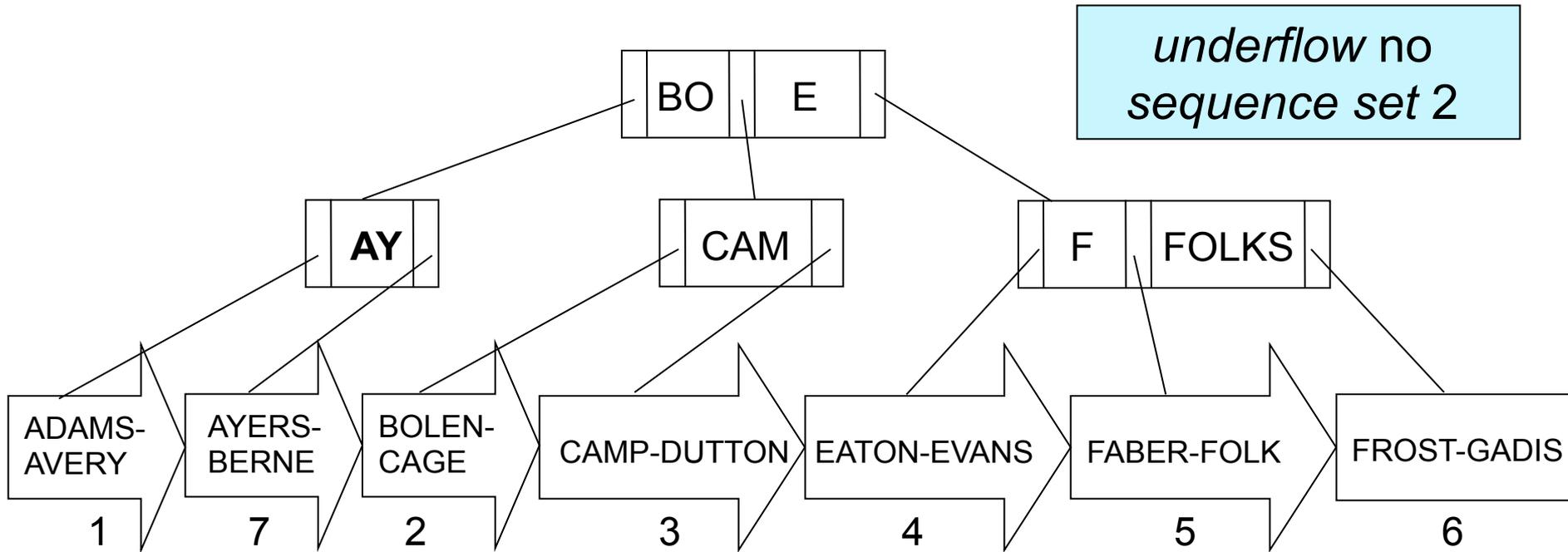
- Efeito no *sequence set*
 - dados do bloco 1 + AVERY distribuídos entre os blocos 1 e 7

Inserção de AVERY

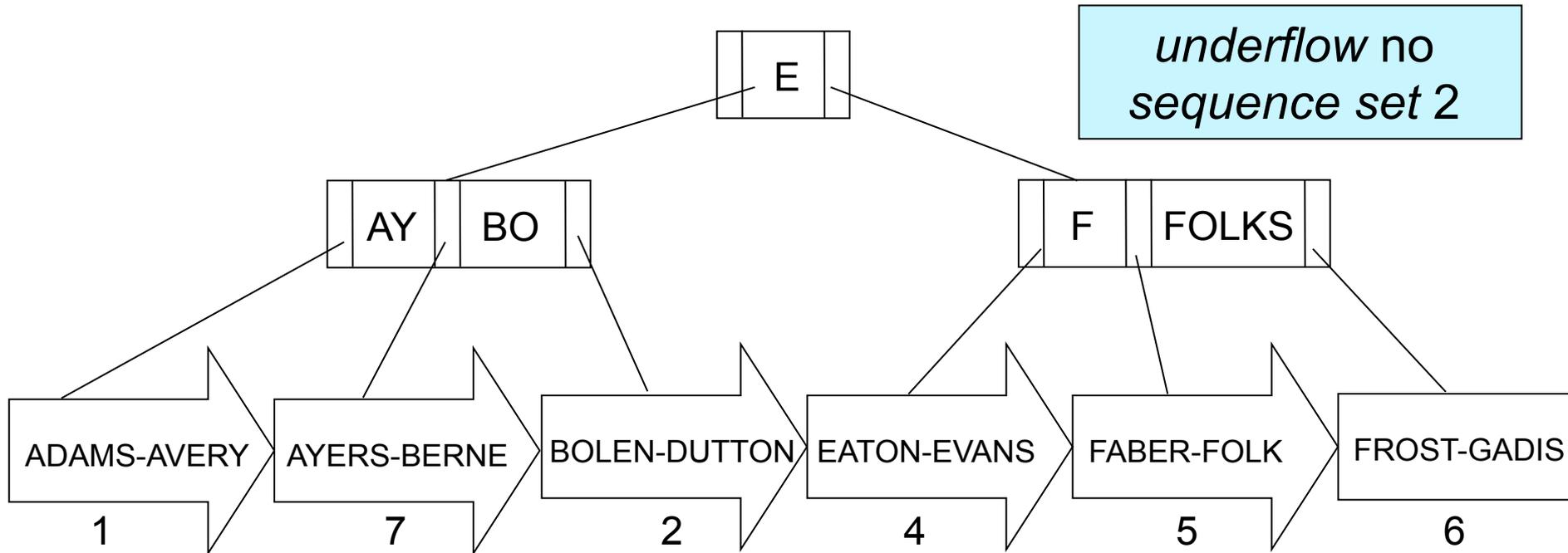


- Efeito na **árvore-B+**
 - separador adicional AY

Remoção de CAEL

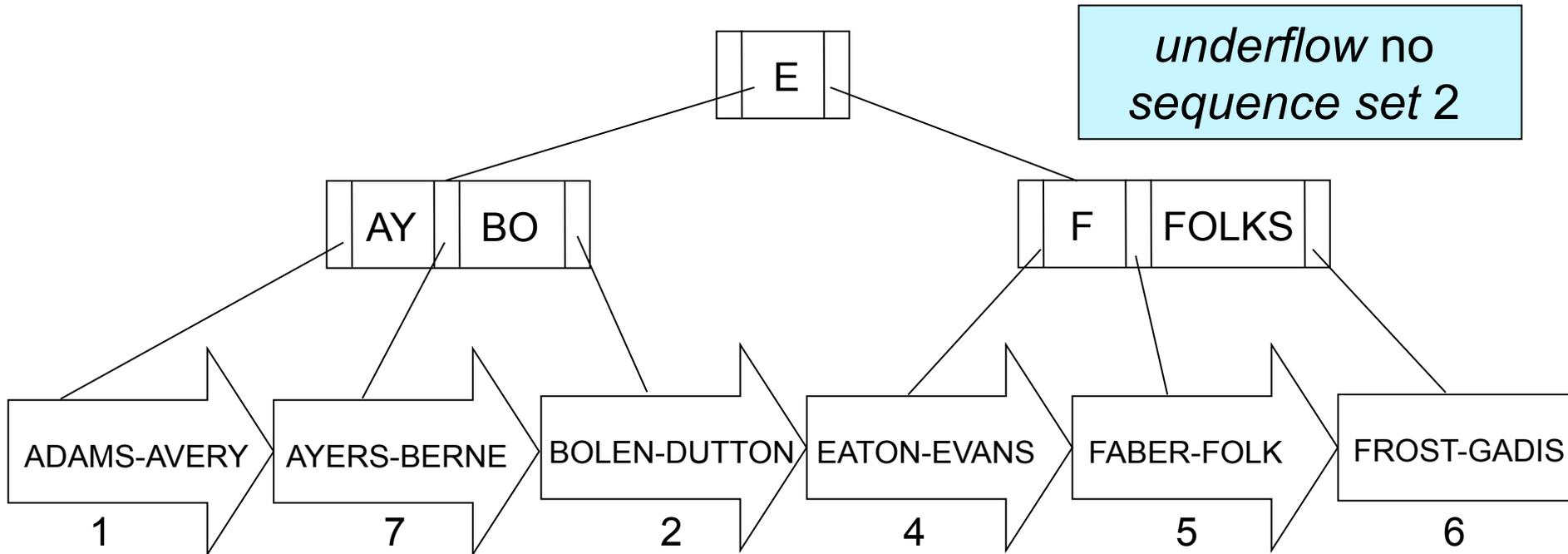


Remoção de CAEL



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

Remoção de CAEL



- 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

alterações são sempre realizadas a partir do arquivo de dados

Inserção e Remoção

- Segundo passo: *Árvore-B⁺*
 - se *split* no *sequence set*
inserir um novο 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

Observações Adicionais

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

=

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

- Facilidade para a implementação da árvore-B⁺ virtual
 - Várias páginas em RAM

Observações Adicionais

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

- Uso de um mesmo arquivo para armazenar os blocos do índice e os blocos do *sequence set*
 - evita *seeks* entre dois arquivos separados

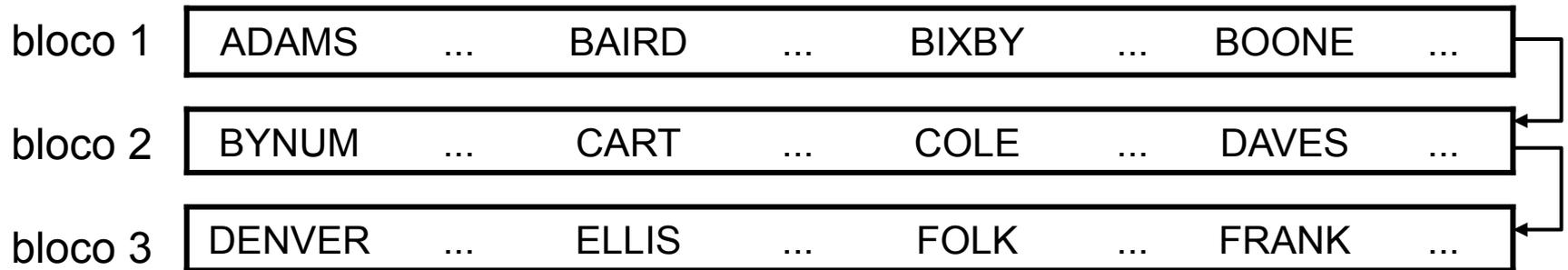
Exercícios de Árvore-B⁺

Características

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

Exercícios

1. Quais os separadores dos *sequence sets*?



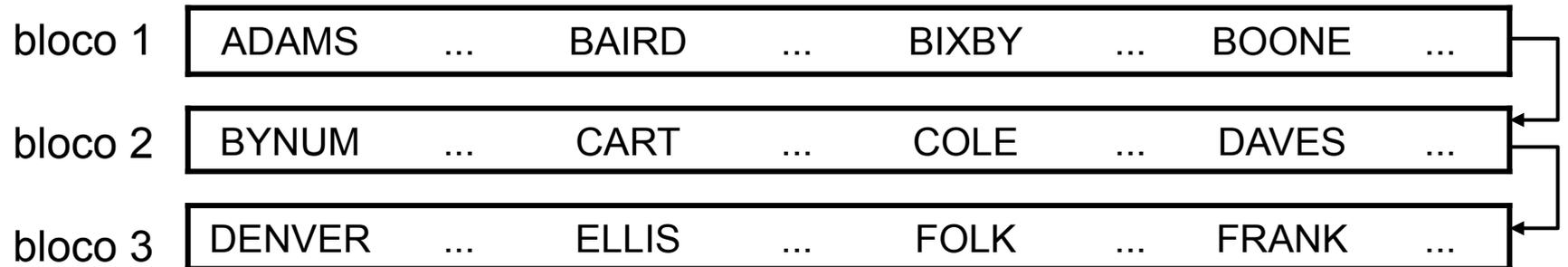
2. Construa a árvore-B⁺

3. Realize as seguintes operações

- inserção de CARTER
- inserção de DRAG
- remoção de BIXBY
- remoção de COLE

Resposta

1. Quais os separadores dos *sequence sets*?

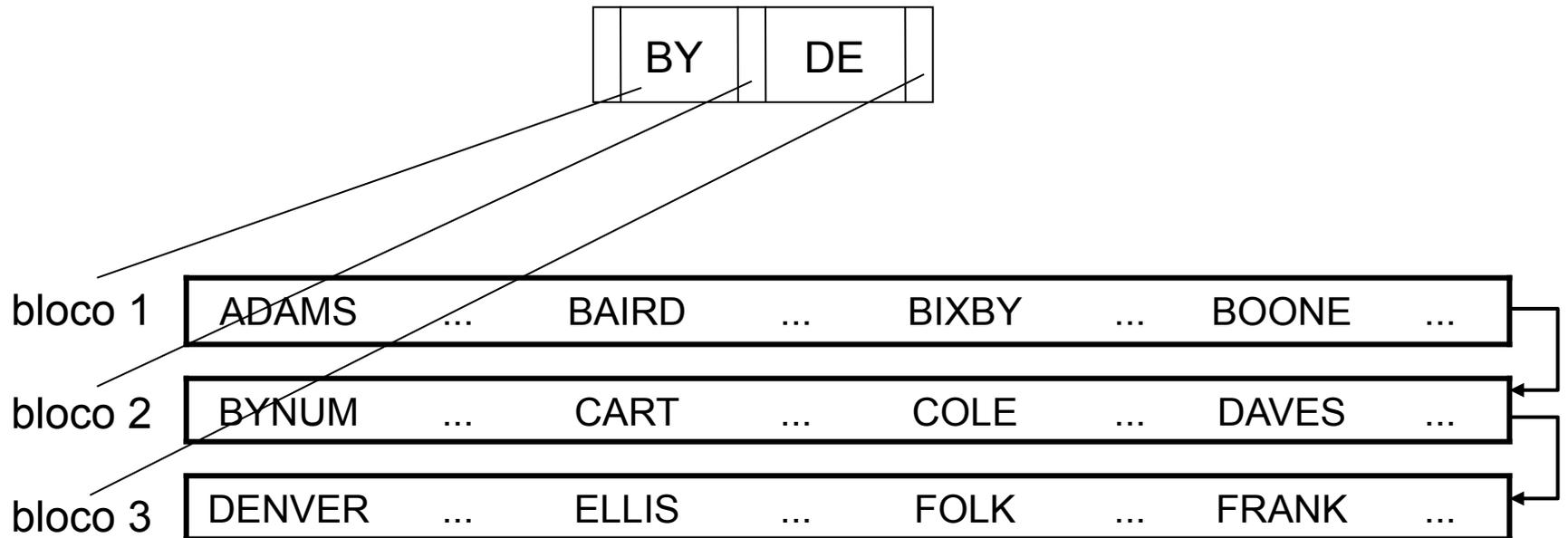


BY

DE

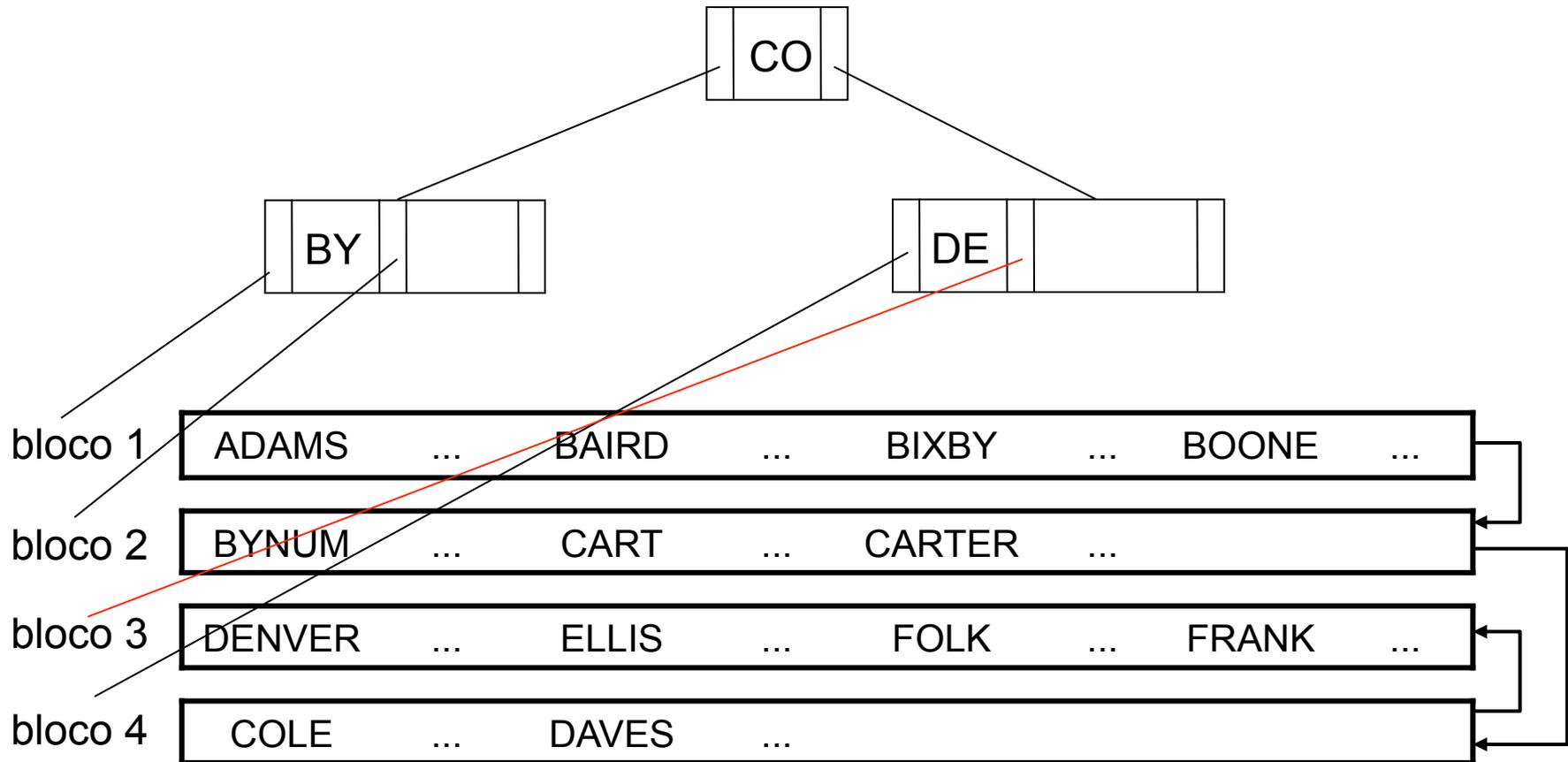
Resposta

2. Construa a árvore-B⁺



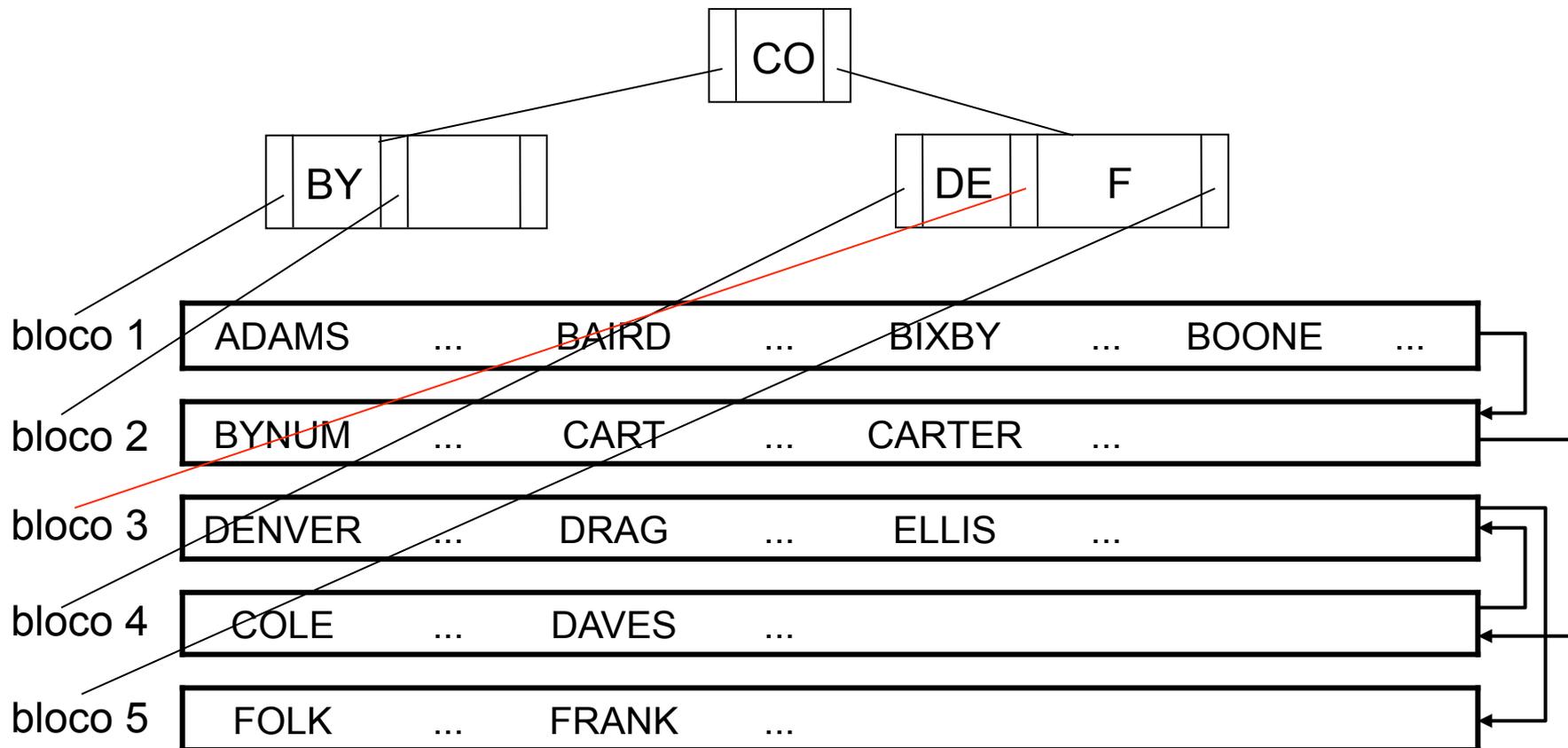
Resposta

3. inserção de CARTER



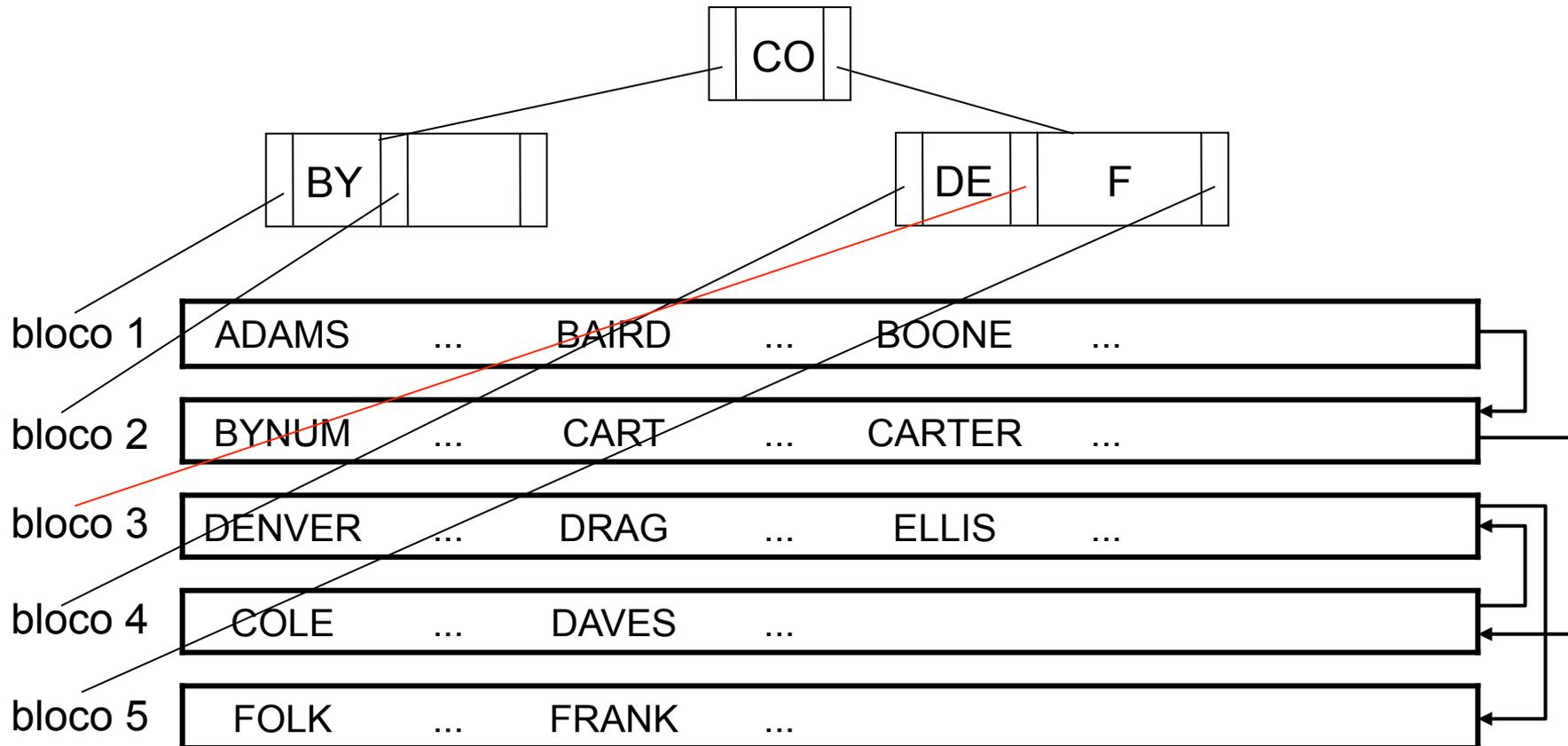
Resposta

3. inserção de DRAG



Resposta

3. remoção de BIXBY



Resposta

3. remoção de COLE

