CS 68: BIOINFORMATICS

Prof. Sara Mathieson
Swarthmore College
Spring 2018

Outline: Mar 8

- Recap Sankoff's algorithm (weighted parsimony)
- Begin: perfect phylogeny
- Gusfield's algorithm

Notes:

- Midterm solutions posted

Recap parsimony

Ancestral state reconstruction via parsimony

- **Input:** rooted, binary phylogenetic tree and leave labels

- **Output:** internal vertex labels that minimize the parsimony score (weighted or unweighted)

Ancestral state reconstruction via parsimony

- **Input:** rooted, binary phylogenetic tree and leave labels

- **Output:** internal vertex labels that minimize the parsimony score (weighted or unweighted)

- For Sankoff we need a mutational scoring matrix (example with characters *a*,*b*), which does not have to be symmetric. Row is the "before" state, column is the "after" state.

σ	a	b
a	0	2
b	1	0

Recap Sankoff's algorithm

<u>Initialization</u>: Let $A_v(x)$ be the minimum parsimony score of assigning character x to vertex v. To begin $A_{\text{leaf}}(x) = 0$ if the leaf is assigned character x, and ∞ otherwise. This prevents us from ever tracing back to a non-assigned leaf label.

Recap Sankoff's algorithm

<u>Initialization</u>: Let $A_v(x)$ be the minimum parsimony score of assigning character x to vertex v. To begin $A_{\text{leaf}}(x) = 0$ if the leaf is assigned character x, and ∞ otherwise. This prevents us from ever tracing back to a non-assigned leaf label.

<u>Bottom-up recursive step</u>: Let c_1 and c_2 be the two children of vertex v. For all x in our character state set, let

$$A_v(x) = \min_y \{A_{c_1}(y) + \sigma(x, y)\} + \min_z \{A_{c_2}(z) + \sigma(x, z)\}.$$

Keep track of a back-pointer to the minimum y and z.

Recap Sankoff's algorithm

Initialization: Let $A_v(x)$ be the minimum parsimony score of assigning character x to vertex v. To begin $A_{\text{leaf}}(x) = 0$ if the leaf is assigned character x, and ∞ otherwise. This prevents us from ever tracing back to a non-assigned leaf label.

Bottom-up recursive step: Let c_1 and c_2 be the two children of vertex v. For all x in our character state set, let

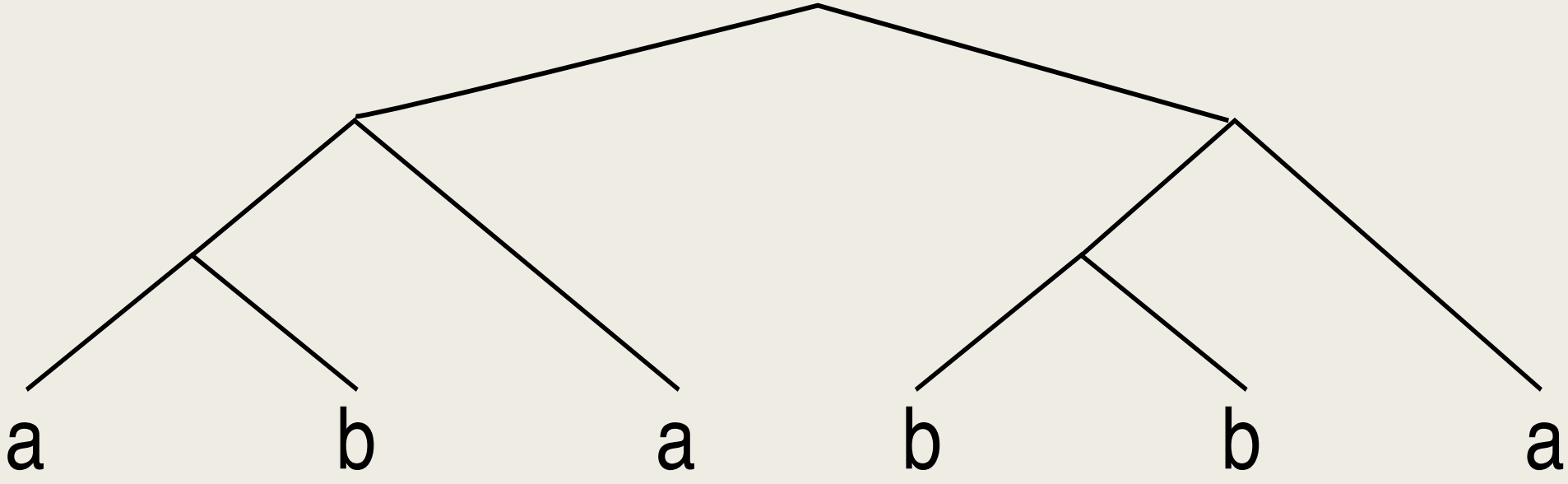
$$A_v(x) = \min_y \{A_{c_1}(y) + \sigma(x, y)\} + \min_z \{A_{c_2}(z) + \sigma(x, z)\}.$$

Keep track of a back-pointer to the minimum y and z.

Top-down traceback: Choose root state x such that $A_{\text{root}}(x)$ is the minimum. Follow back-pointers to find the assigned state of every internal vertex.

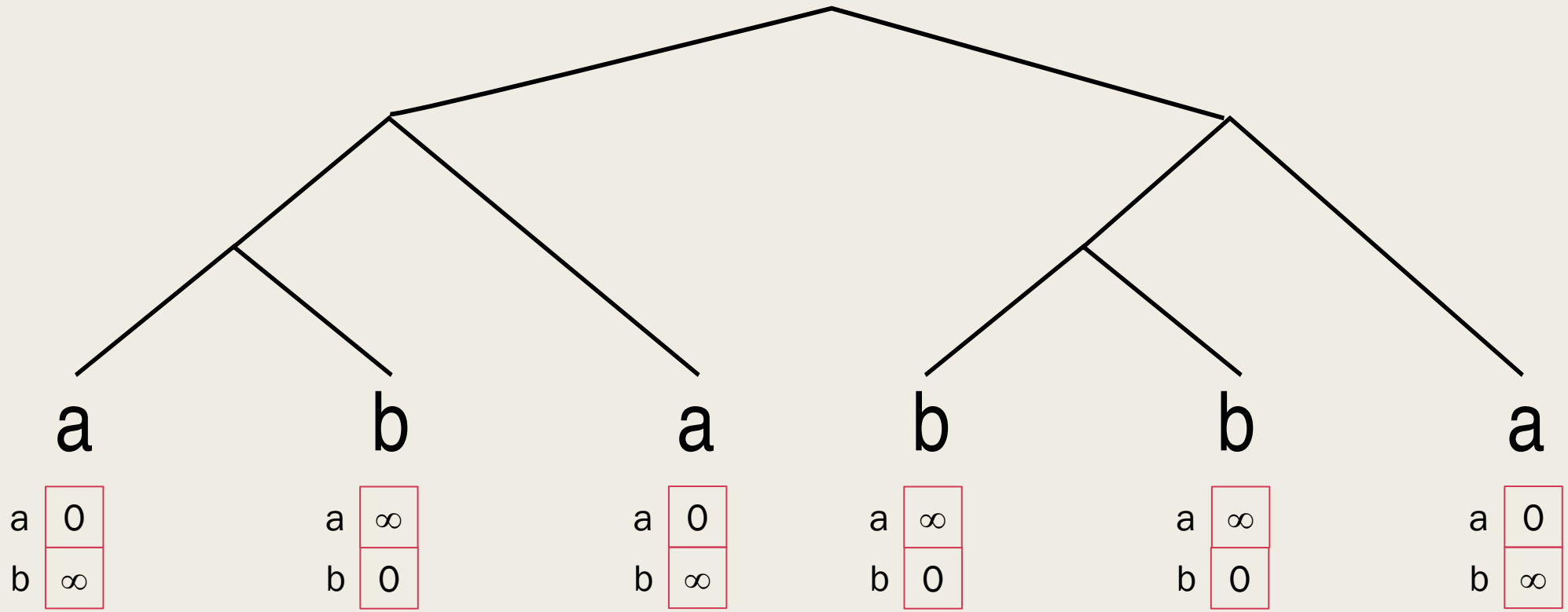
Handout 17 example

σ	a	b
a	0	2
b	1	0



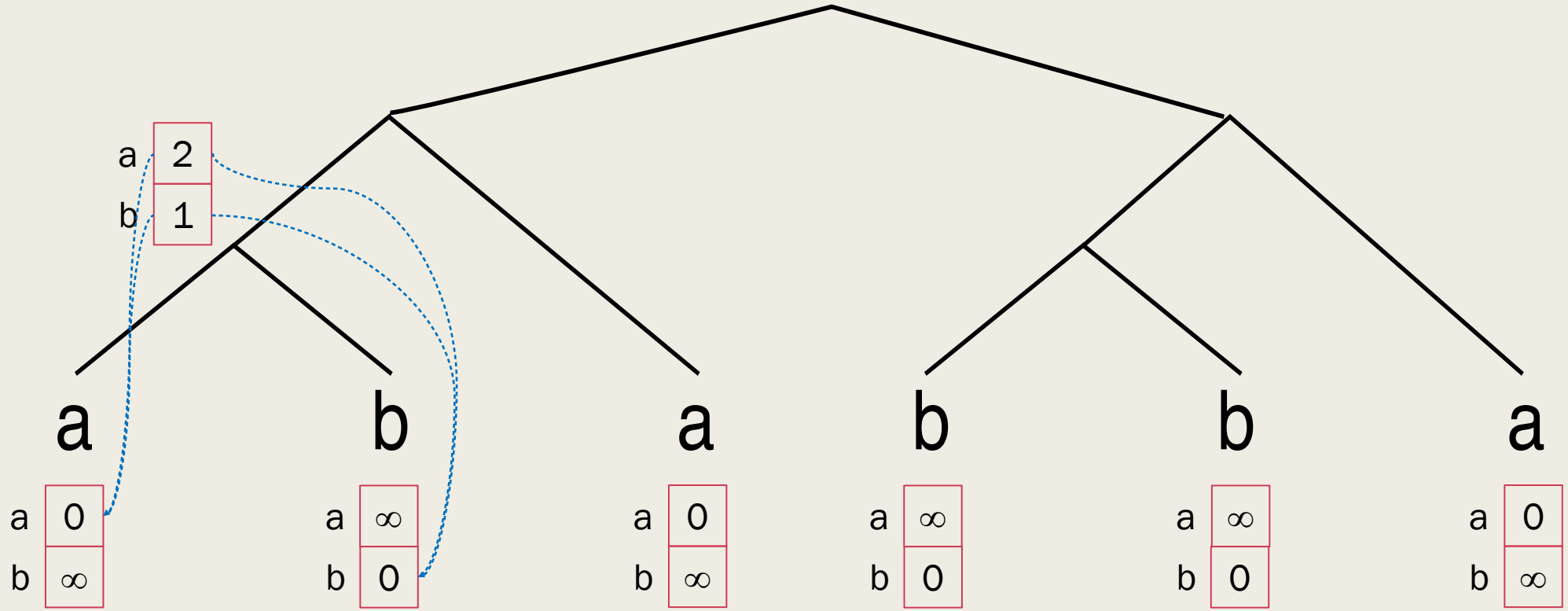
Handout 17 example

σ	a	b
a	0	2
b	1	0



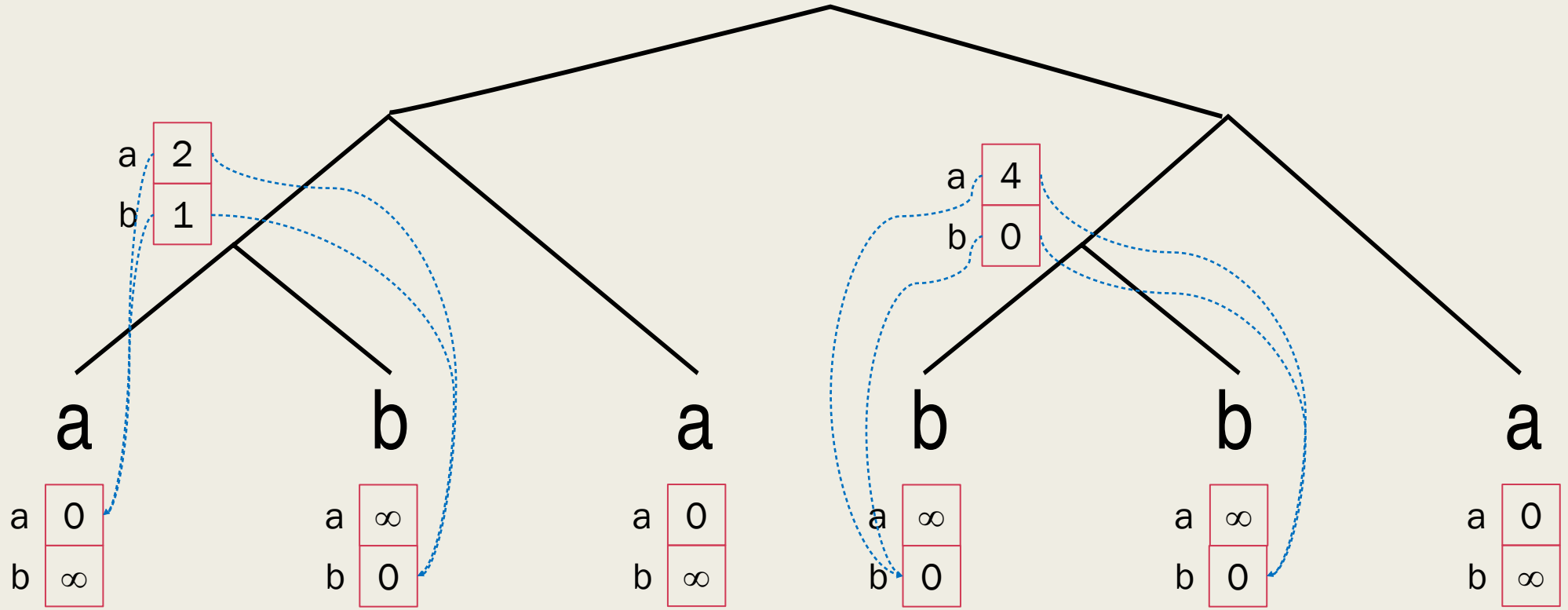
Handout 17 example

σ	a	b
a	0	2
b	1	0



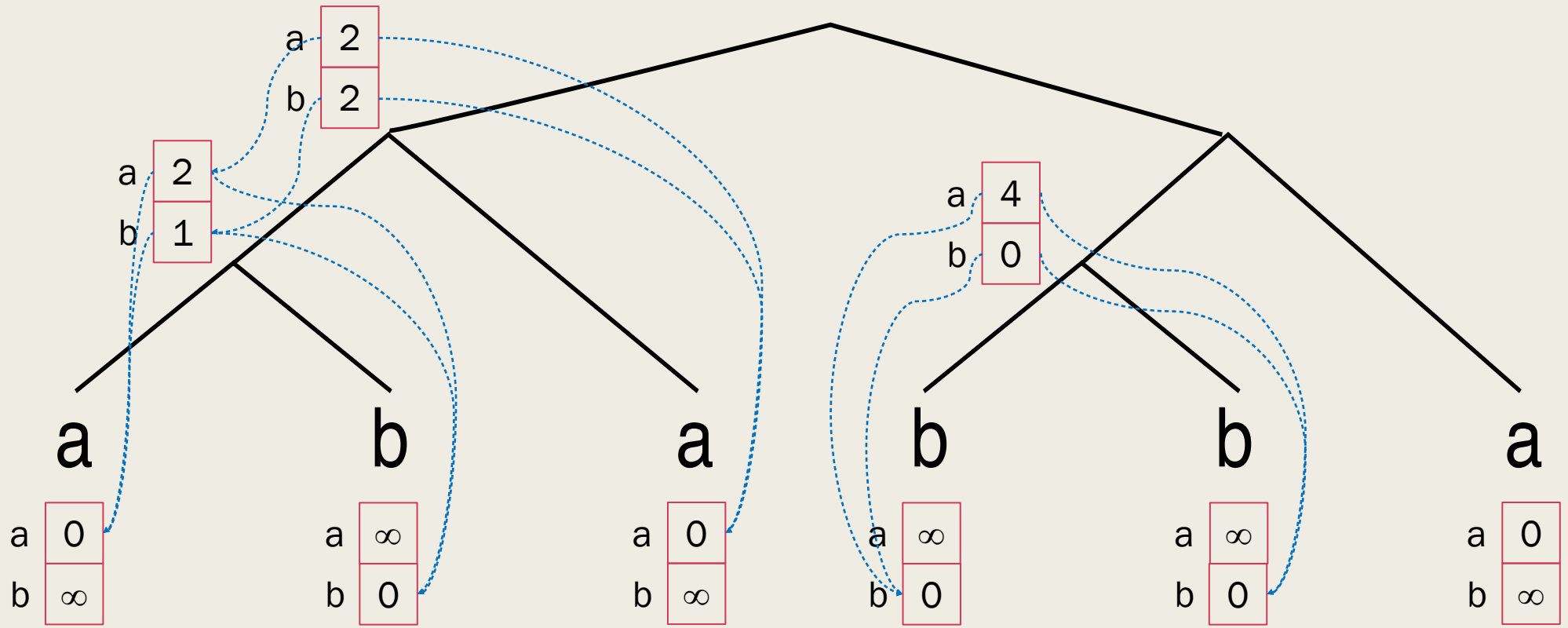
Handout 17 example

σ	a	b
a	0	2
b	1	0



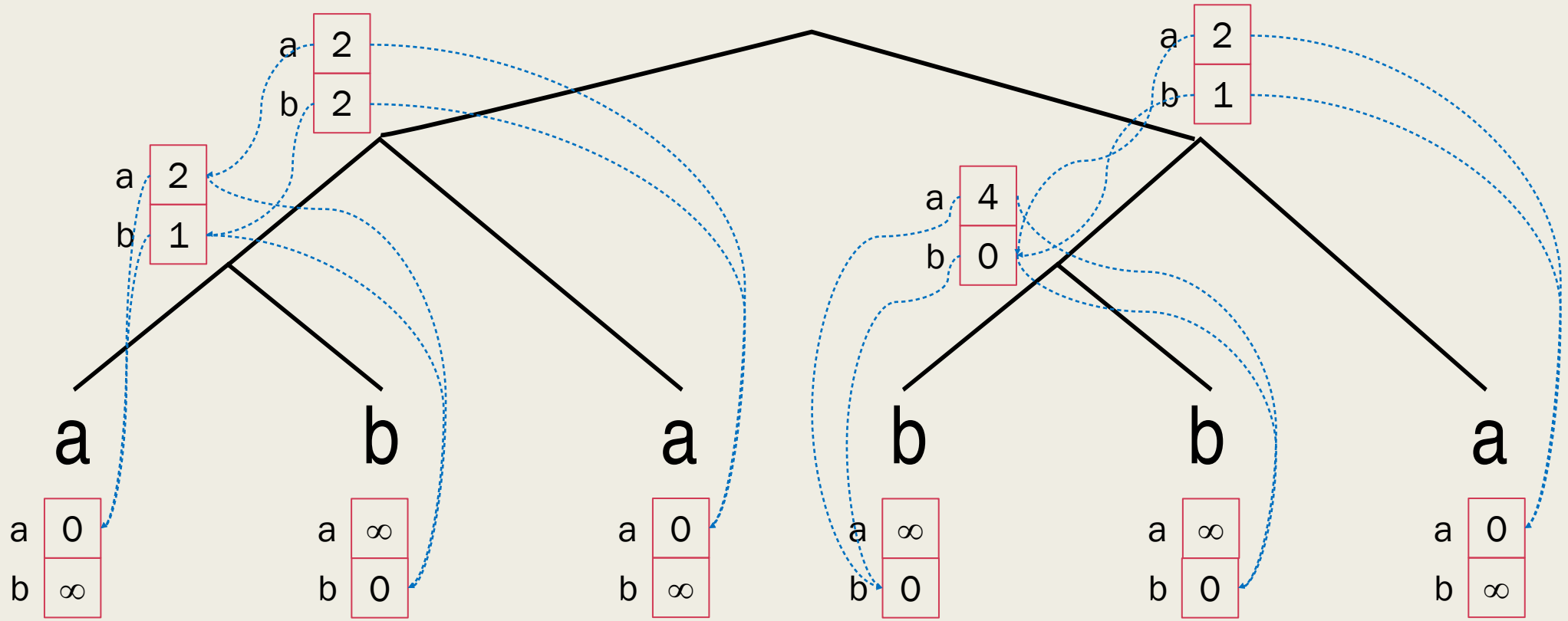
Handout 17 example

σ	a	b
a	0	2
b	1	0



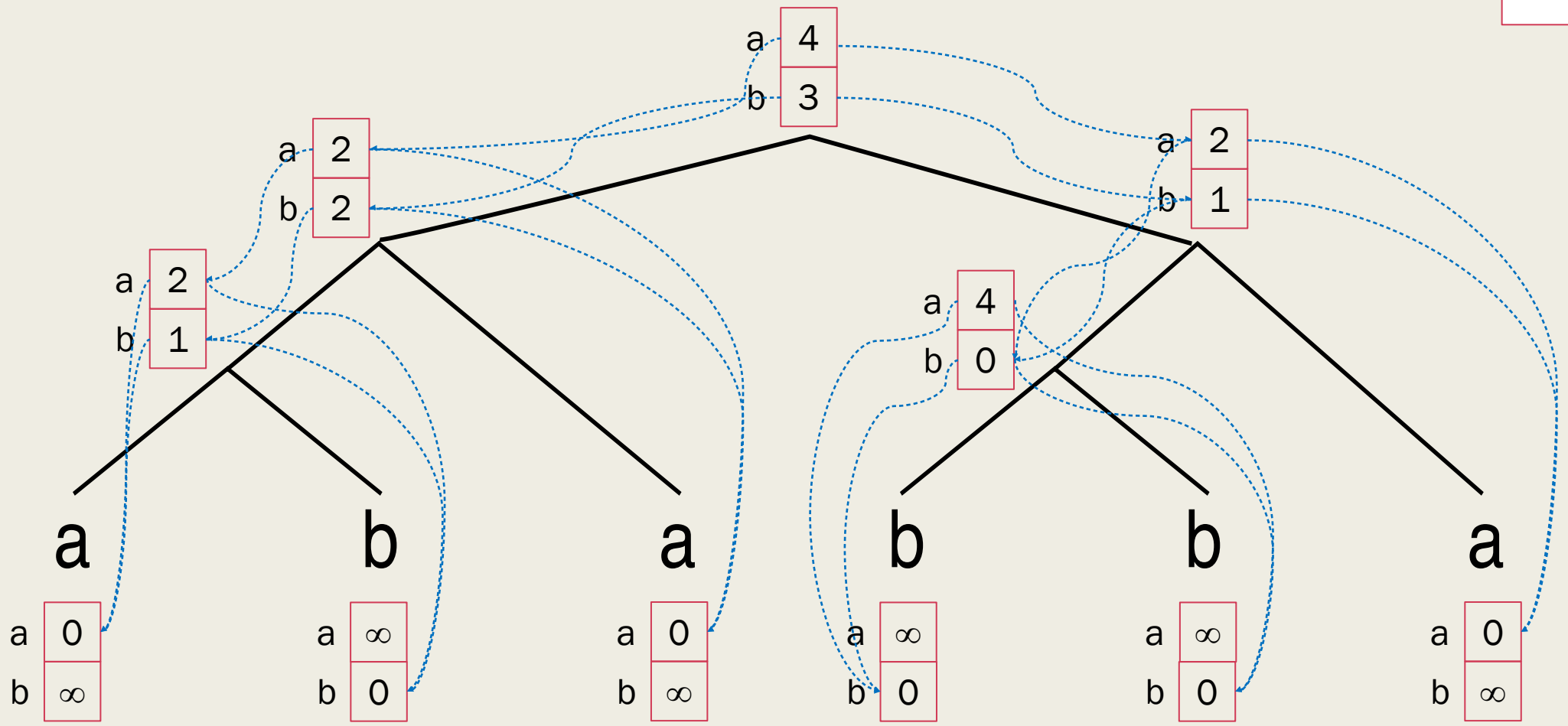
Handout 17 example

σ	a	b
a	0	2
b	1	0



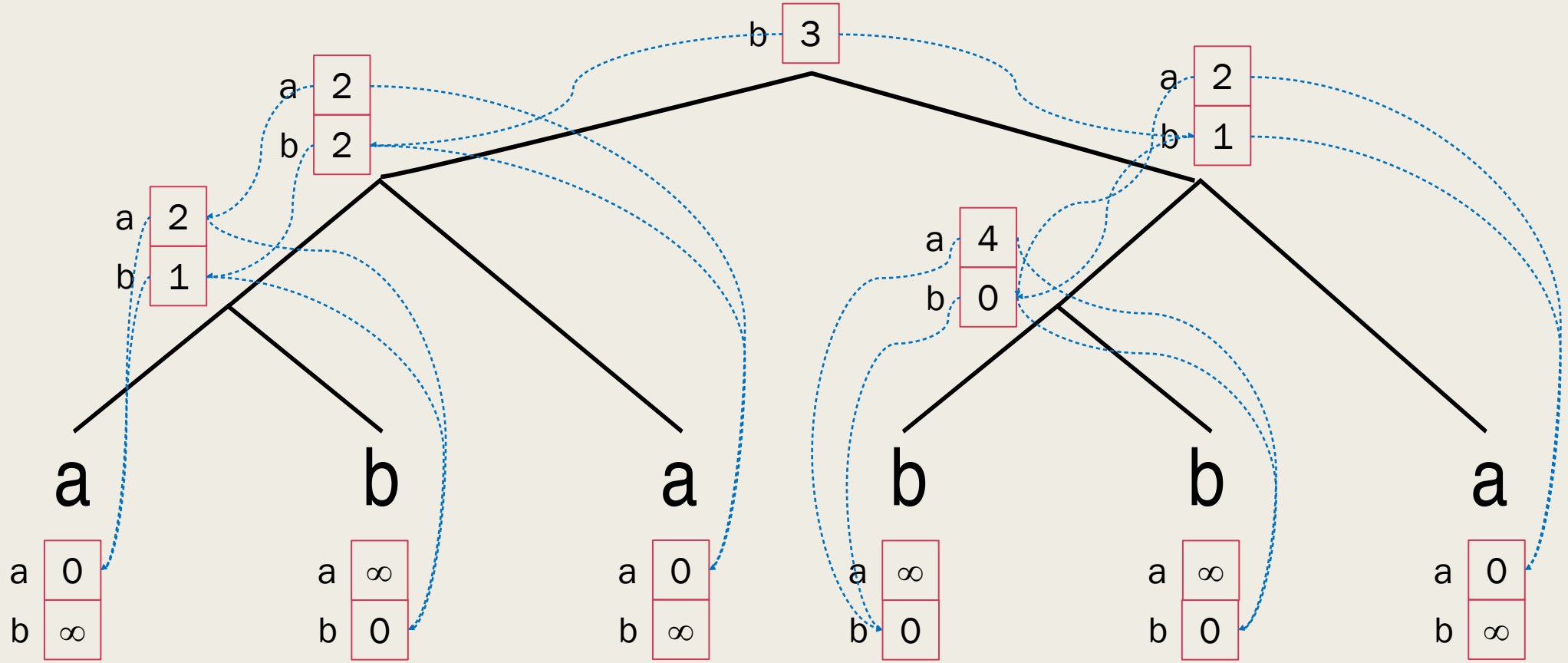
Handout 17 example

σ	a	b
a	0	2
b	1	0



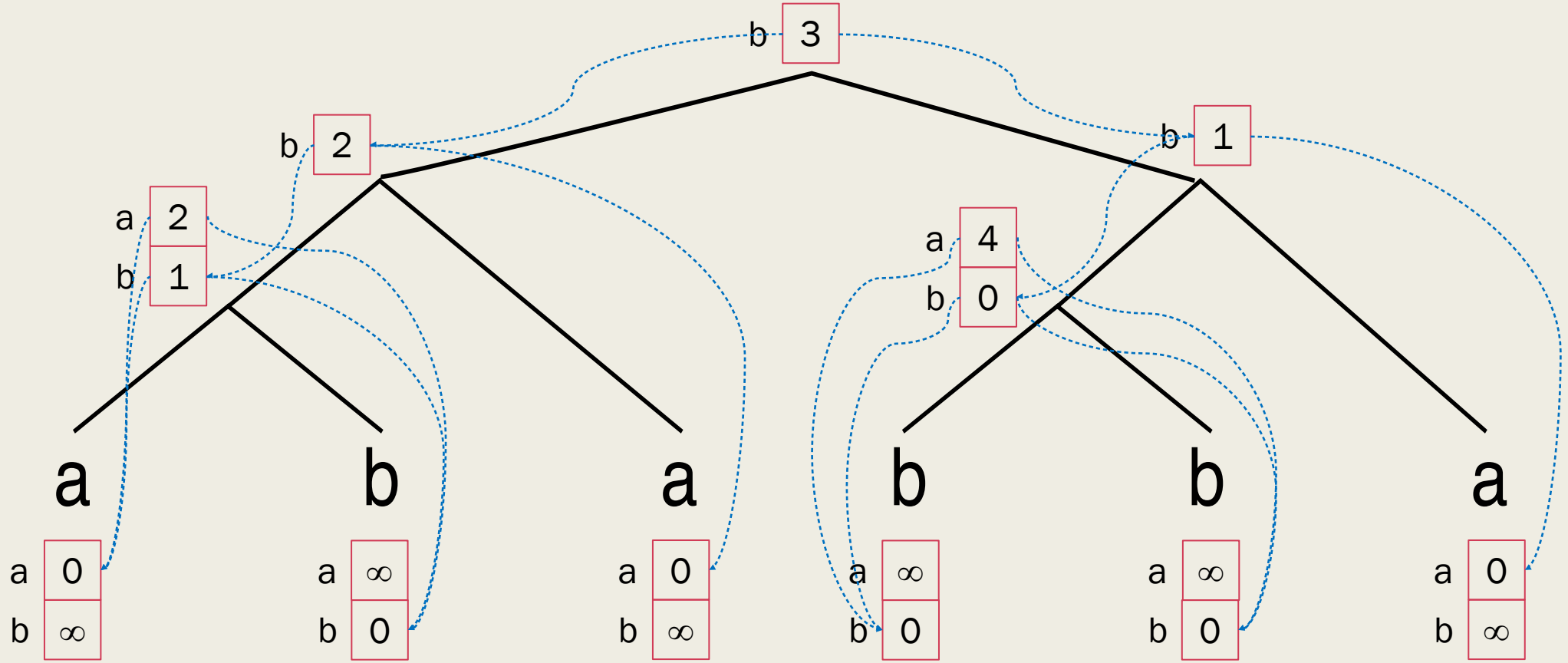
Handout 17 example

σ	a	b
a	0	2
b	1	0



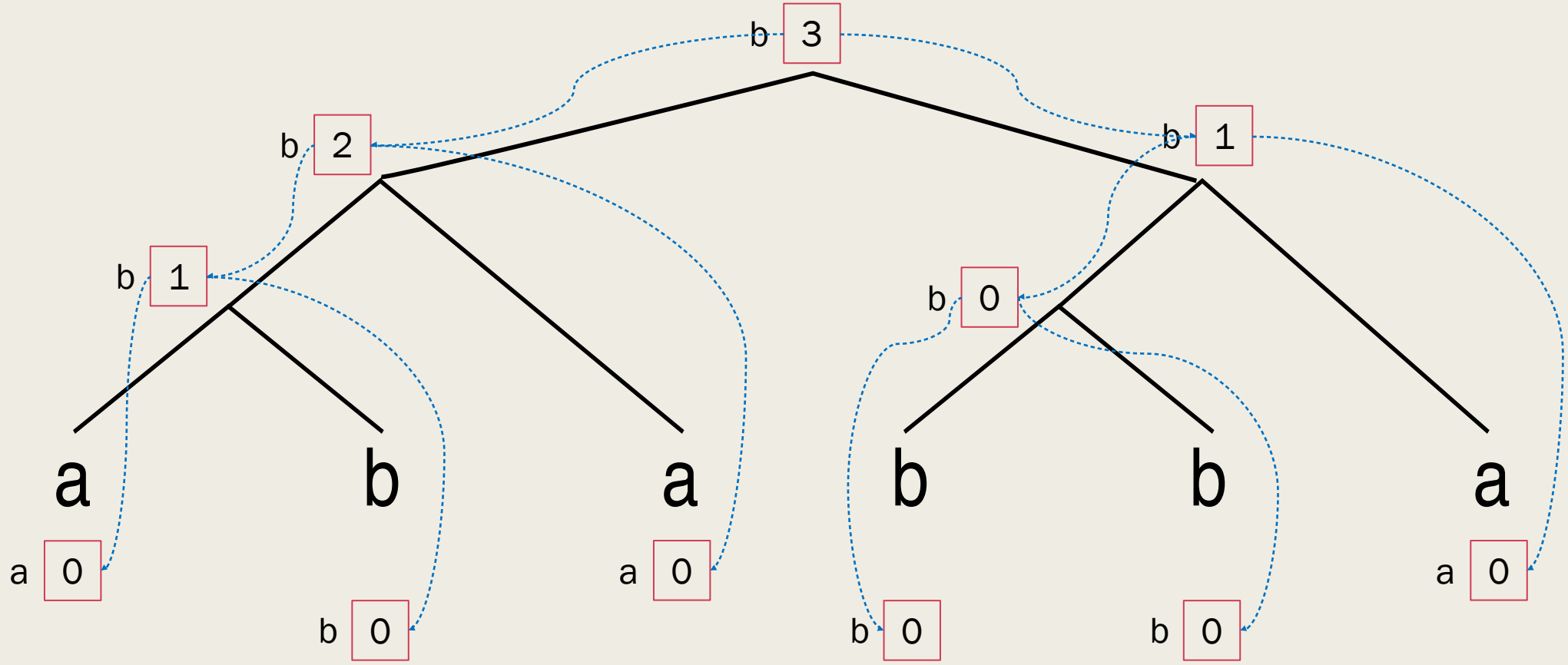
Handout 17 example

σ	a	b
a	0	2
b	1	0



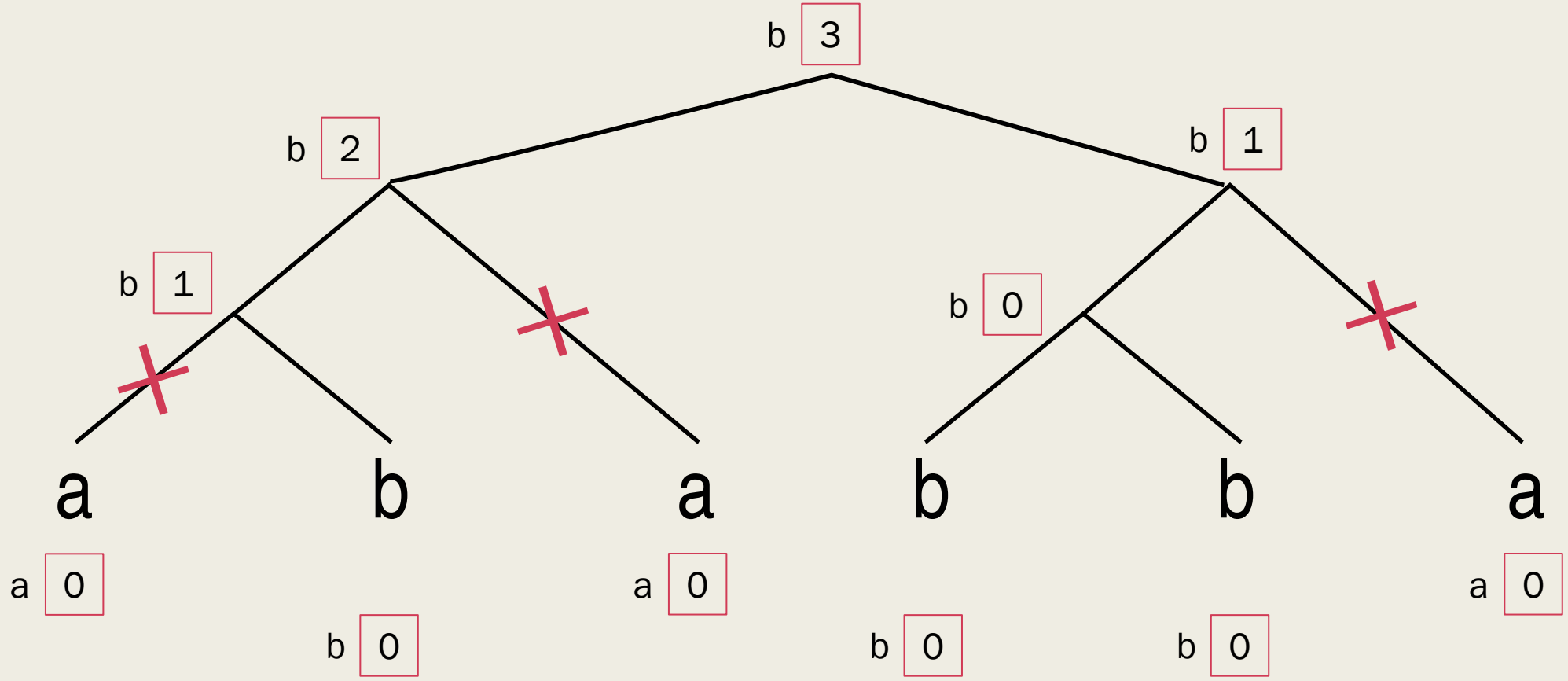
Handout 17 example

σ	a	b
a	0	2
b	1	0



Handout 17 example

σ	a	b
a	0	2
b	1	0



Runtime

- Let n=number of samples (sometimes called taxa)
- Let k=number of character states (i.e. 4 for DNA)
- What is the runtime of Fitch?
- What is the runtime of Sankoff?

Runtime

- Let n=number of samples (sometimes called taxa)
- Let k=number of character states (i.e. 4 for DNA)
- What is the runtime of Fitch?

$$O(nk)$$

- What is the runtime of Sankoff?

Runtime

- Let n=number of samples (sometimes called taxa)
- Let k=number of character states (i.e. 4 for DNA)
- What is the runtime of Fitch?

$$O(nk)$$

- What is the runtime of Sankoff?

$$O(nk^2)$$

Perfect Phylogeny

Introduction

- With Fitch and Sankoff we were only looking at a single site
- When we have multiple sites, an important question is whether or not there exists a phylogeny that is “consistent” with all the sites
- By consistent we often mean that a mutation at a given site only occurs once
- If we can construct a phylogeny where each mutation only occurs once, this is called a *perfect phylogeny*
- We will study one algorithm for constructing a perfect phylogeny (or getting close if one does not exist), called *Gusfield’s algorithm* (~1991)

Perfect phylogeny example

The Perfect Phylogeny Problem

David Fernández-Baca
Department of Computer Science and
Graduate Program in Bioinformatics and Computational Biology
Iowa State University, Ames, IA 50011
E-mail: `fernande@cs.iastate.edu`

	1	2	3	4	5	6
lamprey	0	0	0	0	0	1
shark	1	1	0	1	0	0
salmon	1	1	1	1	0	0
lizard	1	1	1	0	1	0

Figure 1: A data matrix and a phylogeny. The characters are: (a) paired fins, (b) jaws, (c) large dermal bones, (d) fin rays, (e) lungs, (f) rasping tongue.

Perfect phylogeny example

The Perfect Phylogeny Problem

David Fernández-Baca
Department of Computer Science and
Graduate Program in Bioinformatics and Computational Biology
Iowa State University, Ames, IA 50011
E-mail: `fernande@cs.iastate.edu`

	1	2	3	4	5	6
lamprey	0	0	0	0	0	1
shark	1	1	0	1	0	0
salmon	1	1	1	1	0	0
lizard	1	1	1	0	1	0

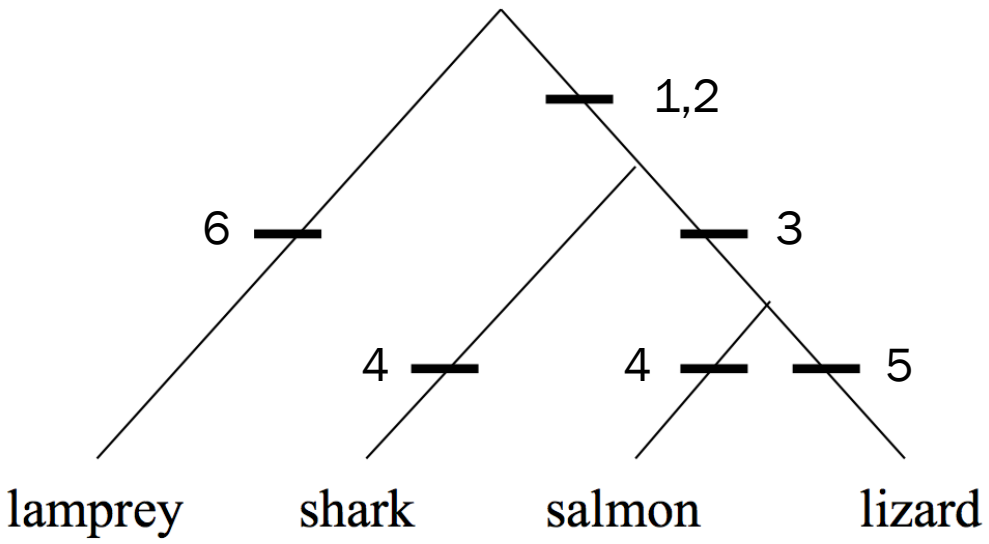


Figure 1: A data matrix and a phylogeny. The characters are: (a) paired fins, (b) jaws, (c) large dermal bones, (d) fin rays, (e) lungs, (f) rasping tongue.

Perfect phylogeny example

The Perfect Phylogeny Problem

David Fernández-Baca
Department of Computer Science and
Graduate Program in Bioinformatics and Computational Biology
Iowa State University, Ames, IA 50011
E-mail: fernande@cs.iastate.edu

	1	2	3	4	5	6
lamprey	0	0	0	0	0	1
shark	1	1	0	1	0	0
salmon	1	1	1	1	0	0
lizard	1	1	1	0	1	0

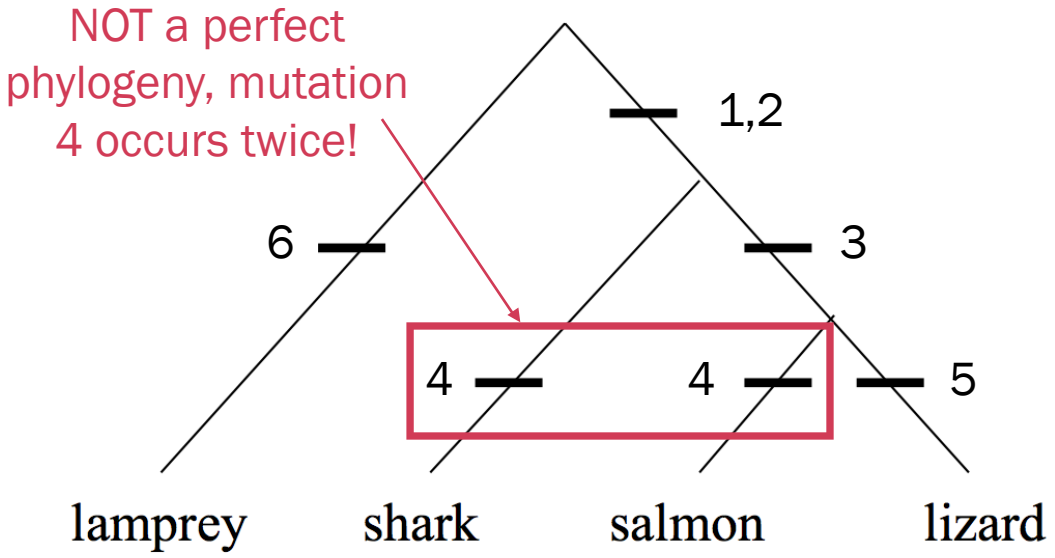


Figure 1: A data matrix and a phylogeny. The characters are: (a) paired fins, (b) jaws, (c) large dermal bones, (d) fin rays, (e) lungs, (f) rasping tongue.

Perfect Phylogeny example

A

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Camel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	0	0	0
Pig	0	0	0	?	0	0	0	0	?	0	0	0	?	?	0	0	?	1	1	1
Peccary	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	1	1
Chevrotain	?	0	?	?	?	?	?	?	?	1	0	?	?	?	1	1	0	?	0	0
Deer	0	0	0	0	0	0	0	1	?	1	1	1	1	1	1	?	1	1	0	0
Graffe	?	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0
Sheep	0	0	0	0	0	?	0	1	1	1	1	1	1	1	1	1	1	1	0	0
Cow	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0
Hippo	0	?	0	1	1	1	1	0	1	1	0	1	1	0	0	0	?	1	0	0
Humpback	1	1	1	1	1	1	1	0	1	1	0	1	1	0	0	0	?	?	0	0
Beaked	1	1	1	1	1	1	1	0	?	1	0	1	1	0	0	0	?	1	0	0

1	PM52	11	aaa792(Bov tA)
2	PM72	12	Gm5
3	M11	13	HIP5(CHR-1)
4	HIP24	14	HIP5(Bov A)
5	KM14	15	c21-352
6	HIP4	16	pgha
7	AF(CHR-1)	17	Fas
8	AF(MER)	18	INO
9	aaa228	19	pgi
10	aaa792(CHR-1)	20	pro

Proc. Natl. Acad. Sci. USA
Vol. 96, pp. 10261–10266, August 1999
Evolution

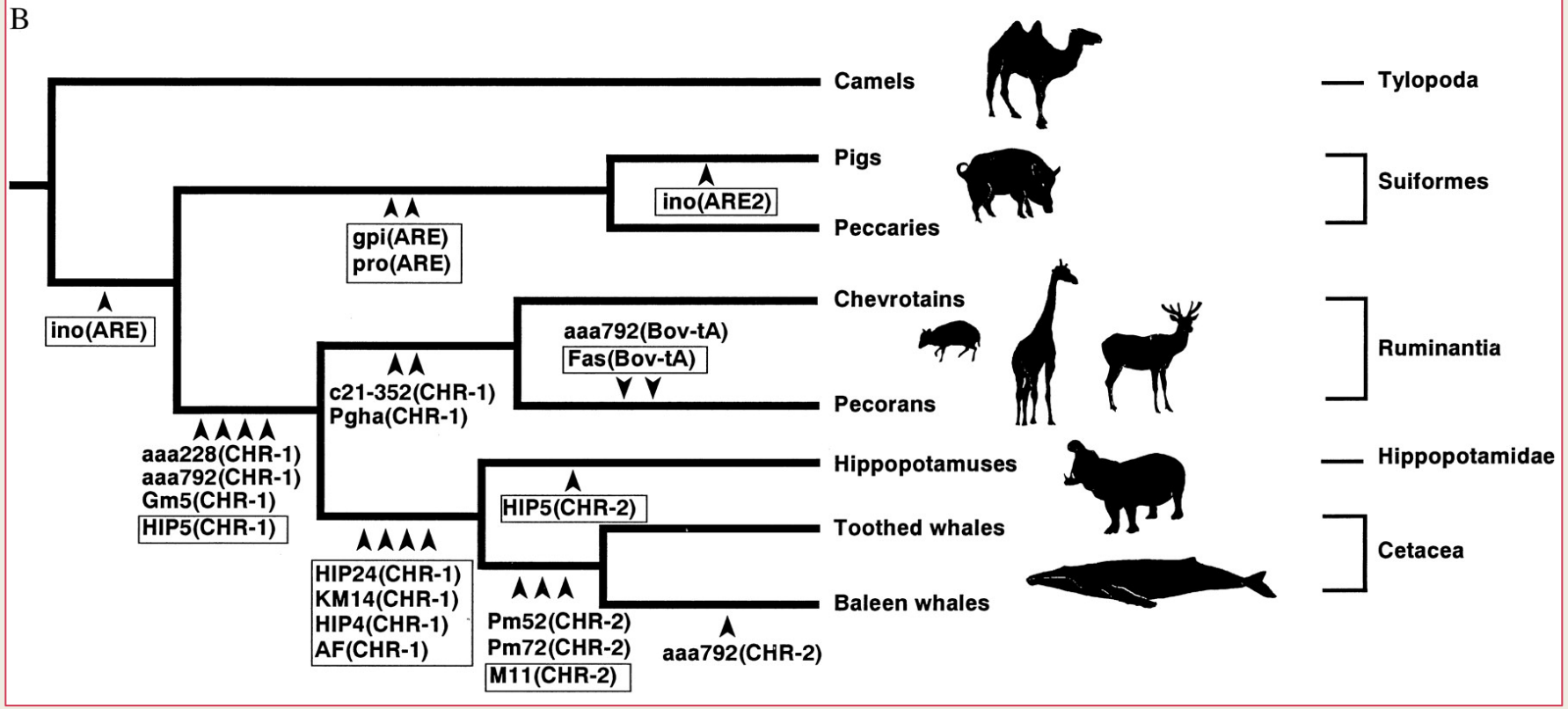
Phylogenetic relationships among cetartiodactyls based on insertions of short and long interpersed elements: Hippopotamuses are the closest extant relatives of whales

MASATO NIKAIDO[†], ALEJANDRO P. ROONEY[‡], AND NORIHIRO OKADA[†§]

[†]Faculty of Bioscience and Biotechnology, Tokyo Institute of Technology, 4259 Nagatsuta-cho, Yokohama, Midori-ku, Kanagawa 226-8501, Japan; and [‡]Institute of Molecular Evolutionary Genetics, Pennsylvania State University, 328 Mueller Laboratory, University Park, PA 16802

Communicated by Masatoshi Nei, Pennsylvania State University, University Park, PA, June 16, 1999 (received for review January 4, 1999)

B



Lab A

	1	2	3	4	5
A	1	1	0	0	0
B	0	0	1	0	0
C	1	1	0	0	1
D	0	0	1	1	0
E	0	1	0	0	0

$O_i \supset O_j \rightarrow$

	i	j
a	1	1
b	0	0
c	1	1
d	1	0

let O_i = set of samples with 1 in col i

$O_i = \{a, c, d\}$

$O_j = \{a, c\}$

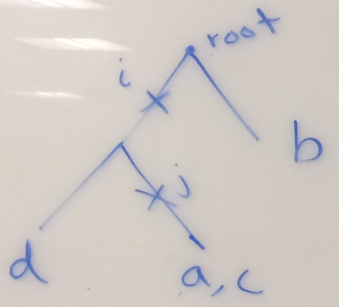
<u>Thm</u>: perfect phylogeny $\iff \forall\ i, j$

- $O_i \cap O_j = \emptyset$ (disjoint)

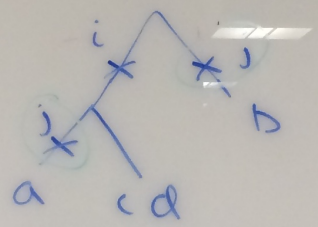
<u>or</u>

- $O_i \subset O_j$ or $O_j \subset O_i$ (containment)

- if $O_i \supset O_j$, then i happened before j.



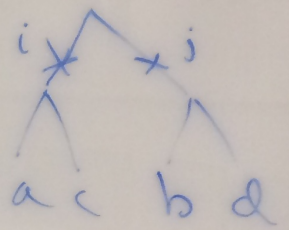
	i	j
a	1	1
b	0	1
c	1	0
d	1	0



not a perfect phylogeny

- if column i > column j as binary #'s, then either $O_i \supset O_j$ or $O_i \cap O_j = \emptyset$ (disjoint)

	i	j
a	1	0
b	0	1
c	1	0
d	0	1



Lab B

	1	2	3	4	5
A	1	1	0	0	0
B	0	0	1	0	0
C	1	1	0	0	1
D	0	0	1	1	0
E	0	1	0	0	0

$m \rightarrow$

$O_i \supset O_j$

	i	j
a	1	1
b	0	0
c	1	1
d	1	0

O_i = set of samples with mutation i

$O_i = \{a, c, d\}$

$O_j = \{a, c\}$

Thm: $\exists$ perfect phylogeny $\iff$

$\forall\, i, j$ either:

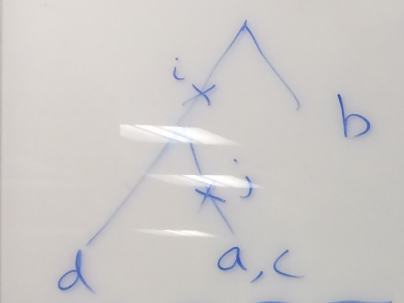
- O_i & O_j are disjoint

  $O_i \cap O_j = \emptyset$

or

- one contains the other

  $O_i \subset O_j$ or $O_j \subset O_i$



① if $O_i > O_j$ then i happened more ancienty than j

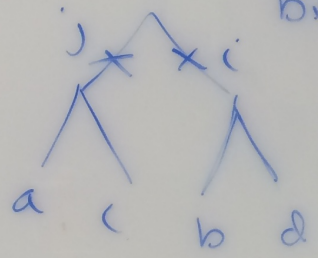
② if column i > column j as binary numbers, then:

- $O_i > O_j$

or

- $O_i \cap O_j = \emptyset$

	i	j
a	0	1
b	1	0
c	0	1
d	1	0



	i	j
a	1	0
b	1	0
c	0	1
d	1	0