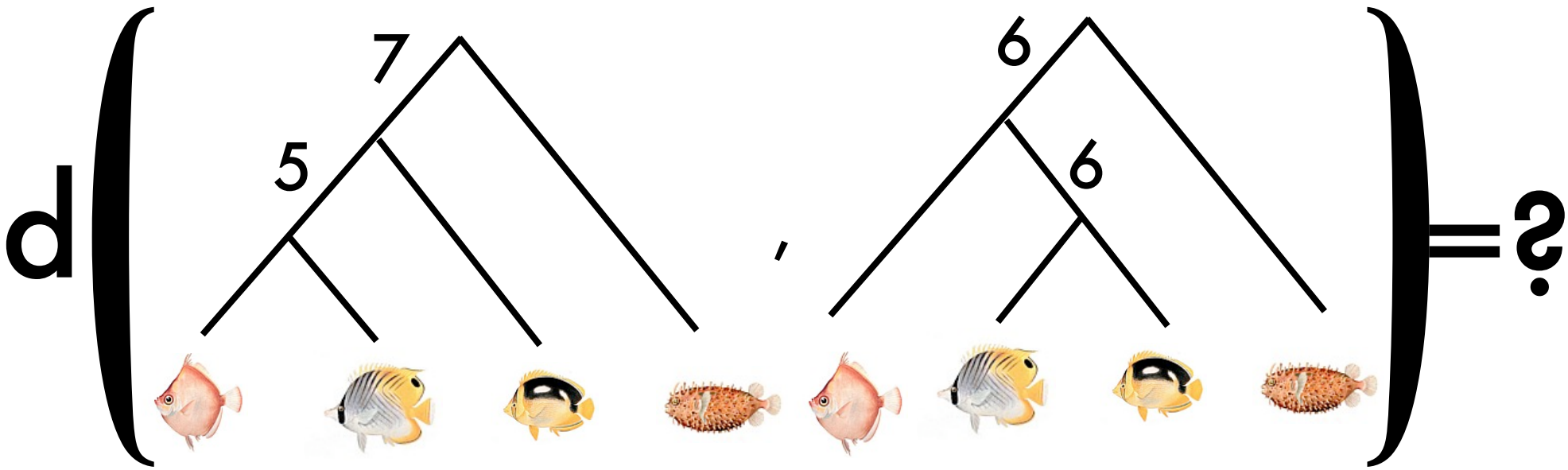
Computing Geodesic Distances in Tree Space in Polynomial Time

Megan Owen
SAMSI/NCSU

Scott Provan
UNC

Problem



- geodesic distance introduced by Billera, Holmes, and Vogtmann in "The geometry of the space of phylogenetic trees," 2001

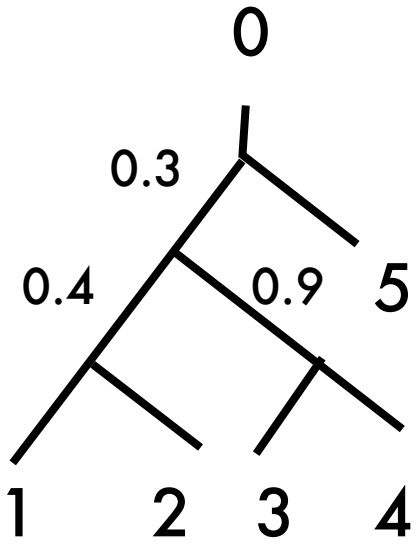
Motivation

- biologists often compare phylogenetic trees on the same set of species:
  - different genes give different trees
  - to test algorithms for constructing phylogenetic trees
- want to compute "average" trees and do statistics on trees
- phylogenetic trees outside of biology

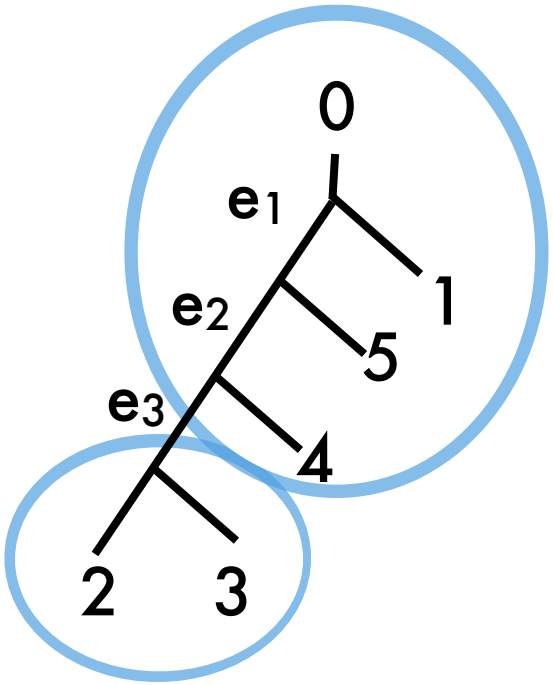
Tree Space $\mathbb{T}_n$

- $\mathbb{T}_n$ = space containing all rooted semi-labeled binary trees with n leaves and interior branch lengths ≥0
- $\mathbb{T}_n$ also contains

  degenerate trees



What is an edge?



- an interior edge partitions the set of leaves into 2:

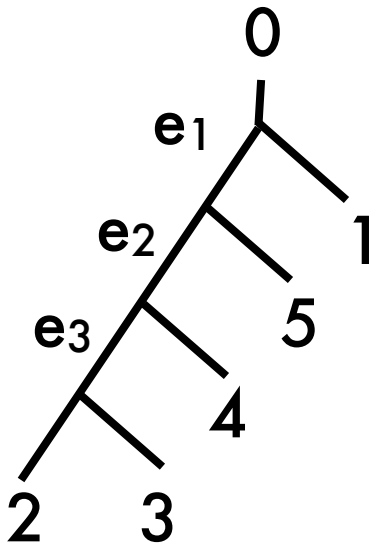
e_3 = 23 | 0145

Edge Compatibility

- $e_x = X|X'$ is *compatible* with $e_y = Y|Y'$ if both can exist in the same tree;

  more formally, if one of $X \cap Y$, $X \cap Y'$,

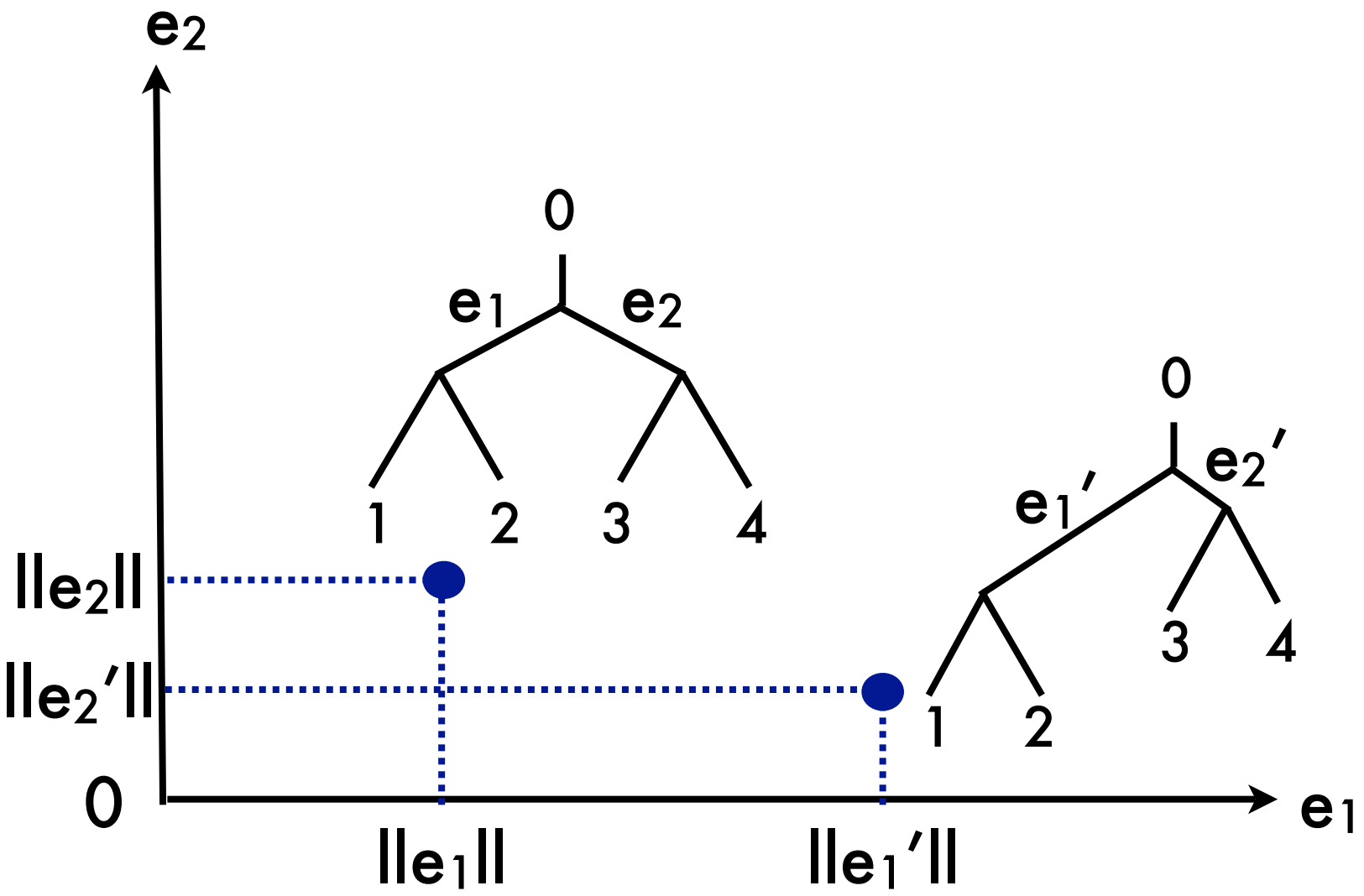
  $X' \cap Y$, or $X' \cap Y'$ is empty



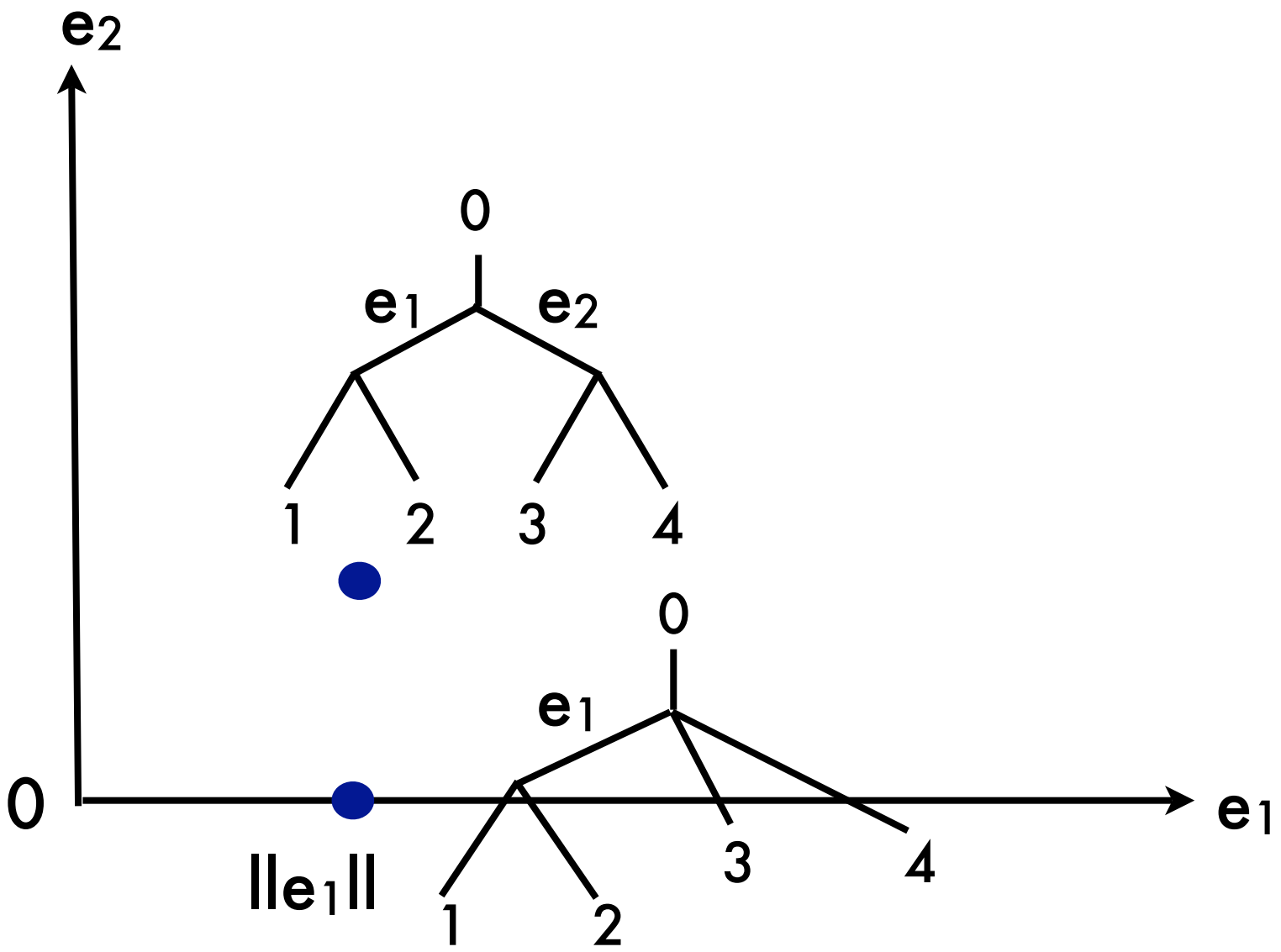
ex. e_3 = 23 | 0145 is compatible with e_2 = 234 | 015

but not with f = 12 | 0345

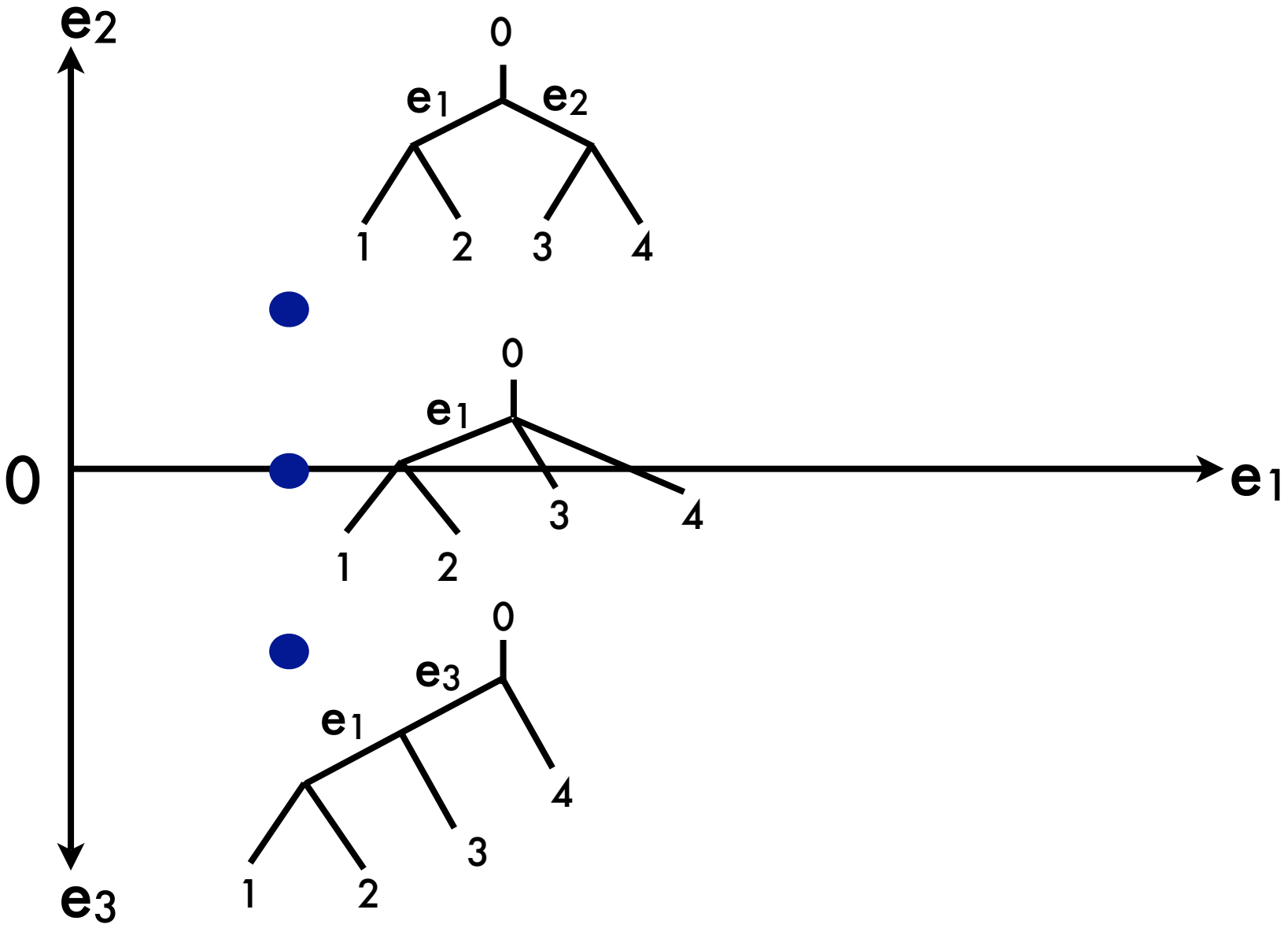
Orthants



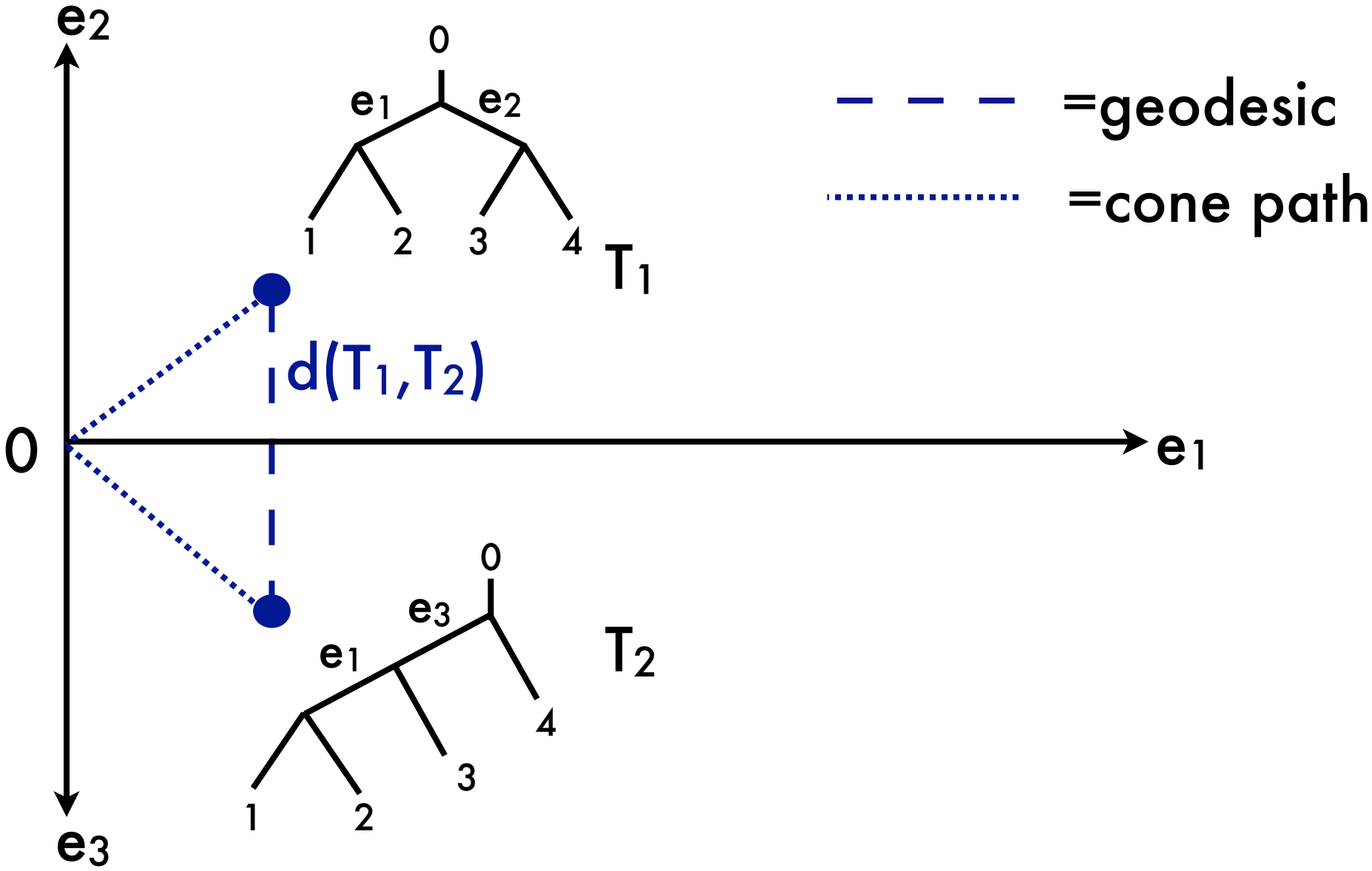
Orthants



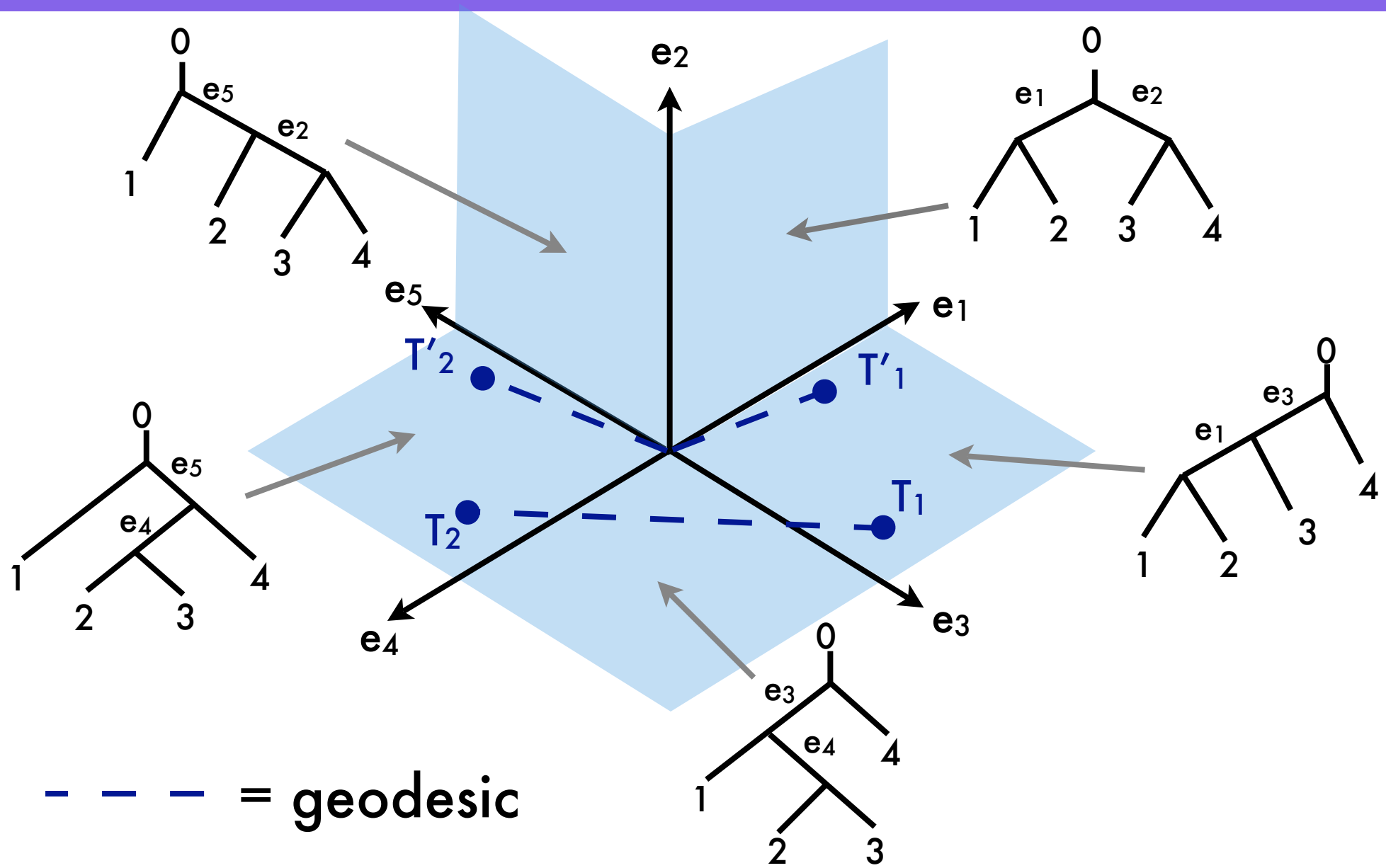
Orthants



Geodesic Distance



Structure of $\mathbb{T}_4$



Properties of $\mathbb{T}_n$

- CAT(0) space (non-positively curved)

  $\Rightarrow$ unique geodesic

  $\Rightarrow$ well-defined mid-point tree

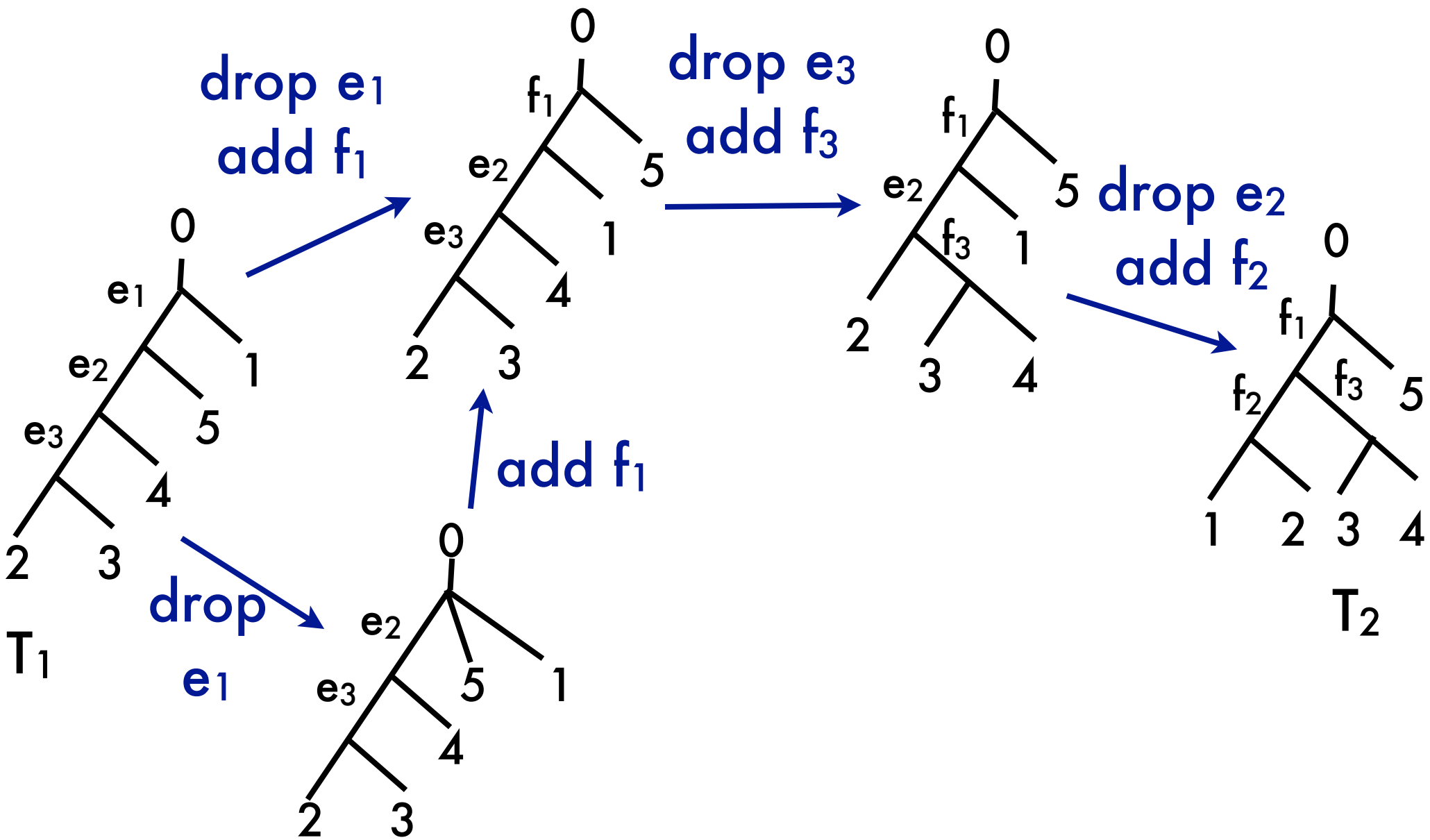
- geodesic = shortest path between two points

Question

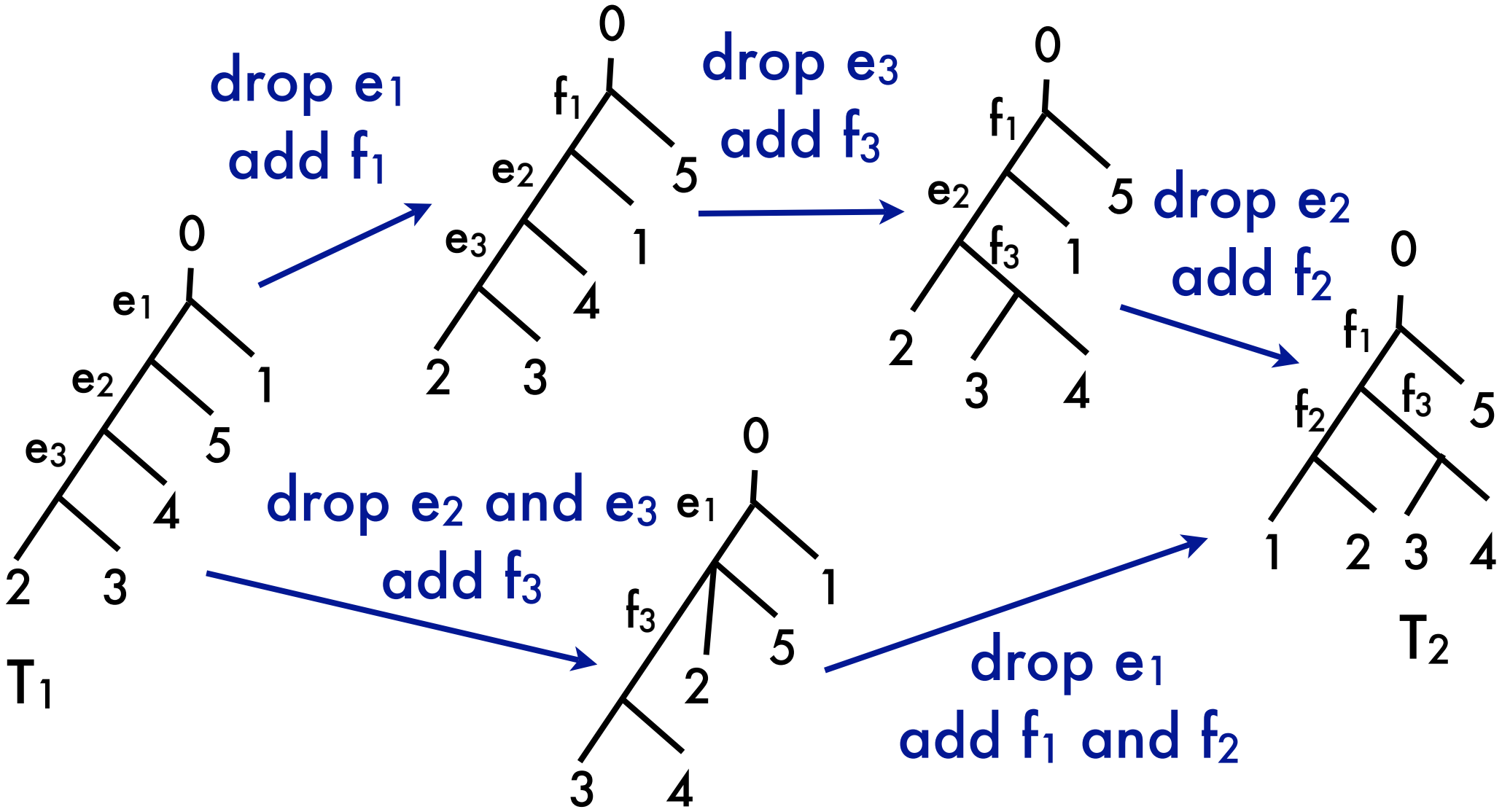
- Can we find an efficient algorithm to compute the geodesic between two trees, T_1 and T_2, in $\mathbb{T}_n$?

- two exponential algorithms:
  - GeoMeTree (Kupczok et al., 2008)
  - GeodeMaps (2009)

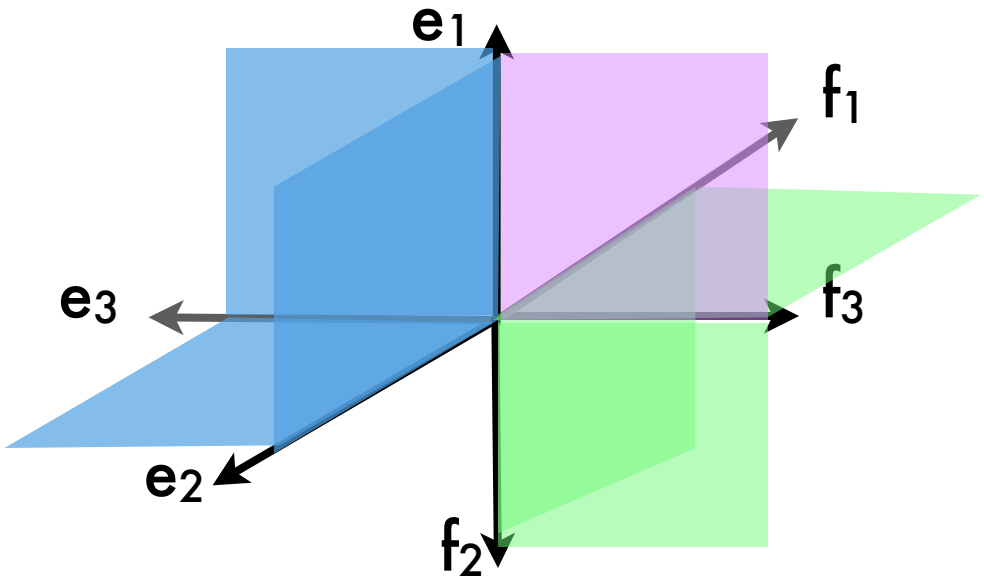
Path Spaces

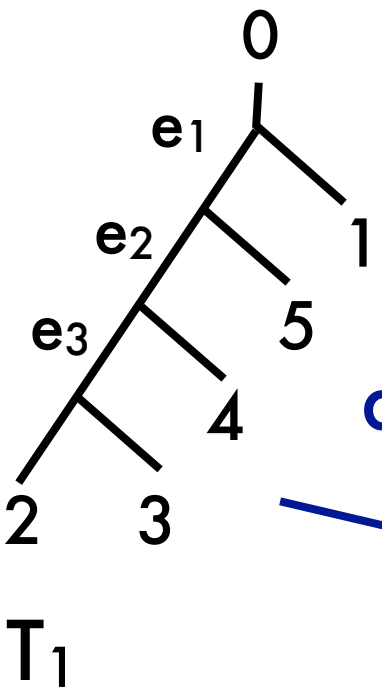


Path Spaces


drop e_1
add f_1
drop e_3
add f_3
drop e_2
add f_2
drop e_2 and e_3
add f_3
drop e_1
add f_1 and f_2
T_1
T_2

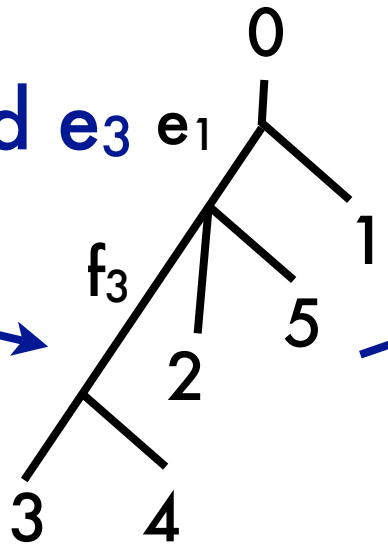
Path Spaces



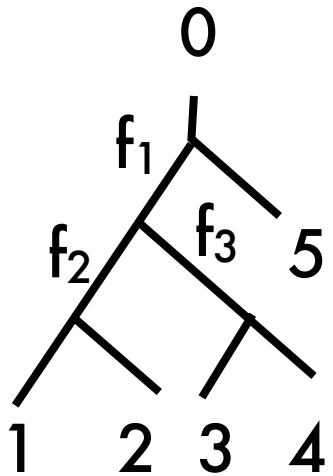


T1

drop e_2 and e_3
add f_3



drop e_1
add f_1 and f_2



T2

Characterizing Geodesics

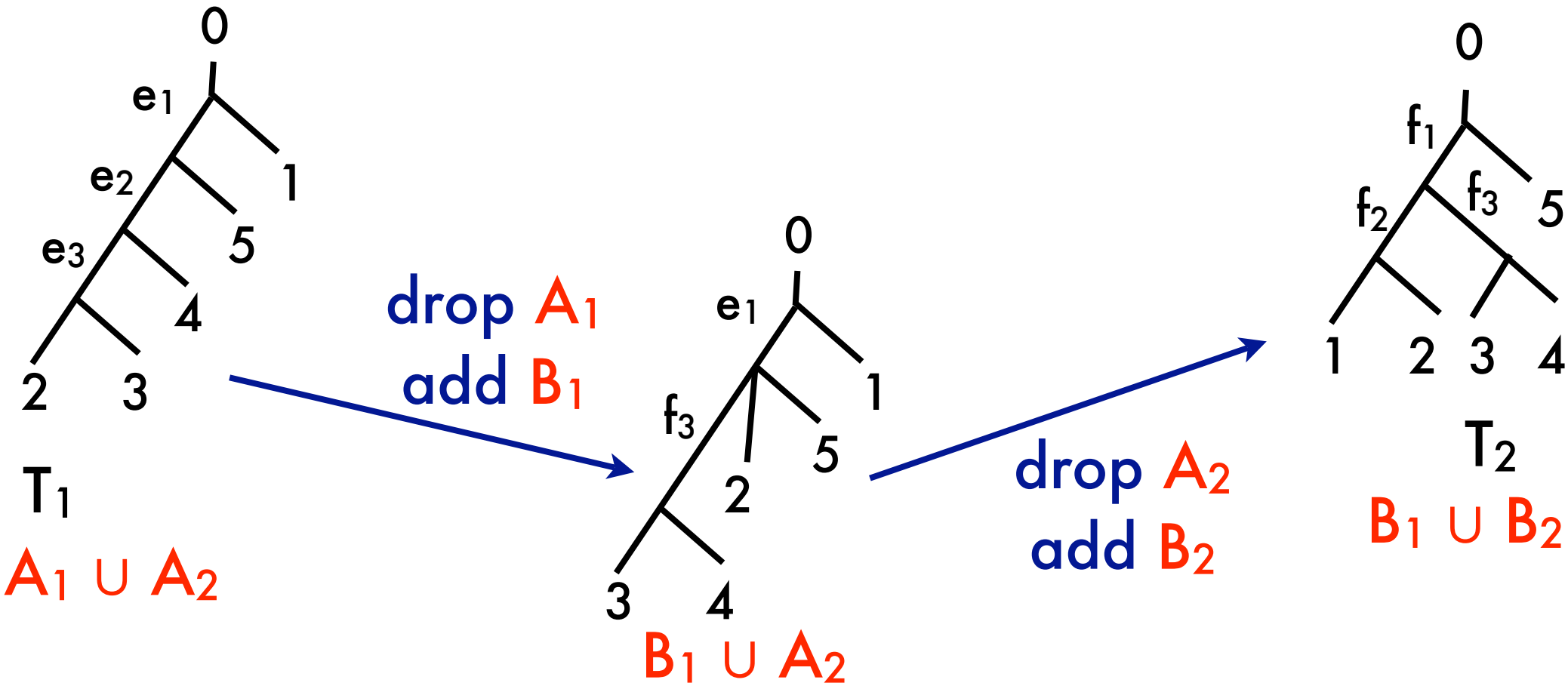
- at i^{th} transition between orthants:
  - edges A_i are dropped
  - edges B_i are added
- $(A_1, \ldots, A_k)$ partitions $E(T_1)$ and $(B_1, \ldots, B_k)$ partitions $E(T_2)$
- geodesic characterized by 3 properties
- Property 1:

  A_i and B_j compatible for all $i > j$

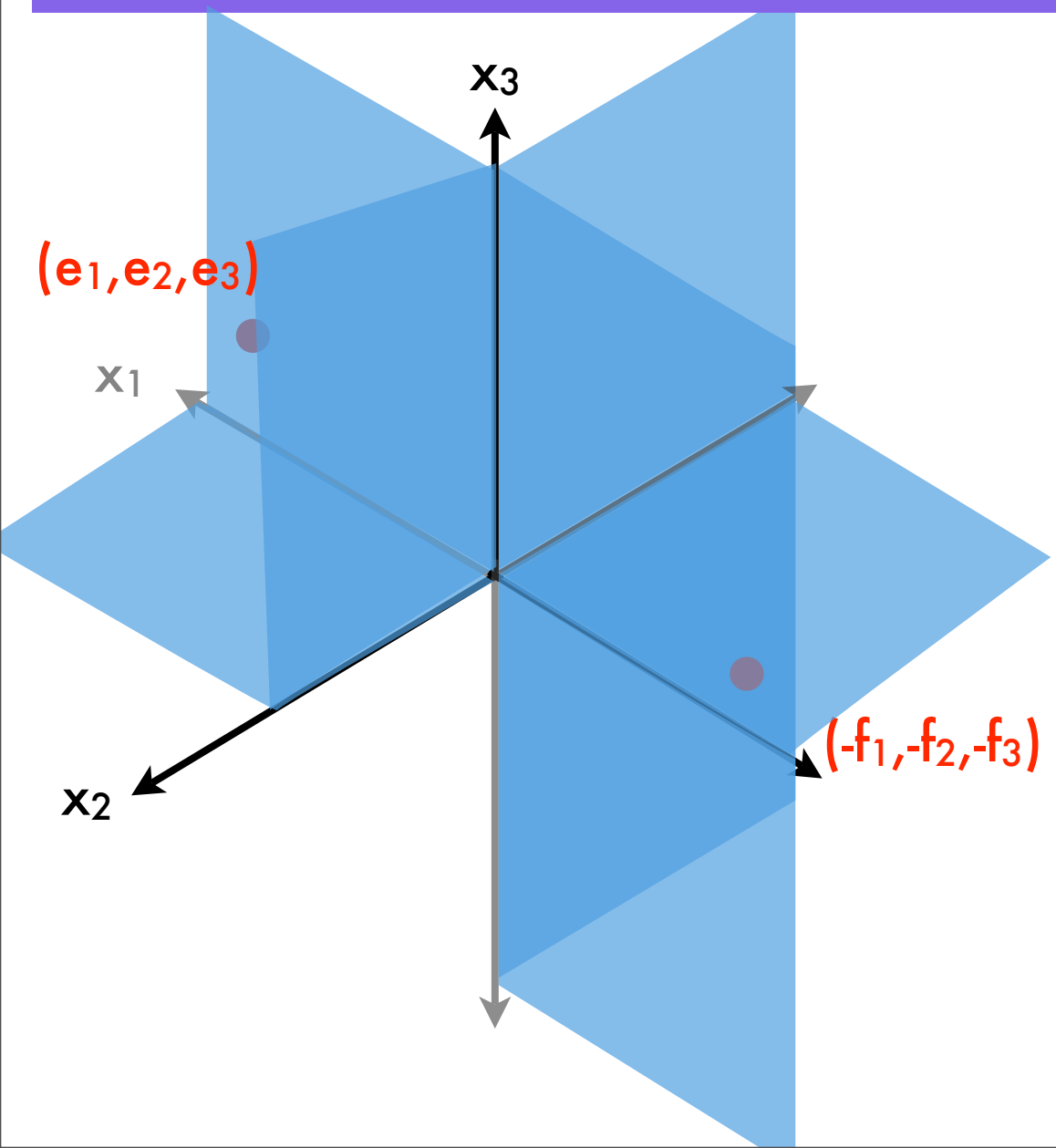
Property 1

- Property 1:

  A_i and B_j compatible for all $i > j$



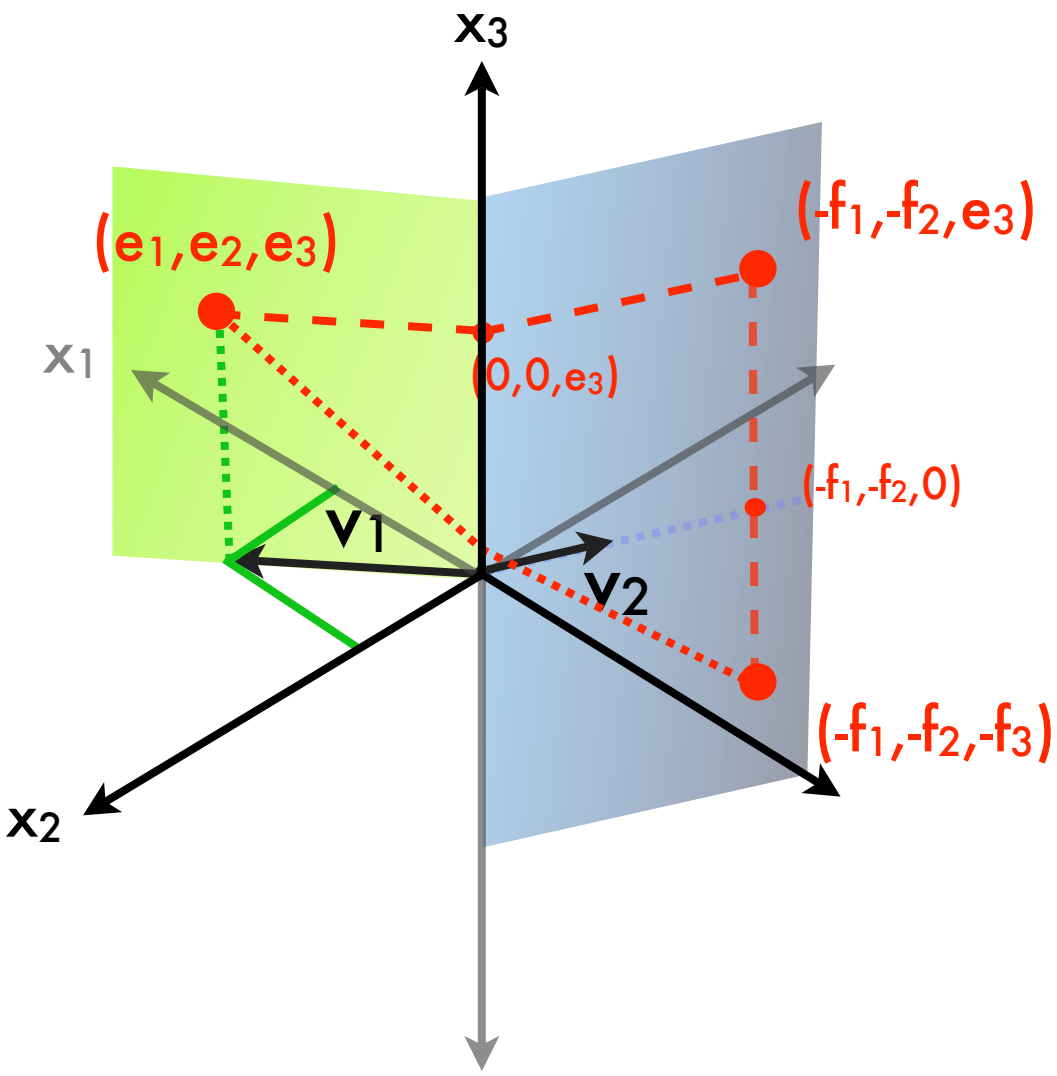
Property 2

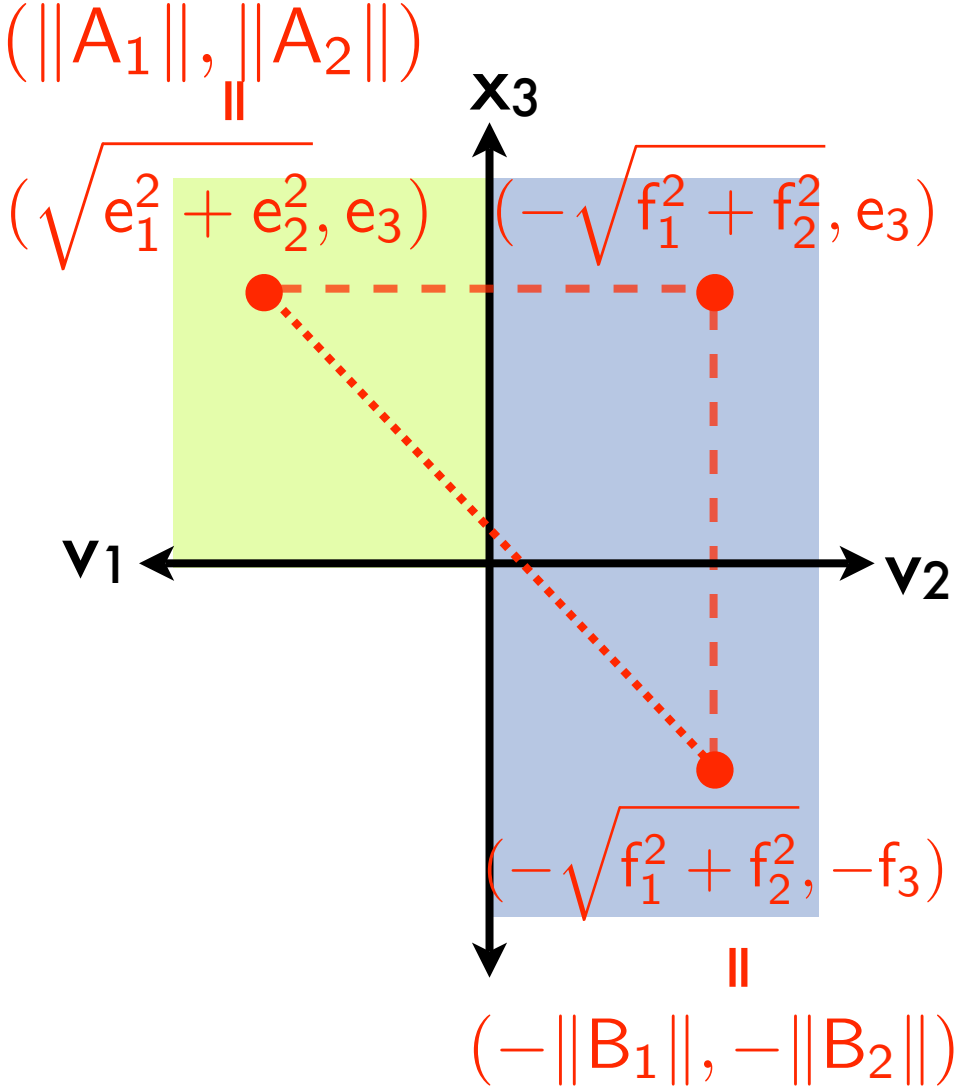


$A_1 = \{ e_1, e_2 \}$
$B_1 = \{ f_1, f_2 \}$

$A_2 = \{ e_3 \}$
$B_2 = \{ f_3 \}$

Isometric to part of $\mathbb{R}^k$





..... = geodesic

Property 2

- line from $(\|A_1\|,\ldots,\|A_k\|)$ to $(-\|B_1\|, \ldots, -\|B_k\|)$ is the geodesic in our region of $\mathbb{R}^k$ iff

$$\frac{||A_1||}{||B_1||} \leq \frac{||A_2||}{||B_2||} \leq \ldots \leq \frac{||A_k||}{||B_k||}$$

$\Rightarrow$ geodesic distance = Euclidean distance

$$= \sqrt{\sum_{i=1}^{k} \|A_i\| + \|B_i\|}$$

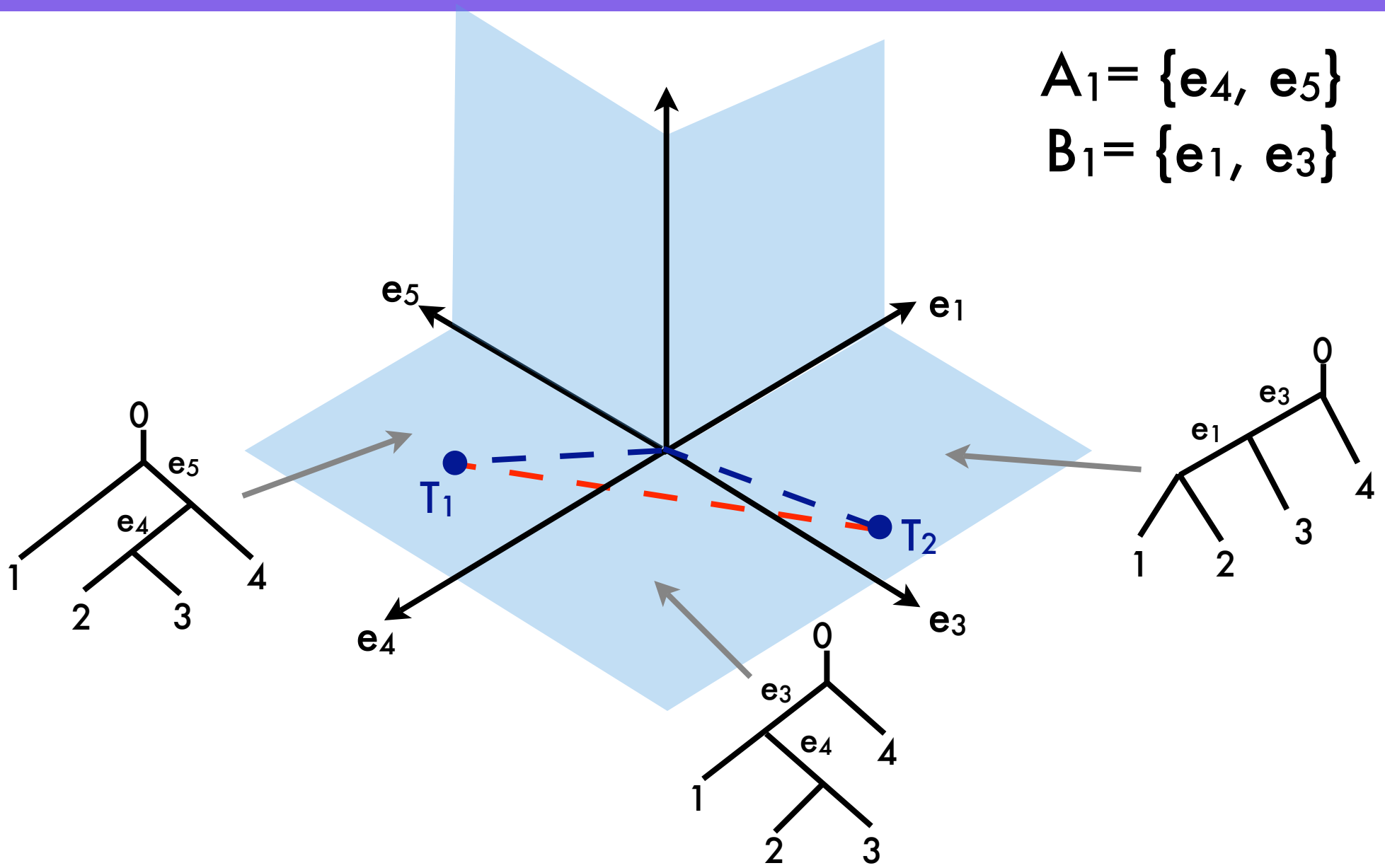
- Property 2:

$$\frac{||A_1||}{||B_1||} \leq \frac{||A_2||}{||B_2||} \leq \ldots \leq \frac{||A_k||}{||B_k||}$$

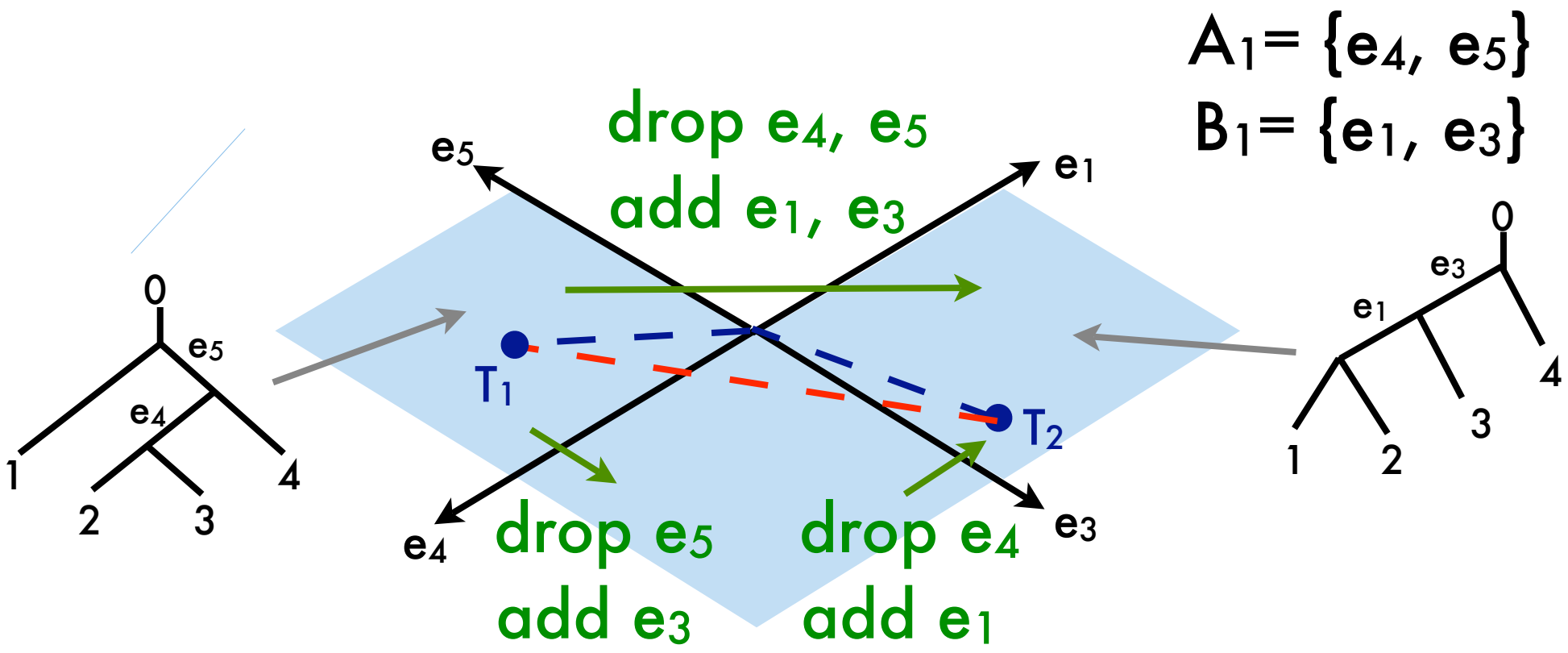
Property 3

$A_1 = \{e_4, e_5\}$

$B_1 = \{e_1, e_3\}$



Property 3

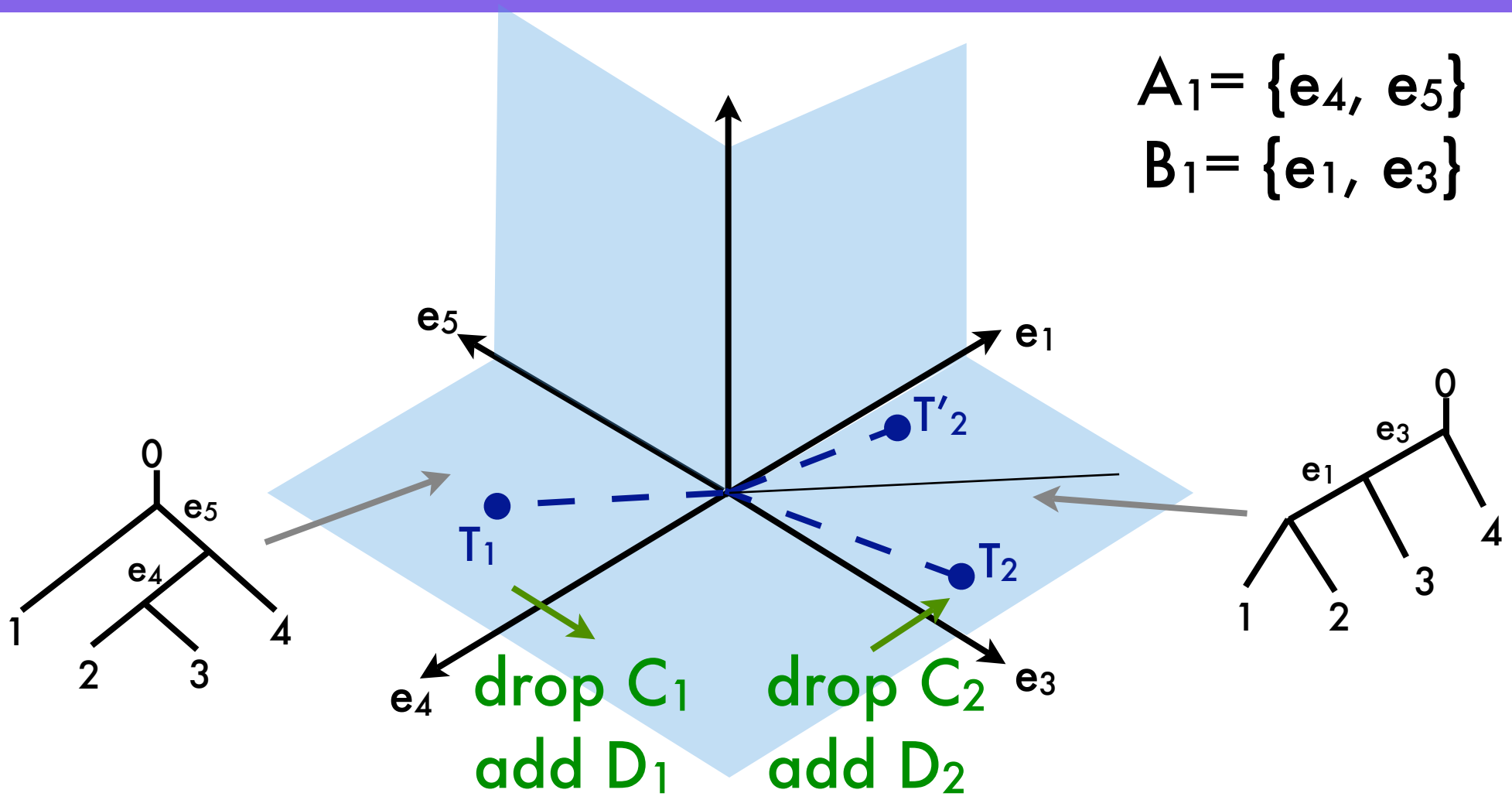


For (A_1, B_1), $\exists$ partition $C_1 \cup C_2$ of A_1, and partition $D_1 \cup D_2$ of B_i, such that C_2 is compatible with D_1.

Property 3

$A_1 = \{e_4, e_5\}$

$B_1 = \{e_1, e_3\}$



Want $\frac{\|C_1\|}{\|D_1\|} < \frac{\|C_2\|}{\|D_2\|}$

Property 3

- Property 3:

  For each pair (A_i, B_i), $\nexists$ partition $C_1 \cup C_2$ of A_i, and partition $D_1 \cup D_2$ of B_i, such that C_2 is compatible with D_1 and $\frac{\|C_1\|}{\|D_1\|} < \frac{\|C_2\|}{\|D_2\|}$.

Theorem:

Partitions $(A_1, \ldots, A_k)$ and $(B_1, \ldots, B_k)$ represent the geodesic iff Properties 1, 2, and 3 hold.

Geodesic Algorithm

Initialize: $A_1 = E(T_1)$, $B_1 = E(T_2)$ (cone path)

P1 and P2 hold.

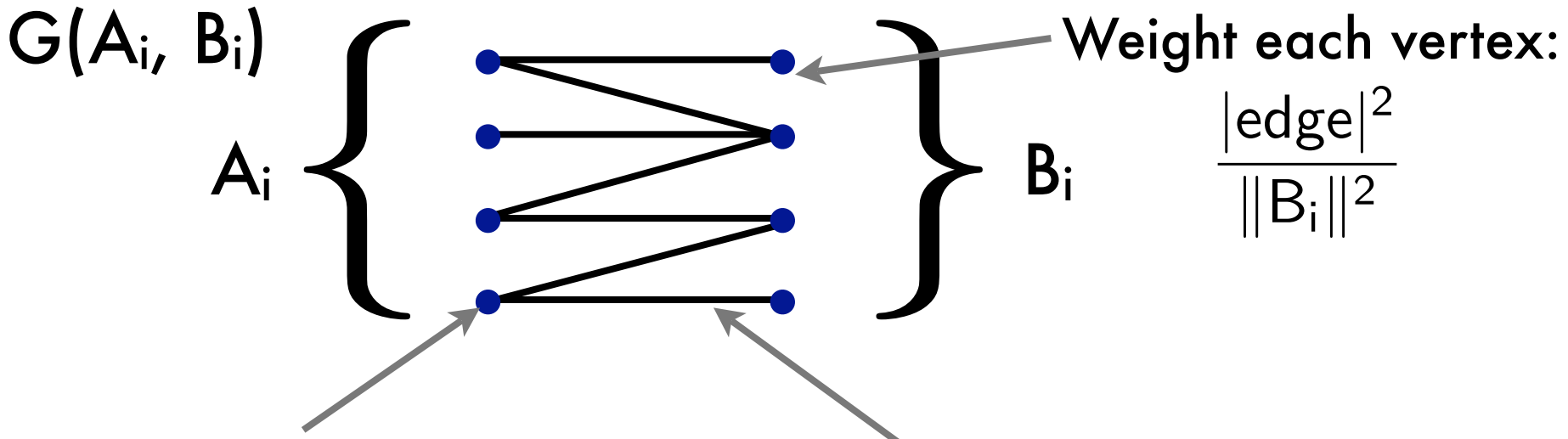
Iterative Step: P1 and P2 hold for $(A_1, \ldots, A_r)$ and $(B_1, \ldots, B_r)$.

Does (A_i, B_i) satisfies Property 3 for every i?

No: split blocks A_i and B_i, and re-index the new partition to get $(A_1, \ldots, A_{r+1})$ and $(B_1, \ldots, B_{r+1})$.

Yes: we are done.

Checking Property 3



Property 3 fails if there is a min. weight vertex cover of $G(A_i, B_i)$ with weight < 1.

Correctness

Lemma: At every step in the geodesic algorithm, Property 1 and 2 hold.

Complexity of checking Property 3: $O(n^3)$
(Solve as a max. flow problem.)

Complexity of the geodesic algorithm: $O(n^4)$

Future Work

- computing centre of mass in tree space
- grouping similar trees using Principal Component Analysis
- using the geodesic distance and tree space to do statistics on trees

Thank You

- NSF through SAMSI for funding

- collaborator Scott Provan