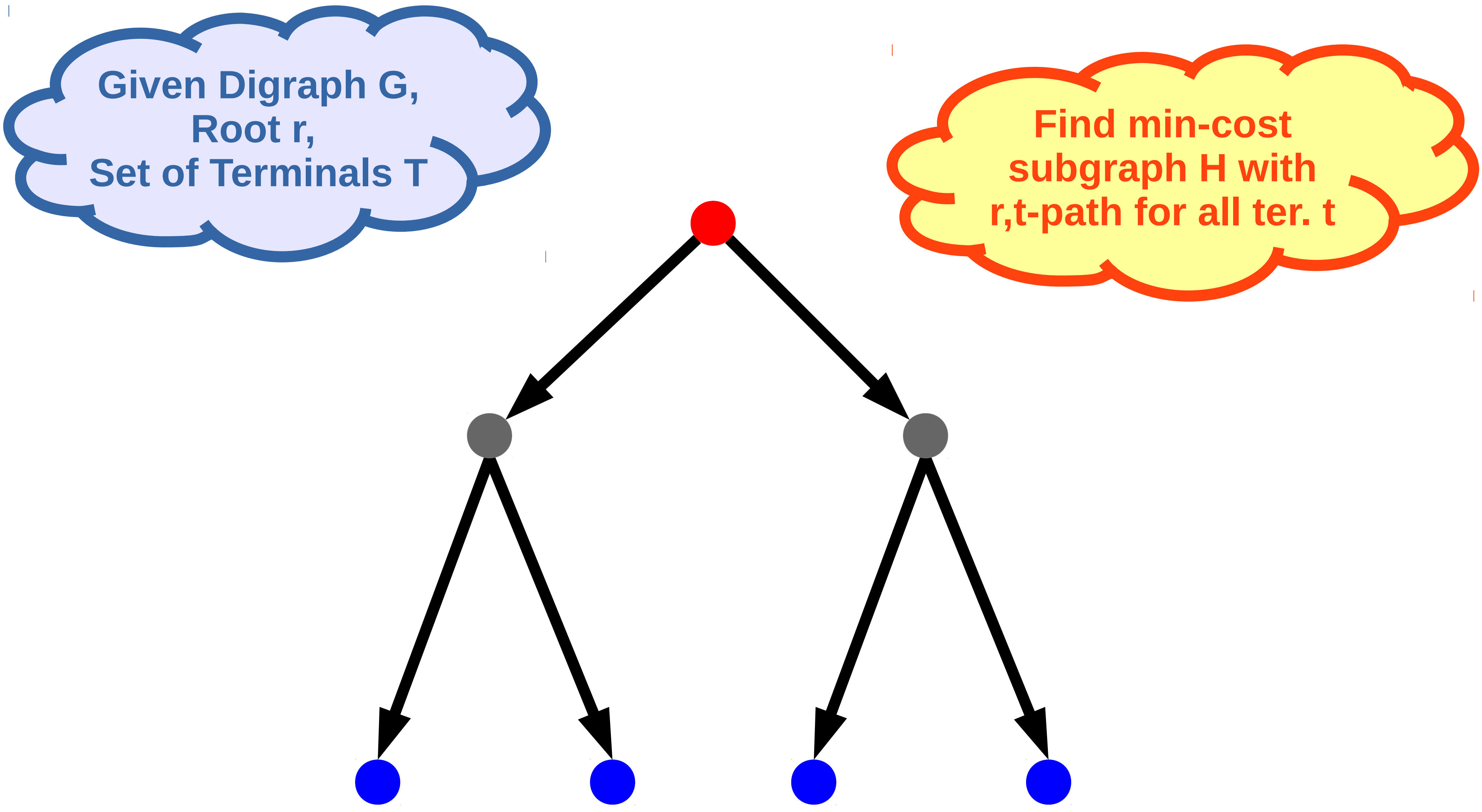
Approximating Directed Steiner Problems via Tree Embedding

Bundit Laekhanukit

Weizmann Institute of Science
Rehovot, Israel

We study a natural generalization of Directed Steiner Tree.

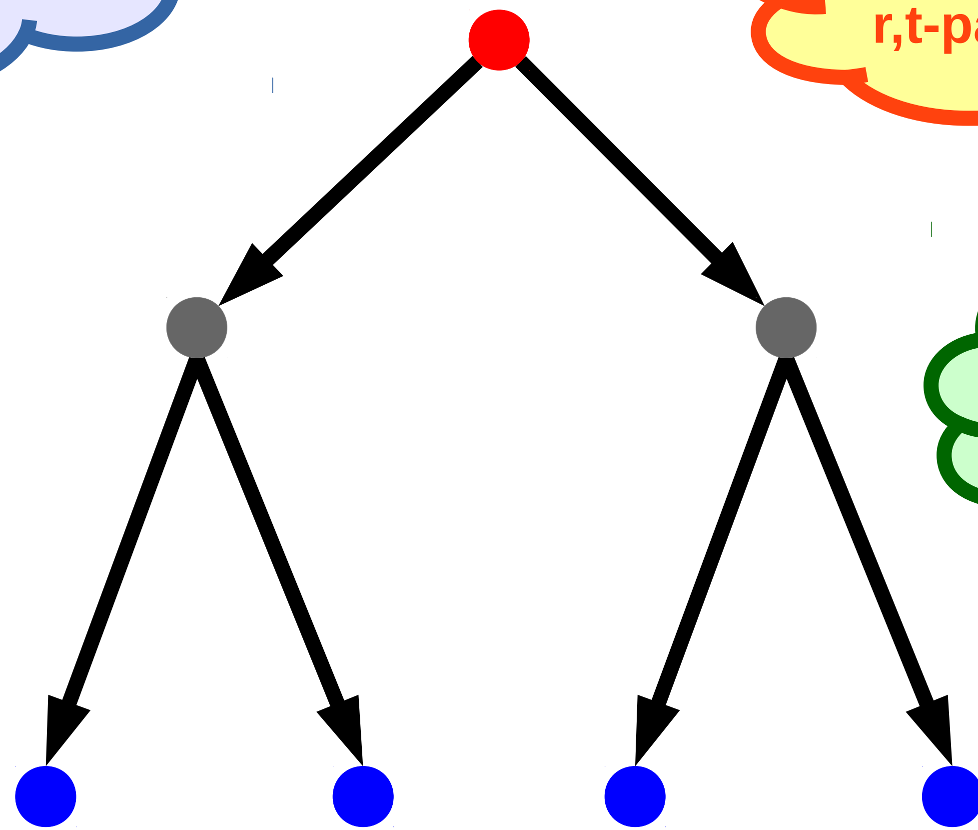
Directed Steiner Tree (DST)



Directed Steiner Tree (DST)

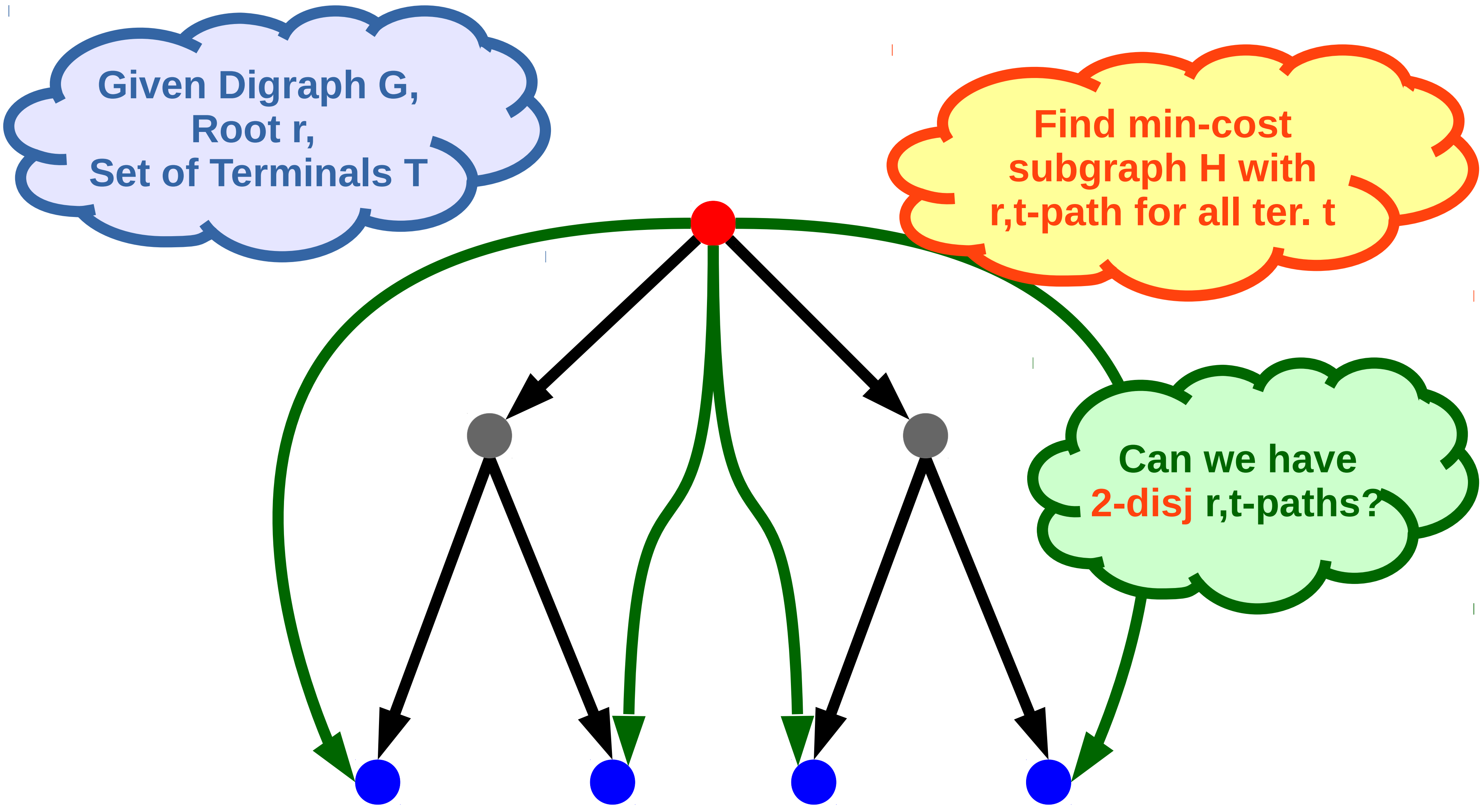
Given Digraph G,
Root r,
Set of Terminals T

Find min-cost
subgraph H with
r,t-path for all ter. t

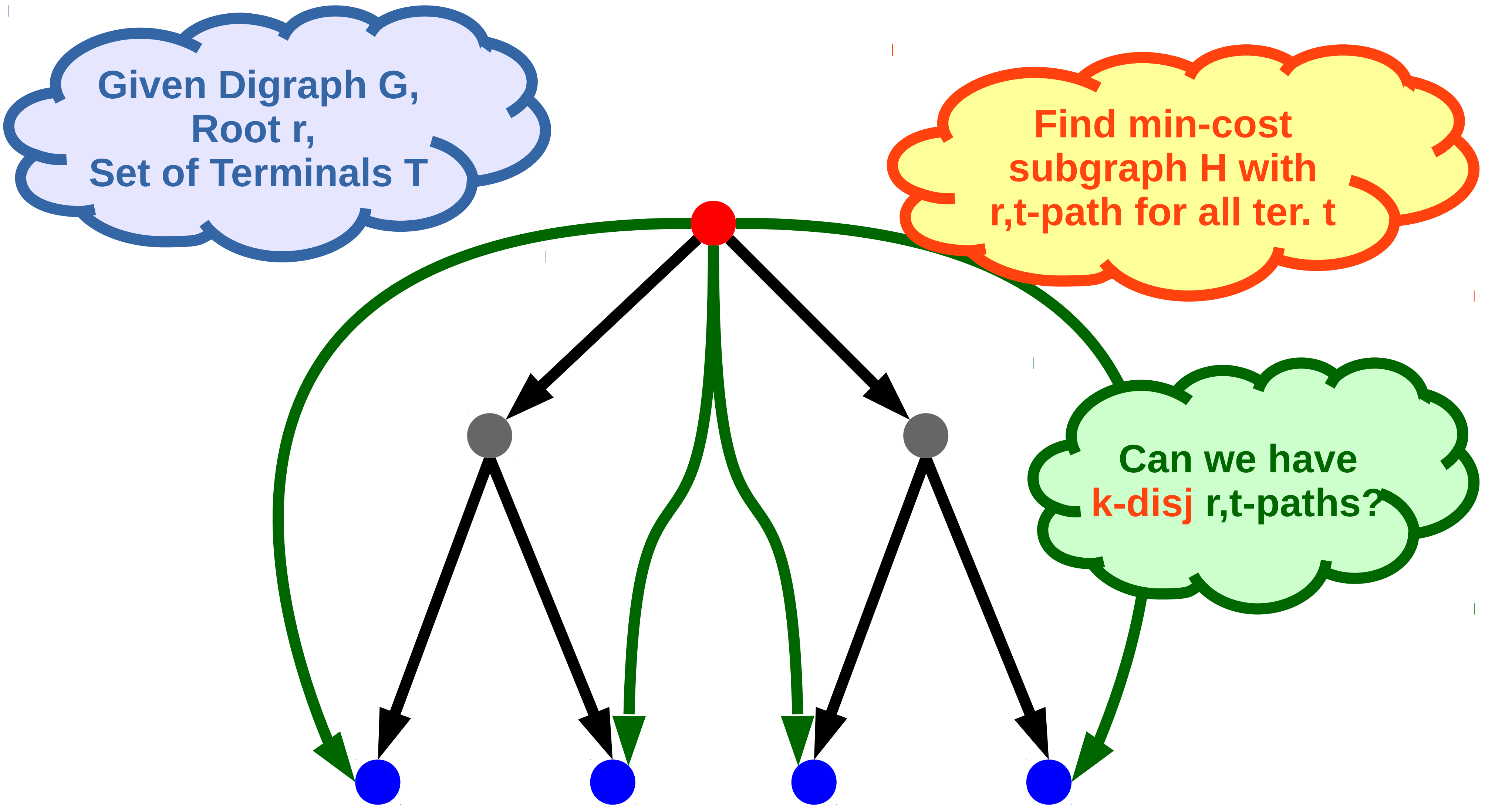


Can we have
2-disj r,t-paths?

Directed Steiner Tree (DST)



k-Connected DST (k-DST)



“A problem that we do not know anything for”

G. Kortsarz

http://crab.rutgers.edu/~guyk/fpt.pptx

What're known in late 90's

Algorithms		
DST	PolyTime	|T|^c, for c>0 [CCCDGGL'99]
	QPolyTime	(log |T|)^3 [CCCDGGL'99]
k-DST		

Hardness	
DST	Set Cover Hard
k-DST	DST Hard

What're known in 2000's

Algorithms		
DST	PolyTime	|T|^c, for c>0 [CCCDGGL'99]
	QPolyTime	(log |T|)^3 [CCCDGGL'99]
k-DST		

Hardness	
DST	(log |T|)^(2-ε) [HK'02]
k-DST	DST Hard

Currently known

Algorithms		
DST	PolyTime	|T|^c, for c>0 [CCCDGGL'99]
	QPolyTime	(log |T|)^3 / log log |T| [GL'15]
k-DST		

Hardness		
DST	(log |T|)^(2-ε) [HK'02]	
2-DST	DST Hard	
k-DST (k >> 2)	k << |T|	k^1/2 [L'14]
	k >> |T|	|T|^1/4 [L'14]
	General	n^c, ∃ c >0 [CLNV'12]

ETH
Gap SAT

This Talks

k-DST

$O(D\, k^{D-1} \log n)$ Approx for D-Shallow [L'15]
[Depth-D DAG is a special case]

2-DST

$\tilde{O}((\log n)^2 (\log |T|)^2)$ Approx for General [GL'16]
[in Quasi-Polynomial-Time]

This Talks

k-DST

O(D k^{D-1} log n) Approx for D-Shallow [L'15]
[Depth-D DAG is a special case]

2-DST

Õ((log n)^2 (log |T|)^2) Approx for General [GL'16]
[in Quasi-Polynomial-Time]

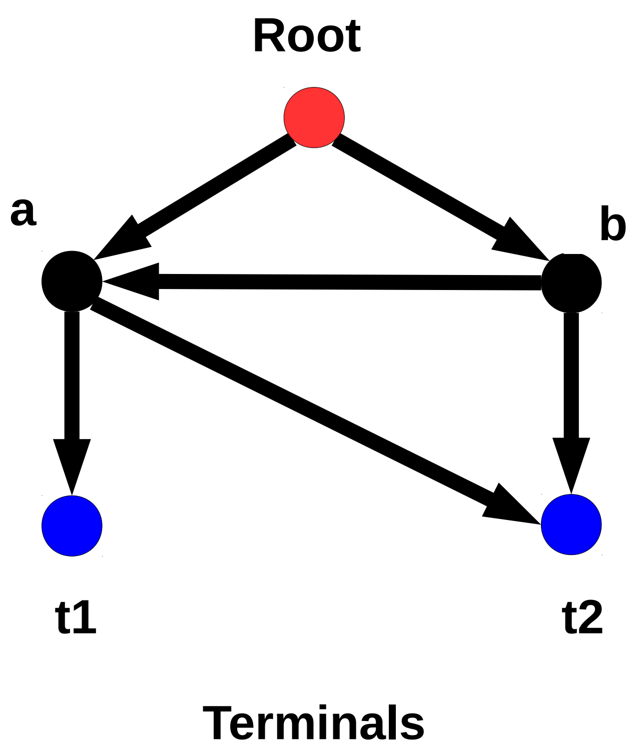
Please blame me for any bug!

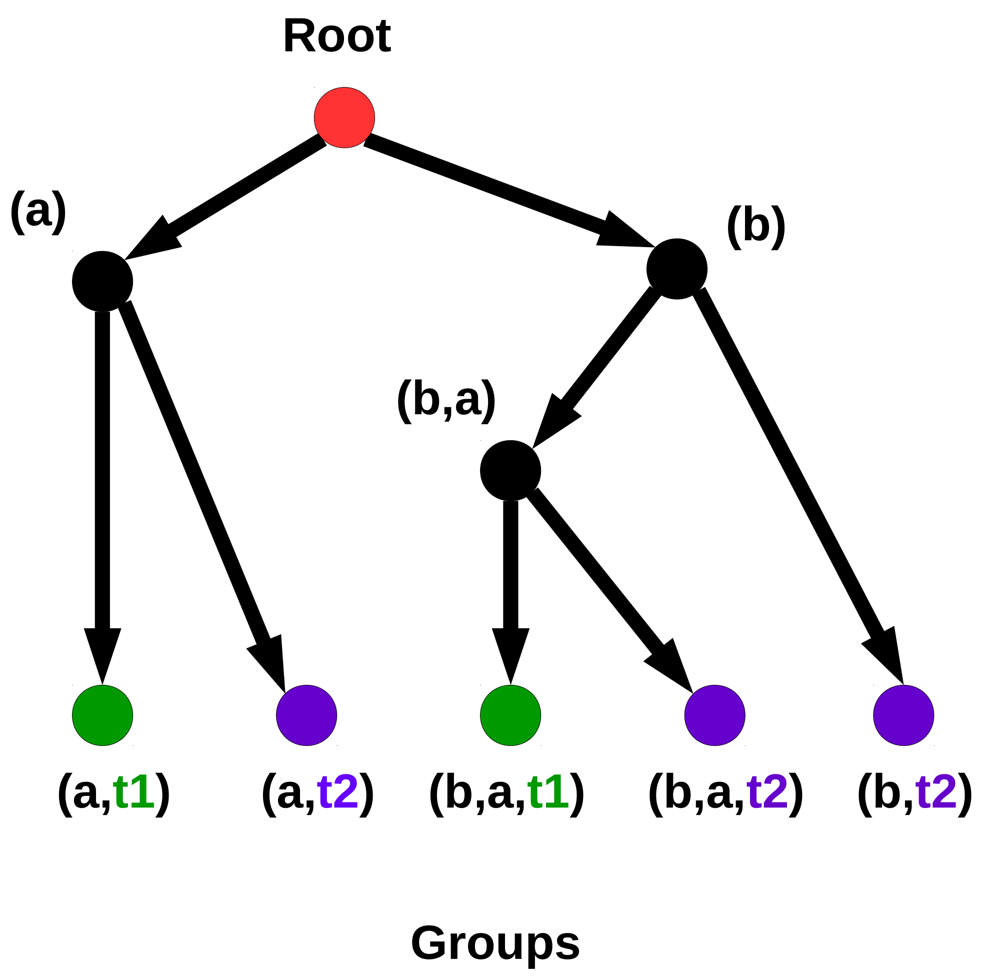
Our **key ingredient** is
a reduction from DST to
Group Steiner Tree
(GST) on Trees

Our **key ingredient** is
a reduction from DST to
Group Steiner Tree
(GST) on Trees

which **does not** work
for k-DST at all!

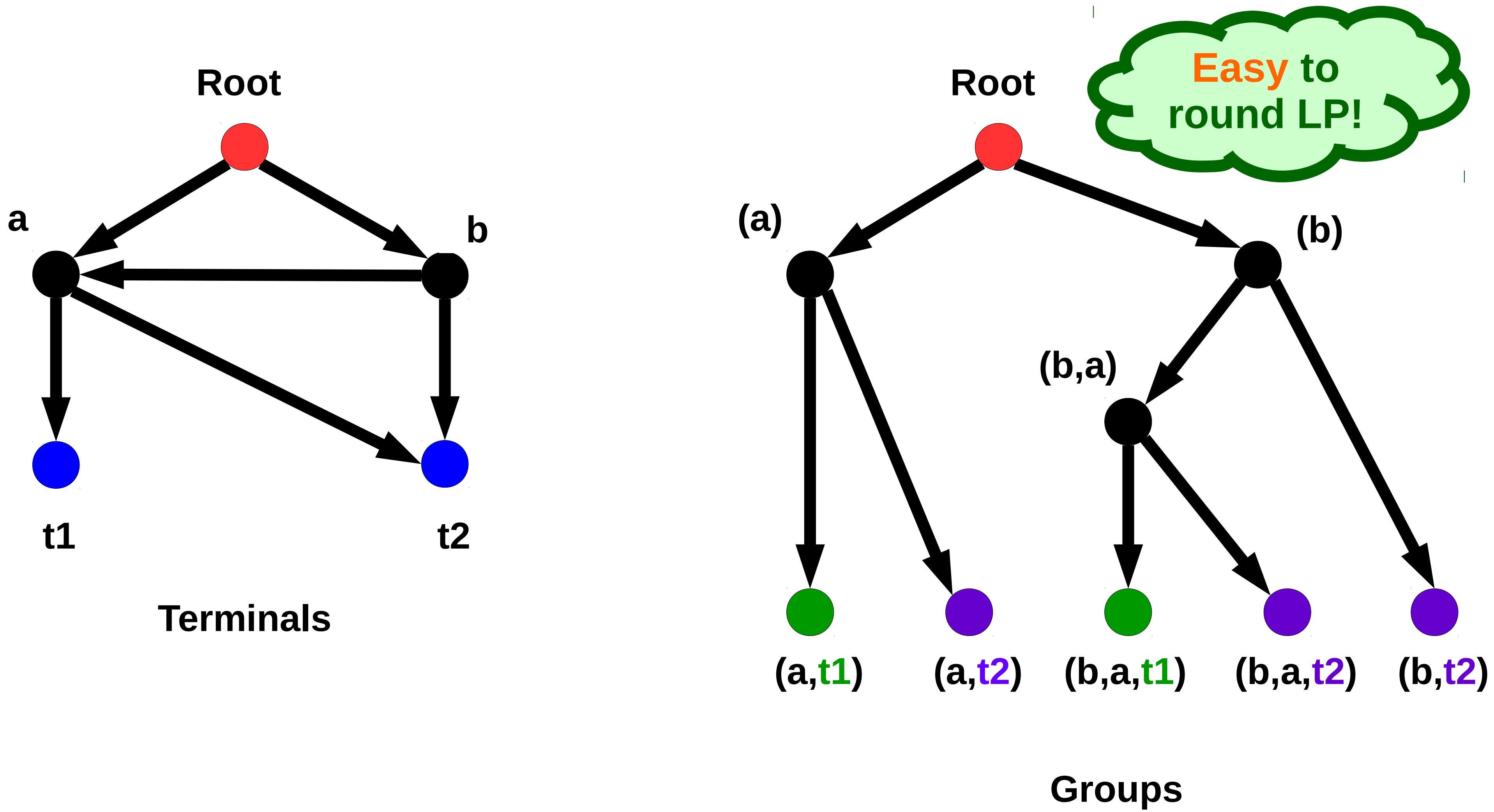
DST to GST on Trees





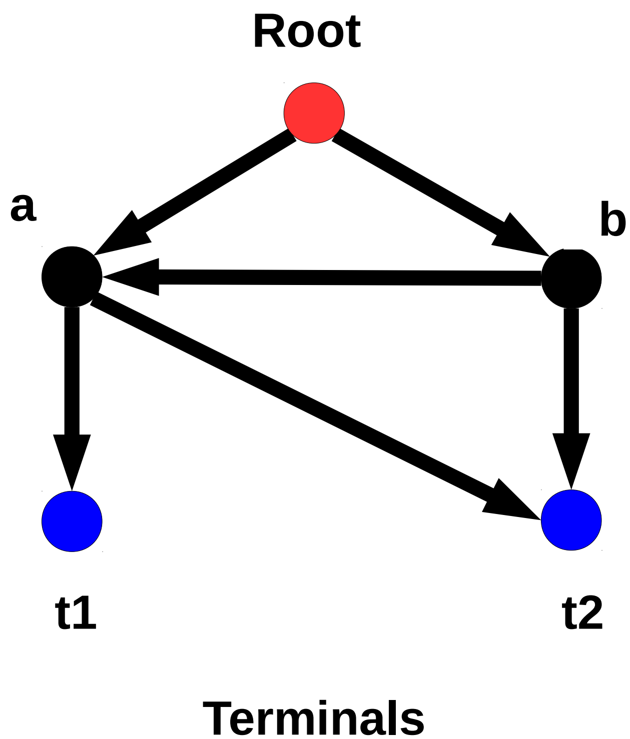
List All paths (of length D)

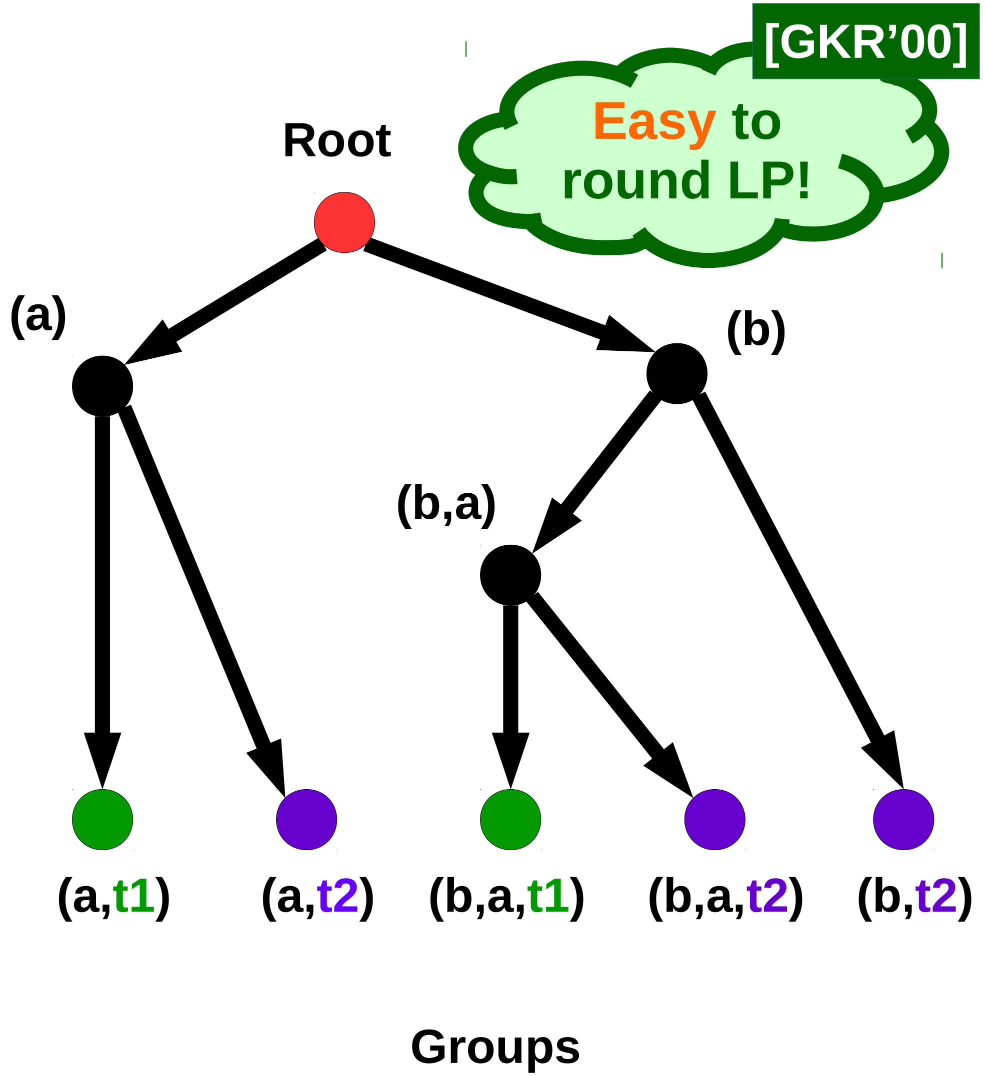
DST to GST on Trees



List All paths (of length D)

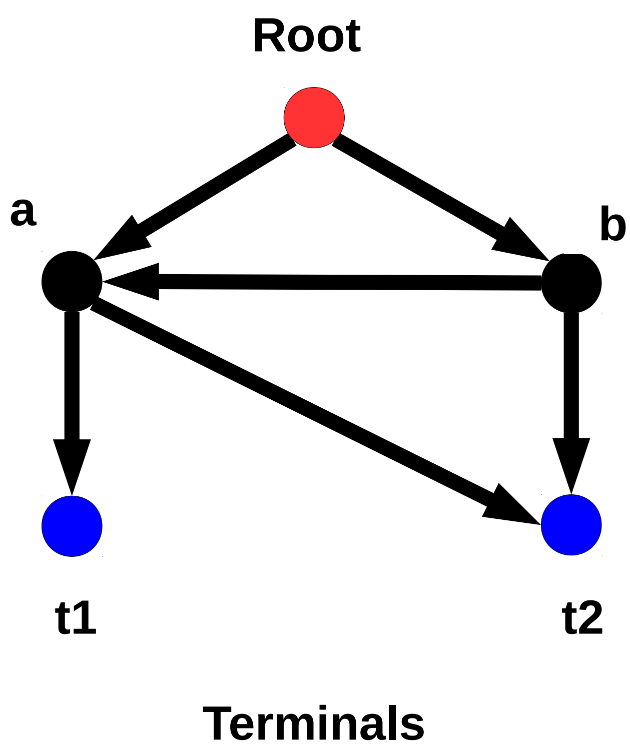
DST to GST on Trees

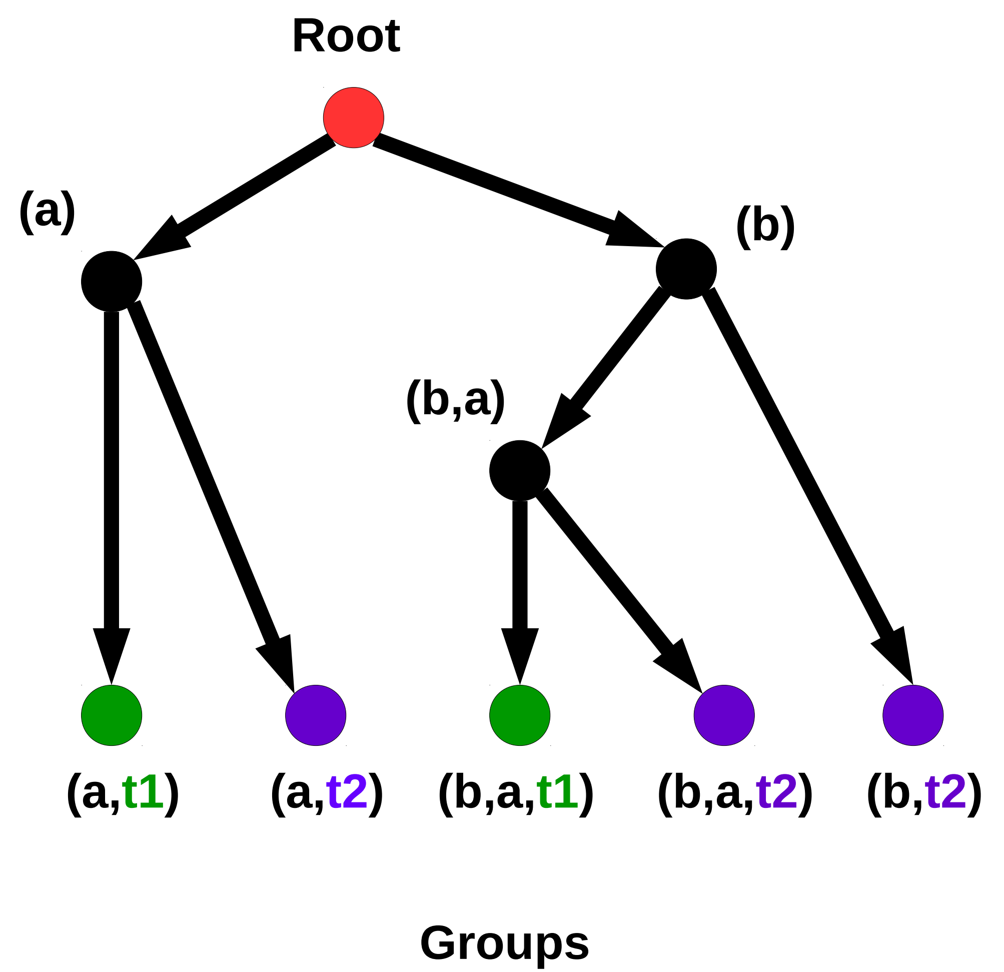




List All paths (of length D)

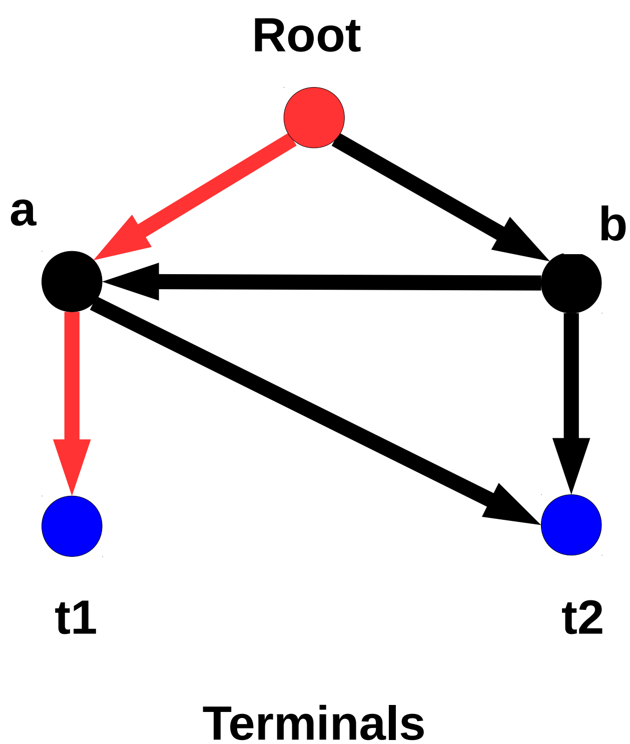
k-DST to GST on Trees?

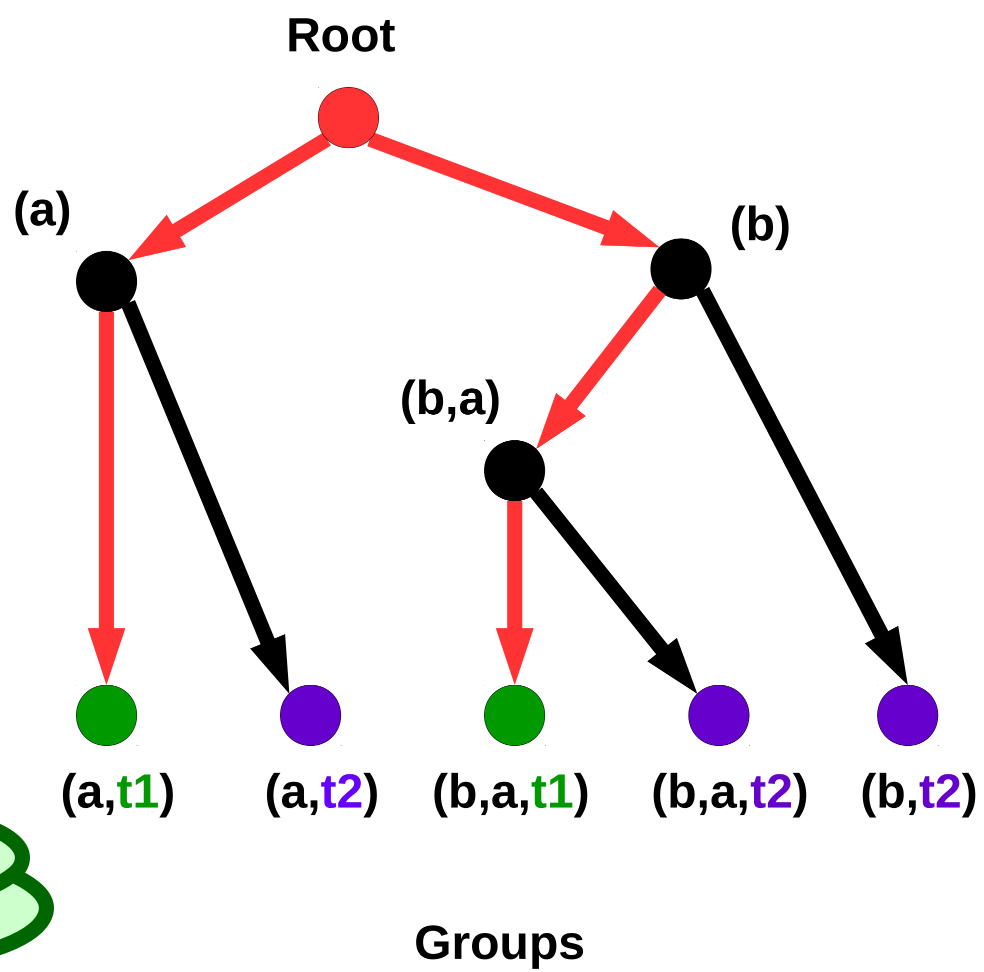




List All paths (of length D)

k-DST to GST on Trees?



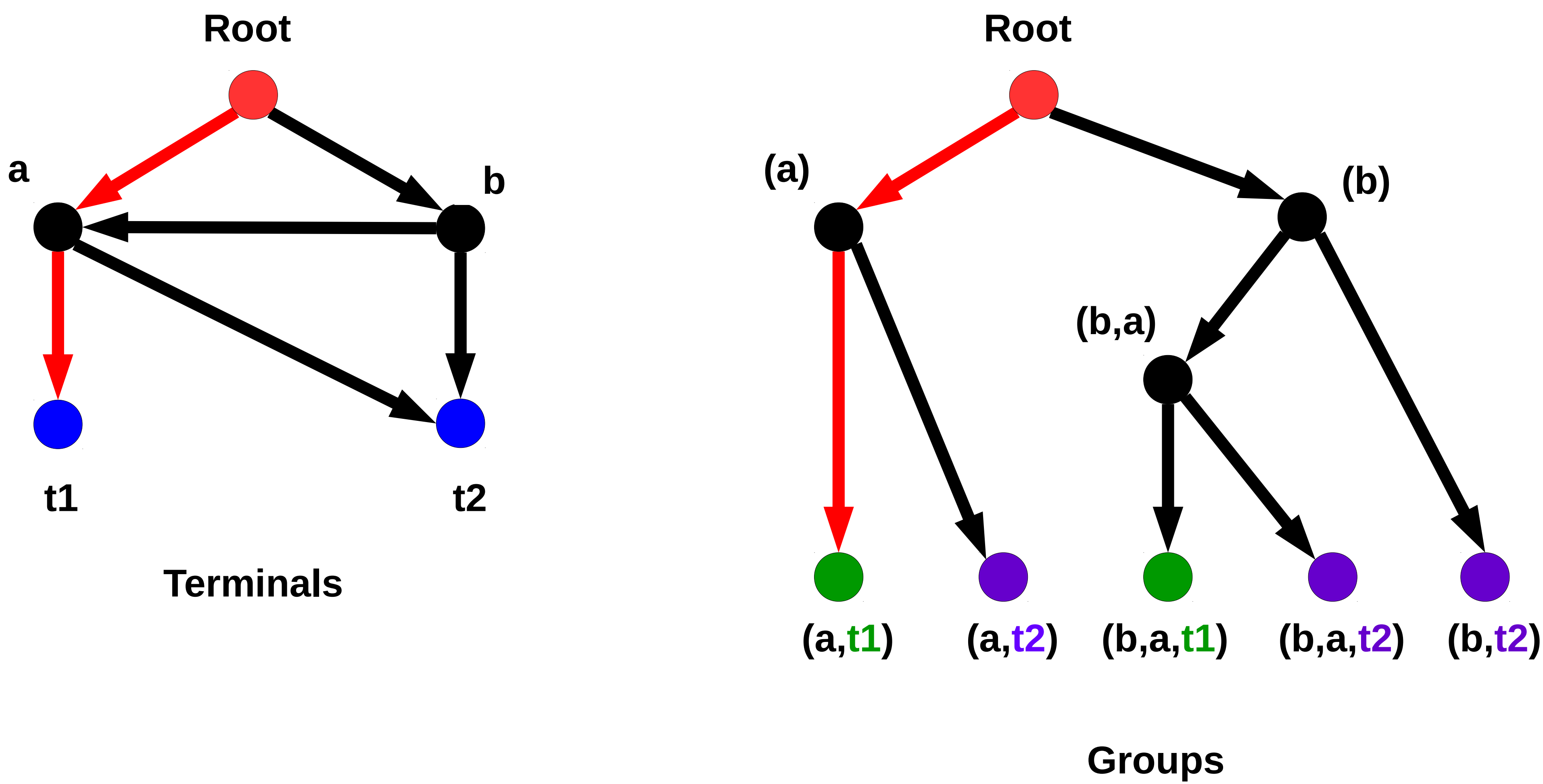


This does not preserve connectivity!

List All paths (of length D)

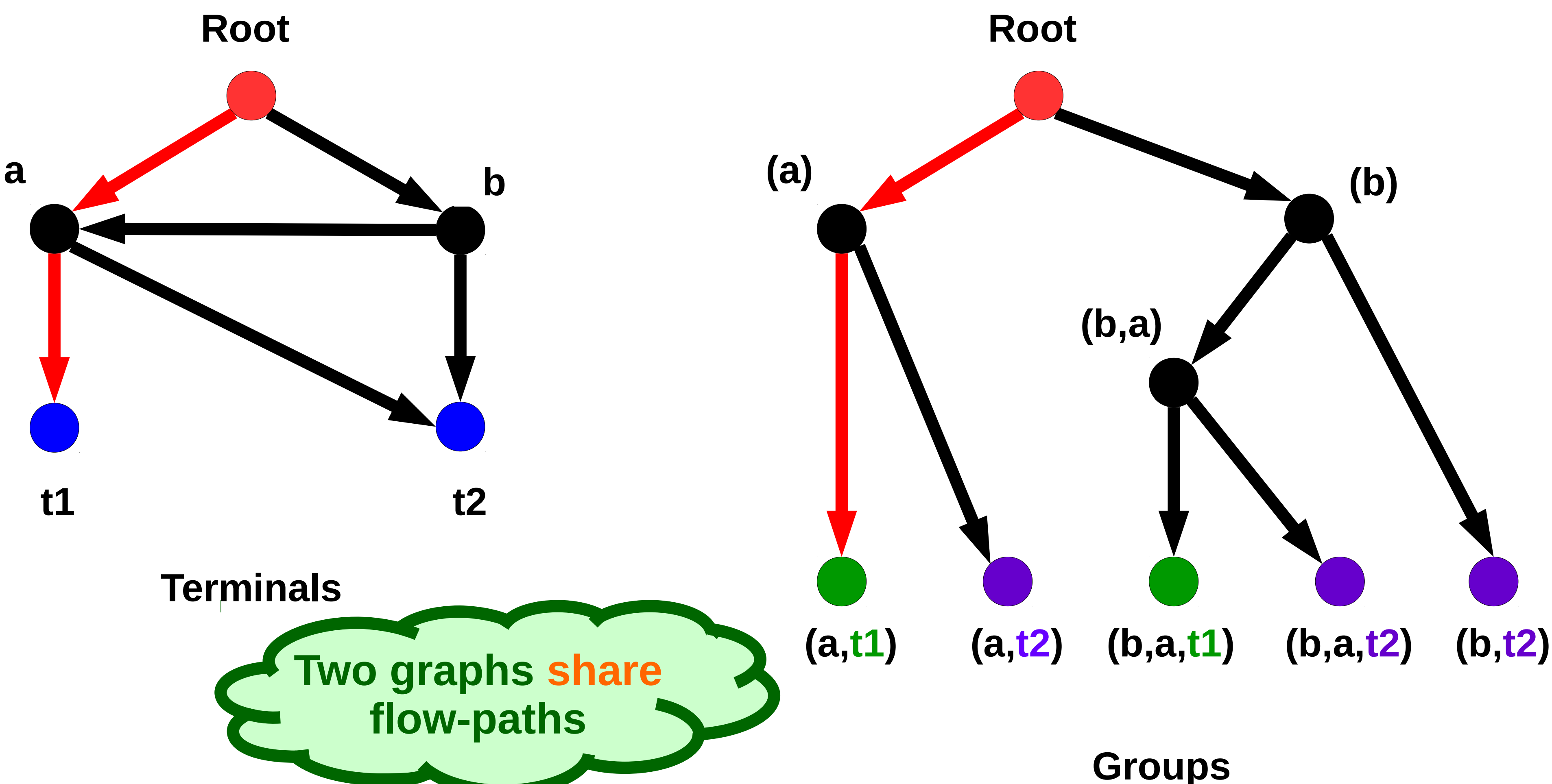
How to make it work?

k-DST to GST on Trees?



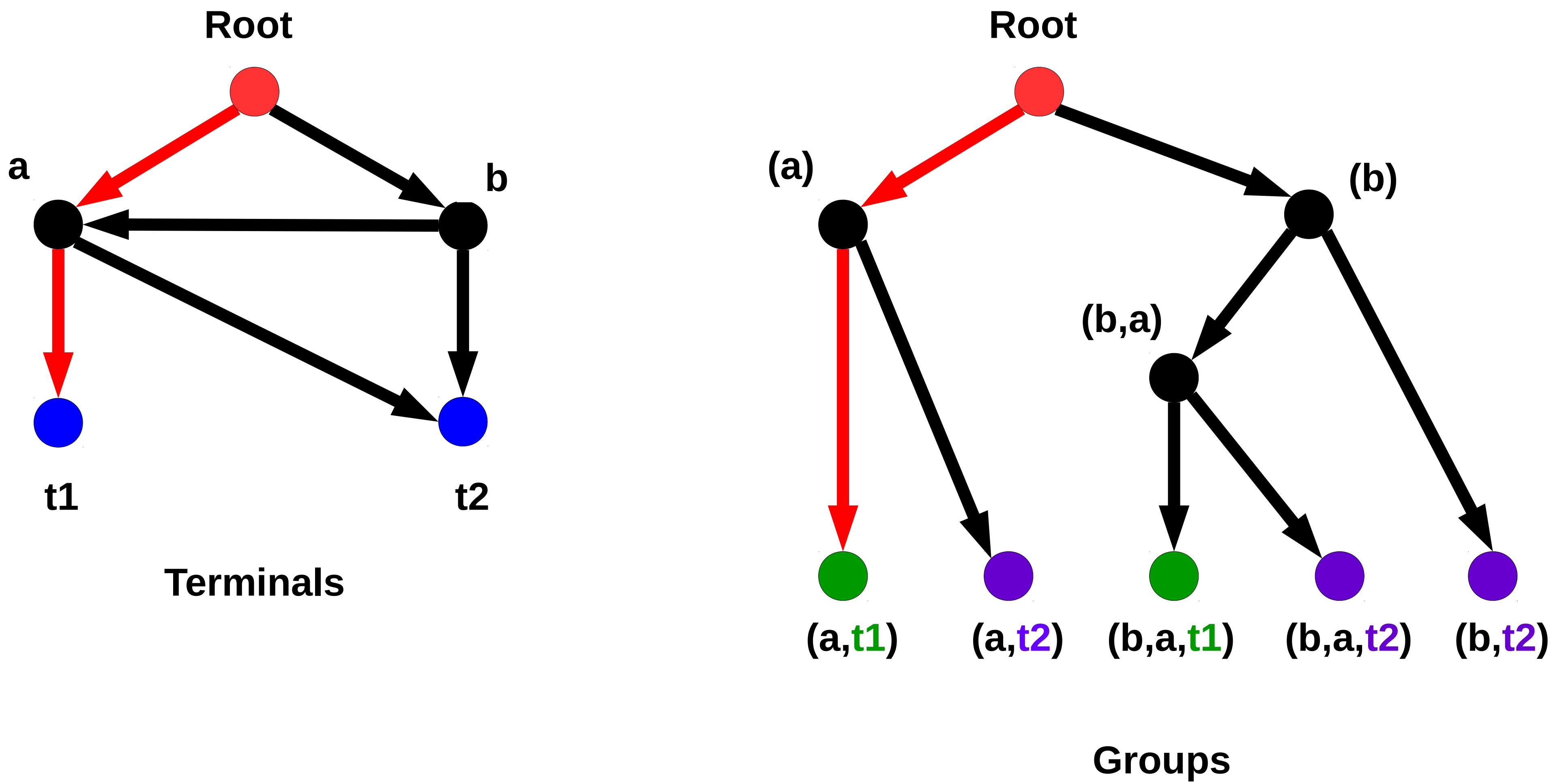
There is a 1-1 map of paths.

k-DST to GST on Trees?



There is a 1-1 map of paths.

k-DST to GST on Trees!!!



Consolidate two graphs with LP

Analysis

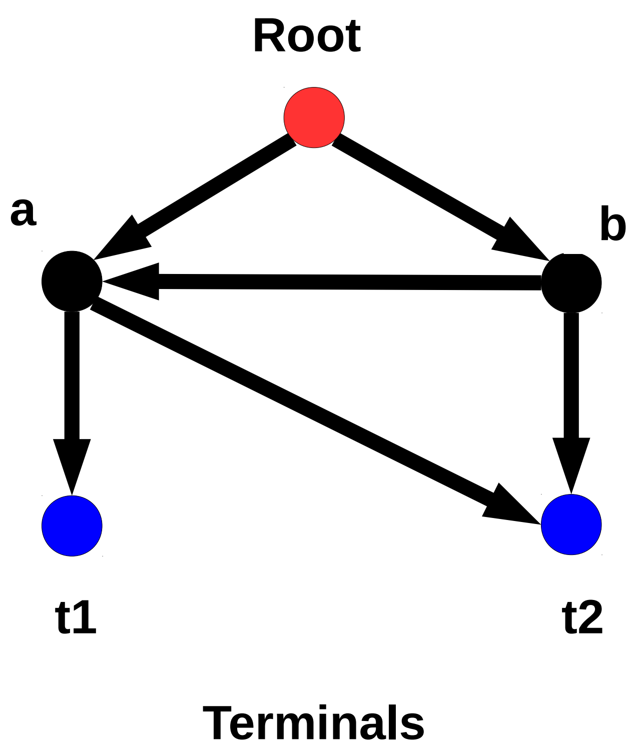
Pay $O(k^{D-2})$ for k-DST => Tree-GST

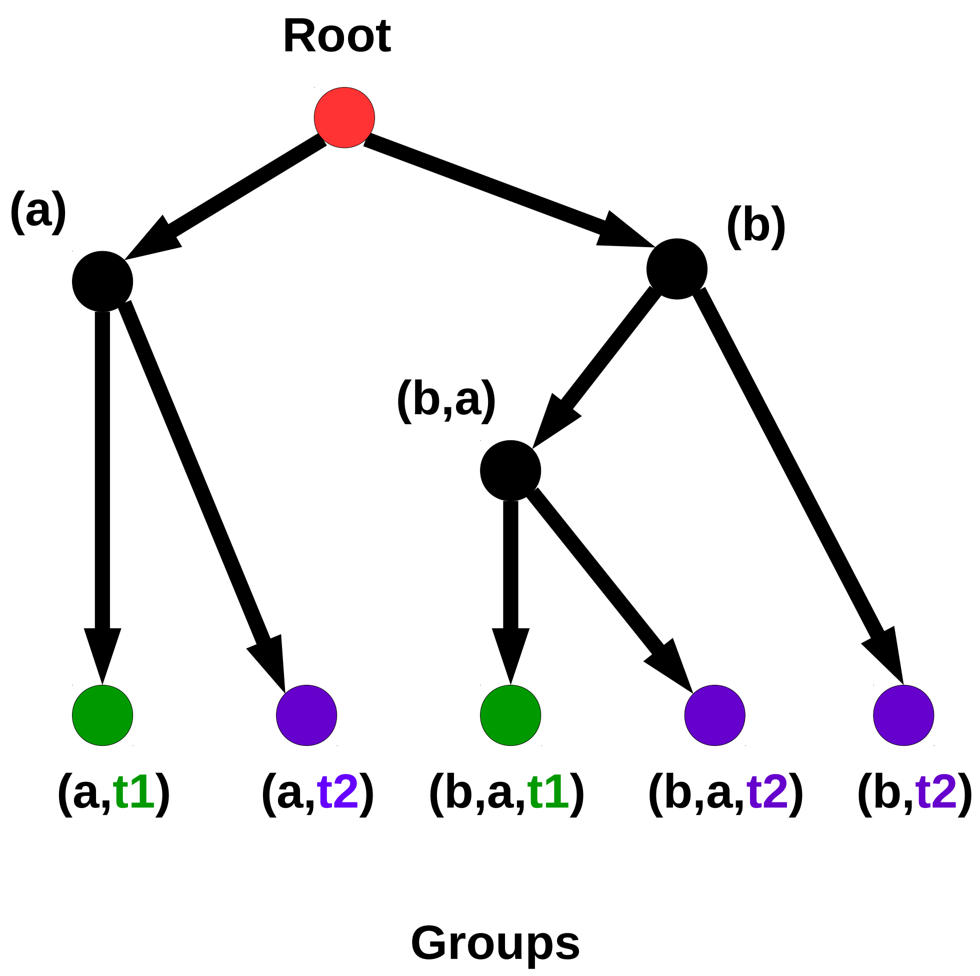
Round Tree-GST $O(k D \log n)$ times

Total Cost = $O(D k^{D-1} \log n)$ OPT

Generalization to 2-DST on General Graphs.

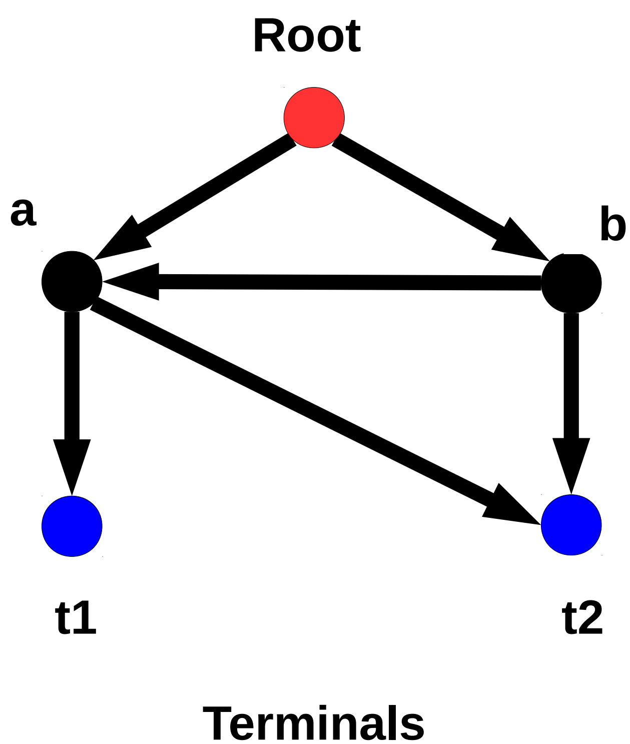
2-DST to GST on Trees

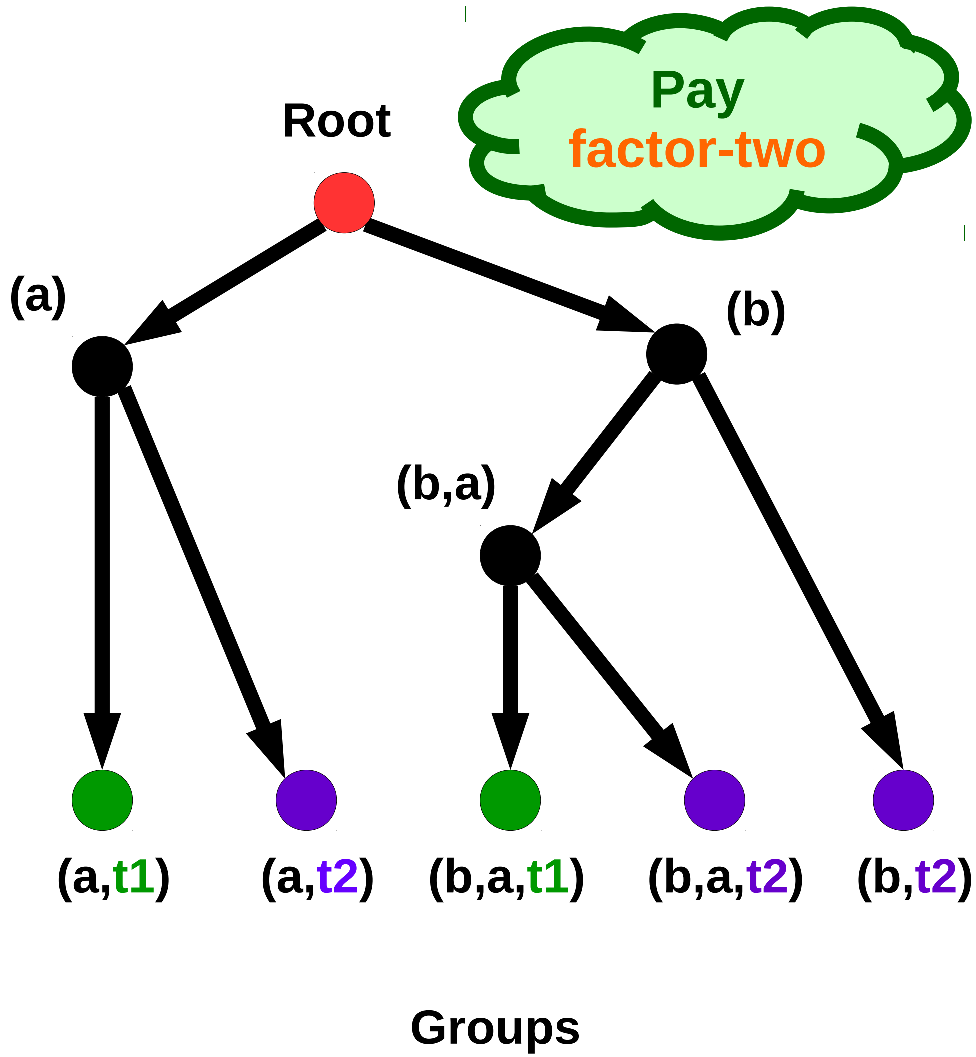




List All paths (of every length)

2-DST to GST on Trees

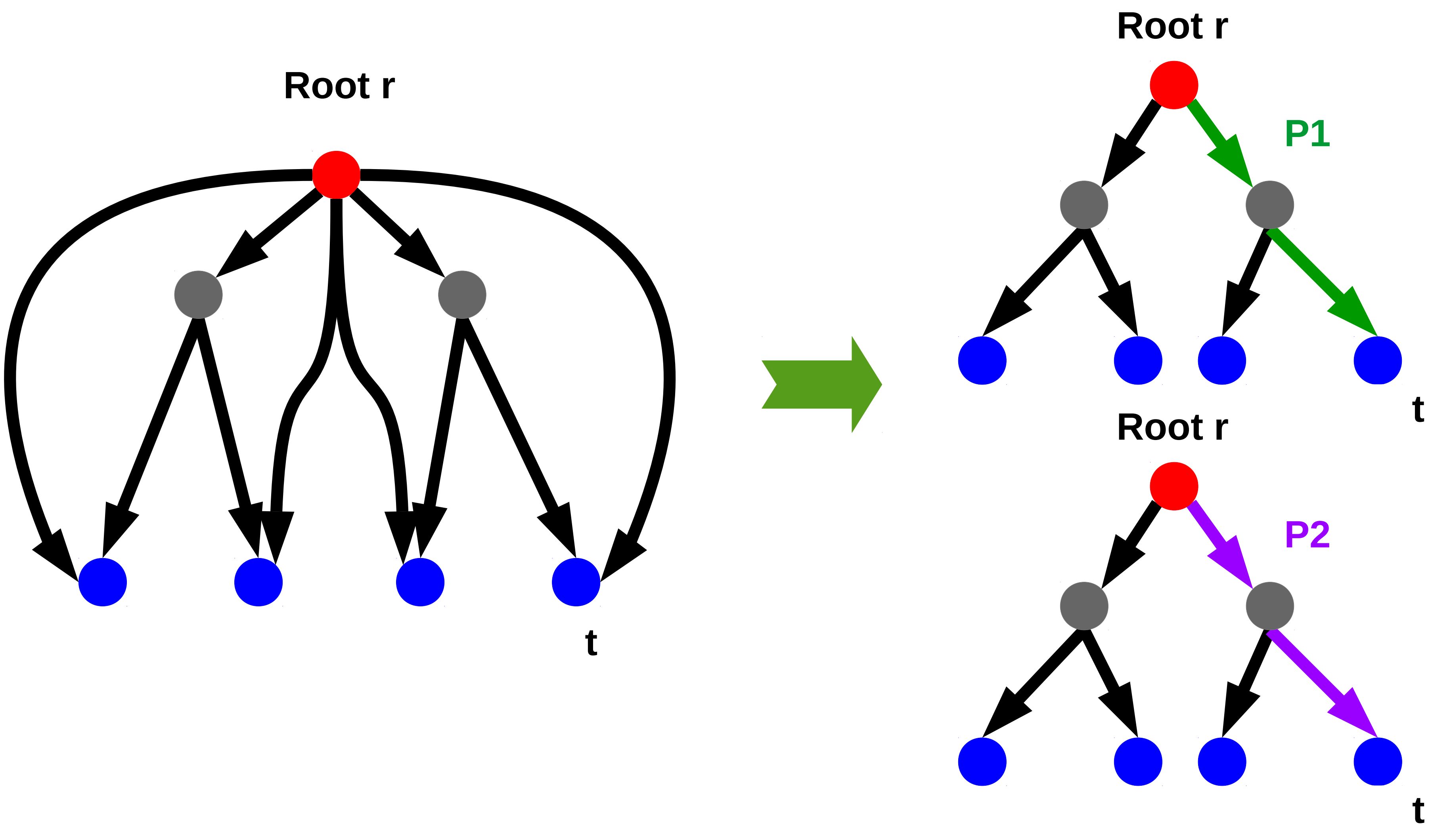




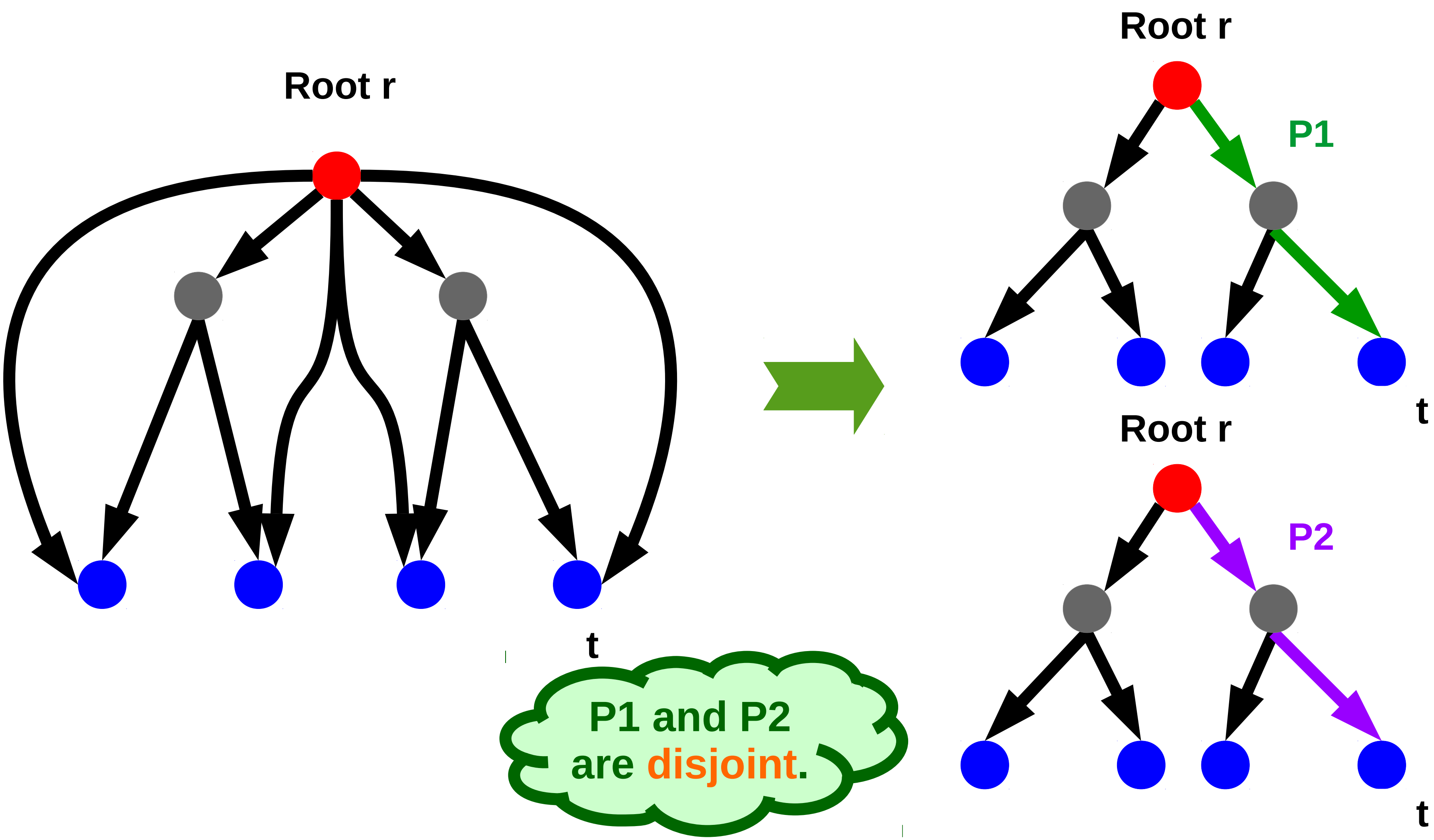
List All paths (of every length)

How can we pay only factor-two?

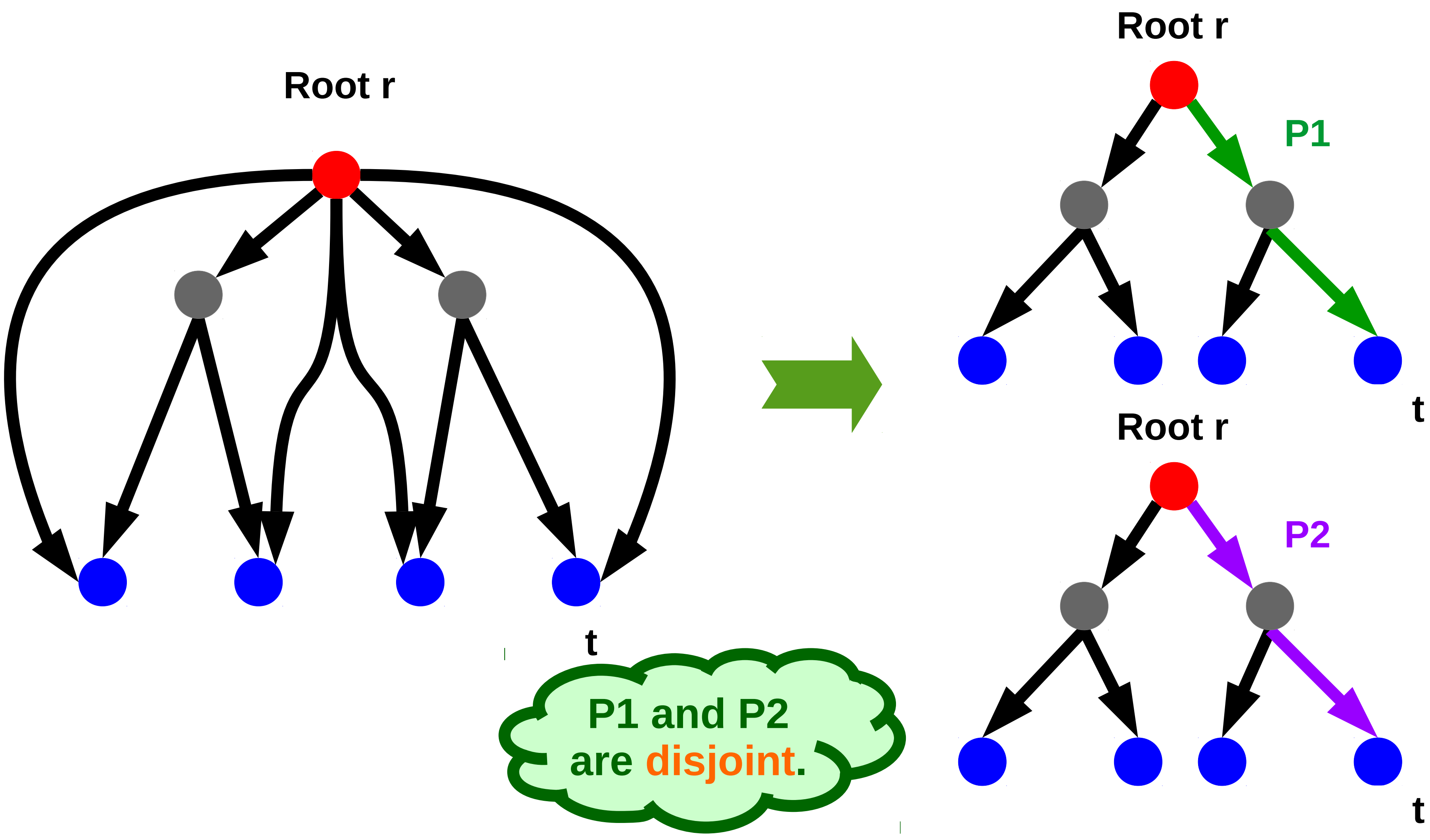
Independent Trees



Independent Trees

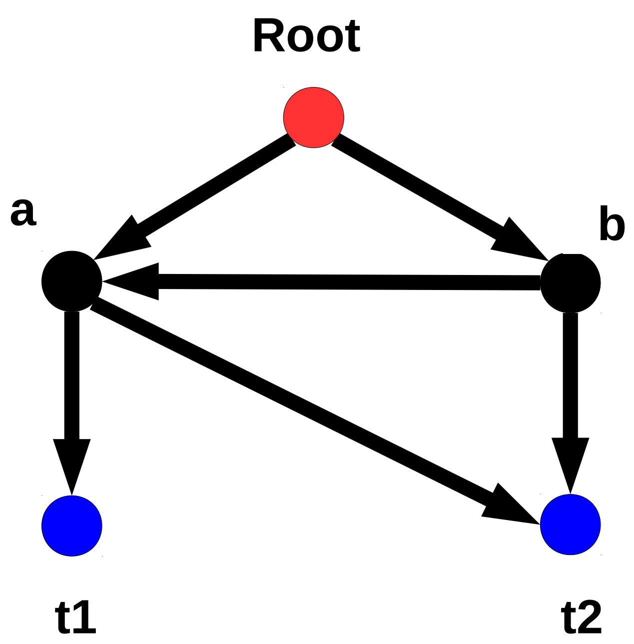


Independent Trees

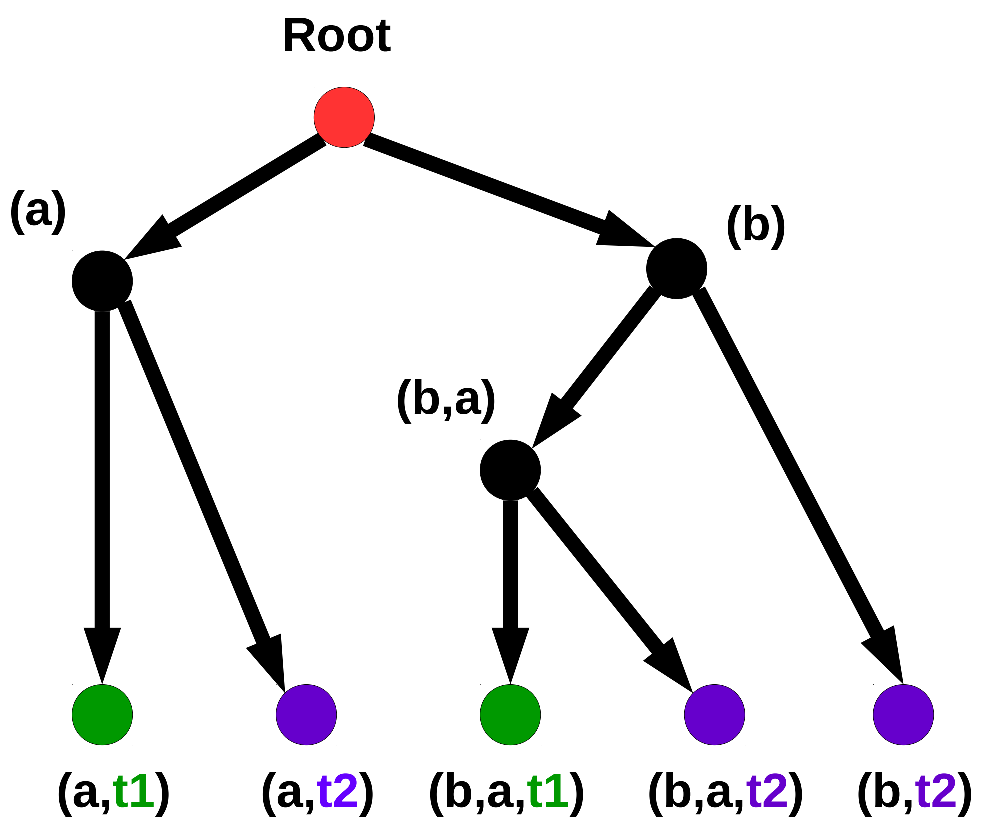


2-DST has 2 Indep. Trees [GT05,K11] (but not Edge Disjoint)

Consolidate LPs again?

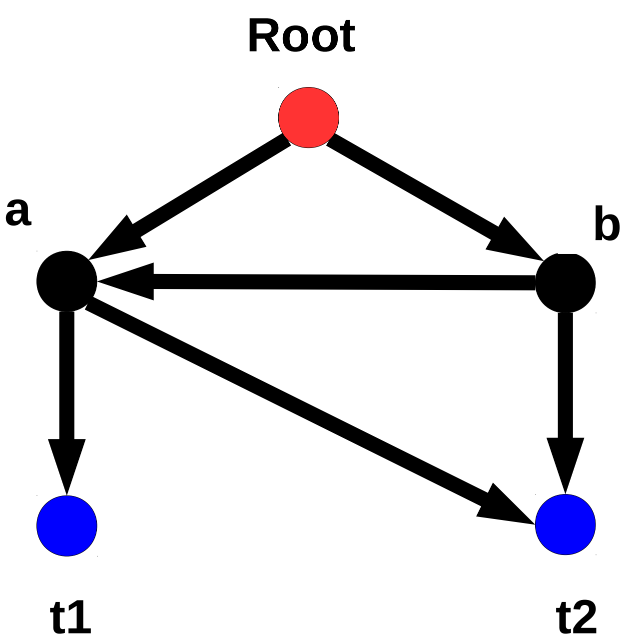


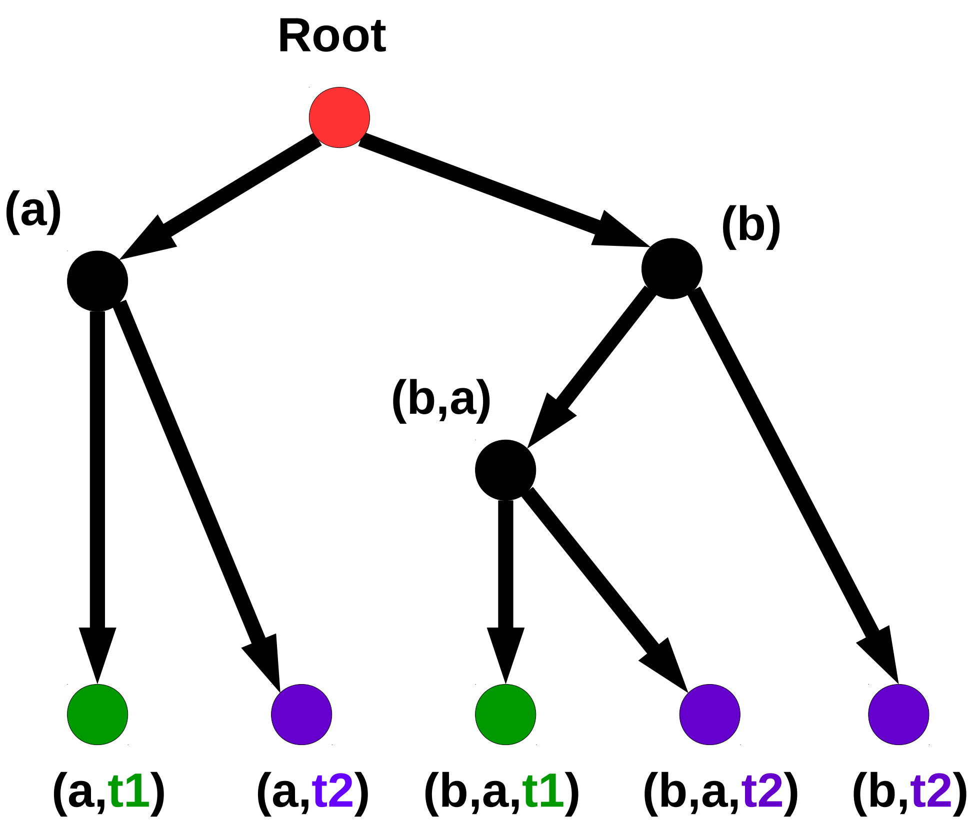
LP1: LP-DST



LP2: LP-GST

Consolidate LPs again?



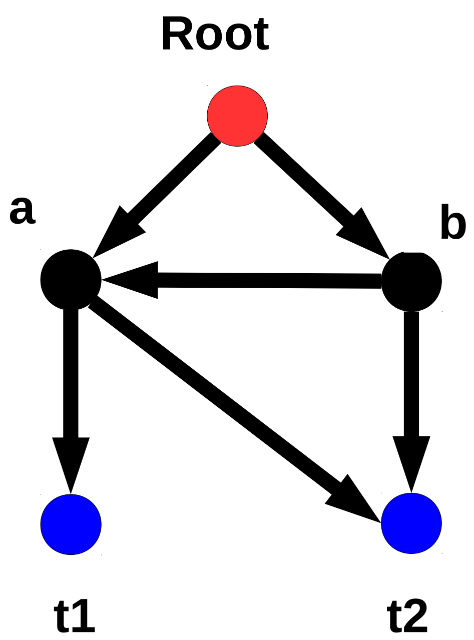


LP1: LP-DST

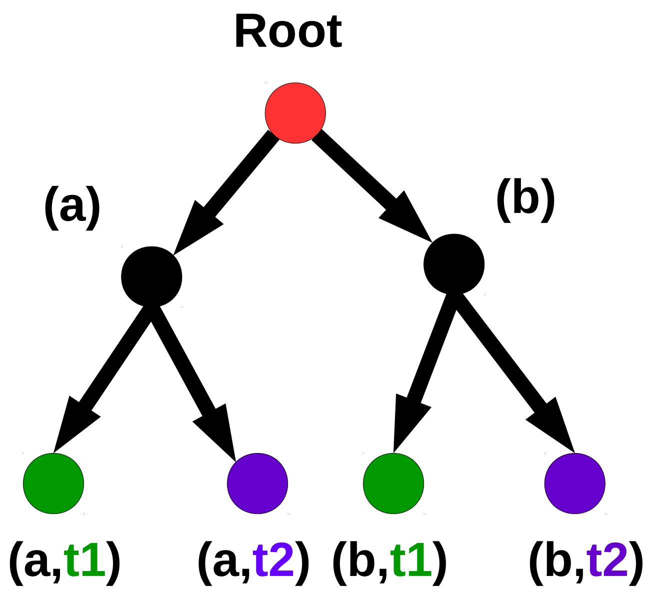
LP2: LP-GST

Problematic!
LP2 has expo-size

Apply Zelikovsky's Height Reduction!



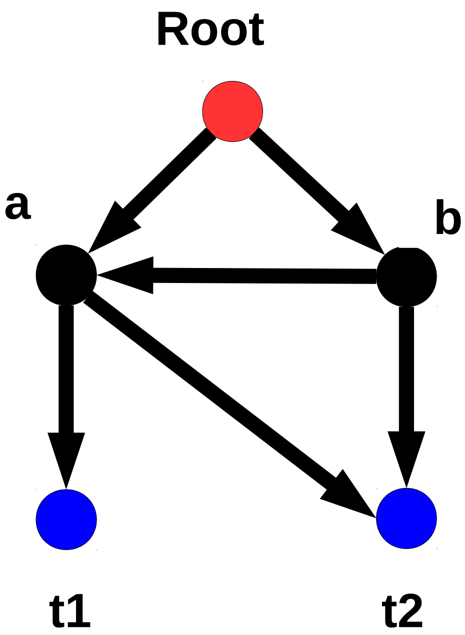
LP1: LP-DST



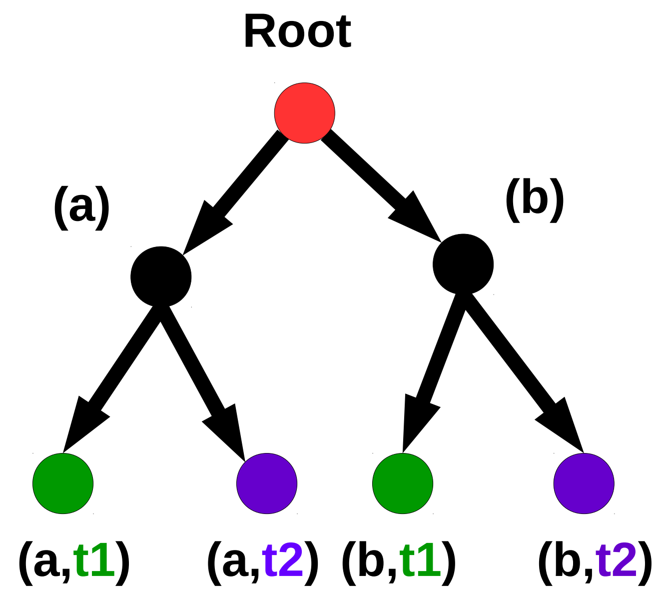
LP2: LP-GST (low-depth)

Problematic!
No metric-completion

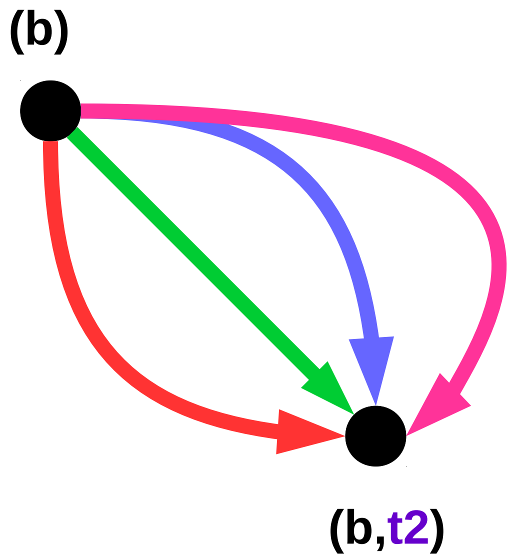
Apply Zelikovsky's Height Reduction!



LP1: LP-DST



LP2: LP-GST (low-depth)



LP3: Mapping-LP

Let LP find a mapping!
Then consolidate all LPs!

Analysis

Pay O(β) for 2-DST => Short Tree-GST

Round Tree-GST O(D log n) times

(Rand.) Path-Maping O(β log D) times

Total Cost = O(β^2 D log D log n) OPT

β = Height-Reduction Factor

Conclusion

$O(D k^D \log n)$-Approx for k-DST on D-shallow instances.

Polylog-Approx for 2-DST on General Graphs (in Quasi-Poly time).

Open Problem

3-DST does not have
3 indep.trees [BK'11].

Open Problem

3-DST does not have 3 indep.trees [BK'11].

Can we decompose k-DST into f(k) log^c n trees s.t. k trees support k-disj paths?

Thank you!