Offline Drawing of Dynamic Trees: Algorithmics and Document Integration

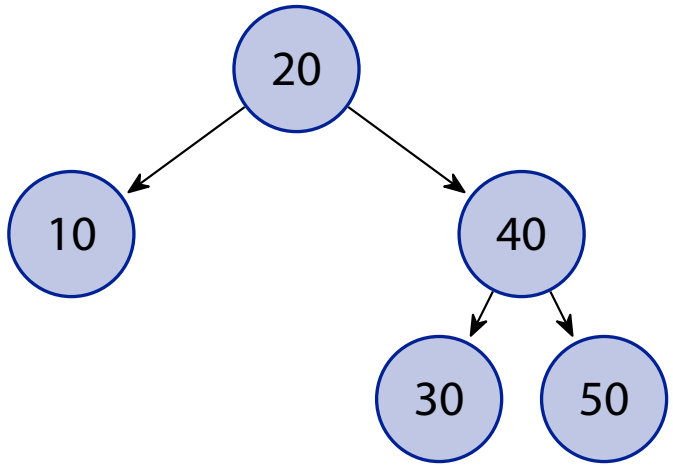
Graph Drawing Conference 2016

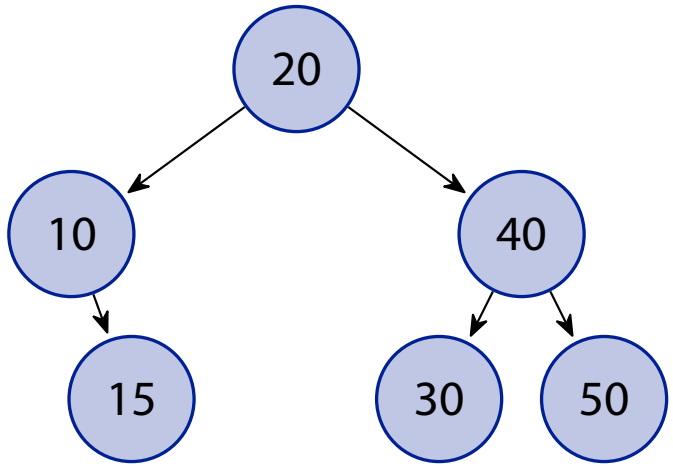
Malte Skambath[1] Till Tantau[2]

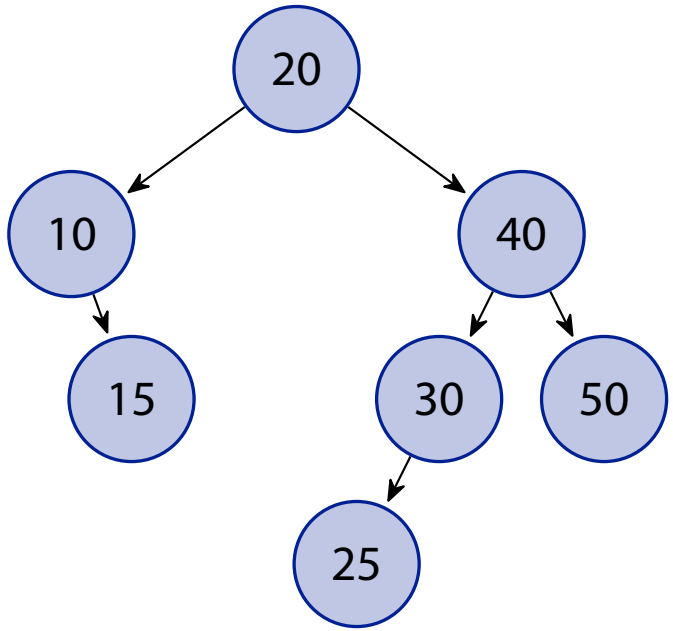
[1]Department of Computer Science
Kiel University

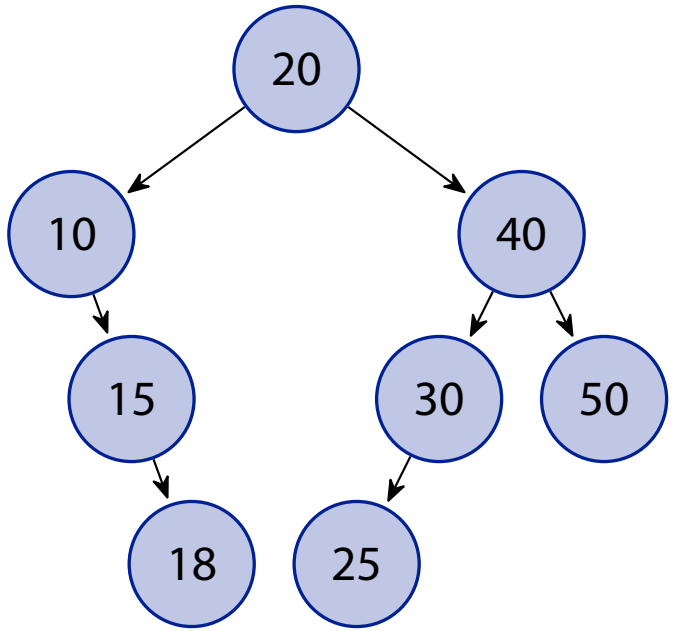
[2]Institute of Theoretical Computer Science
Universität zu Lübeck

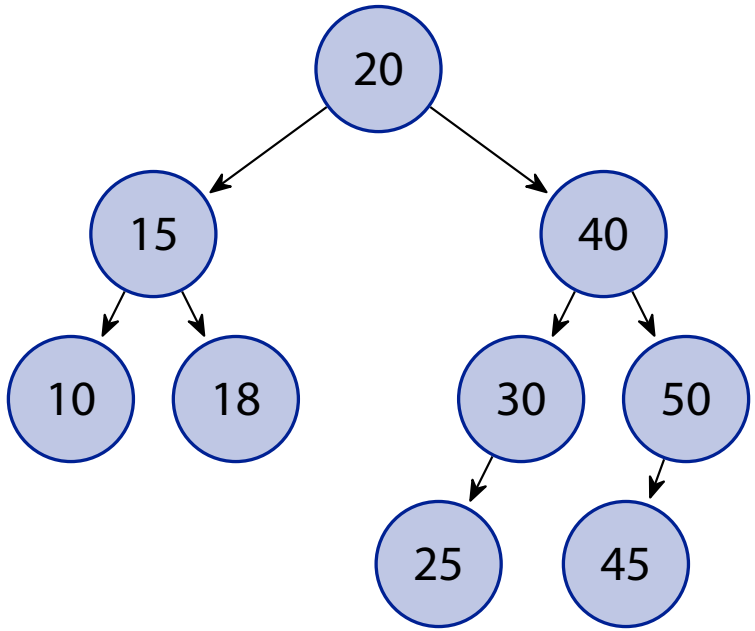
September 21, 2016

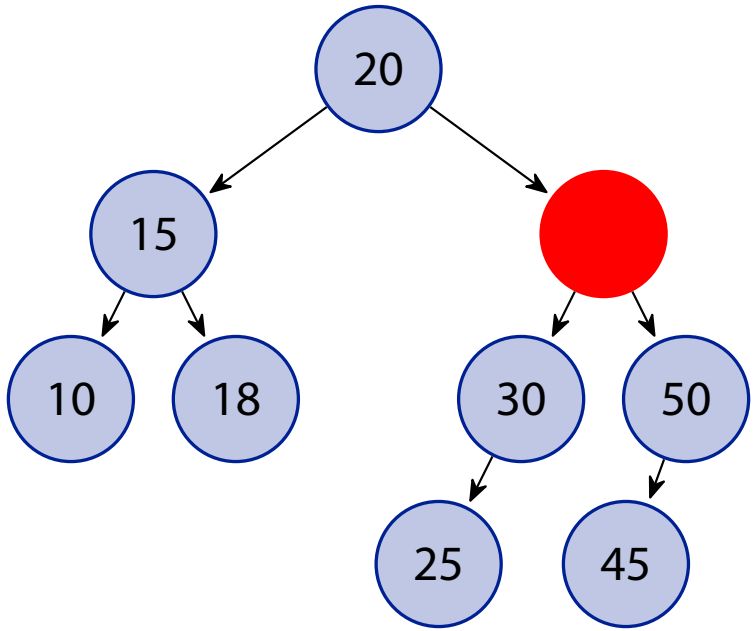

20
10
40
30
50

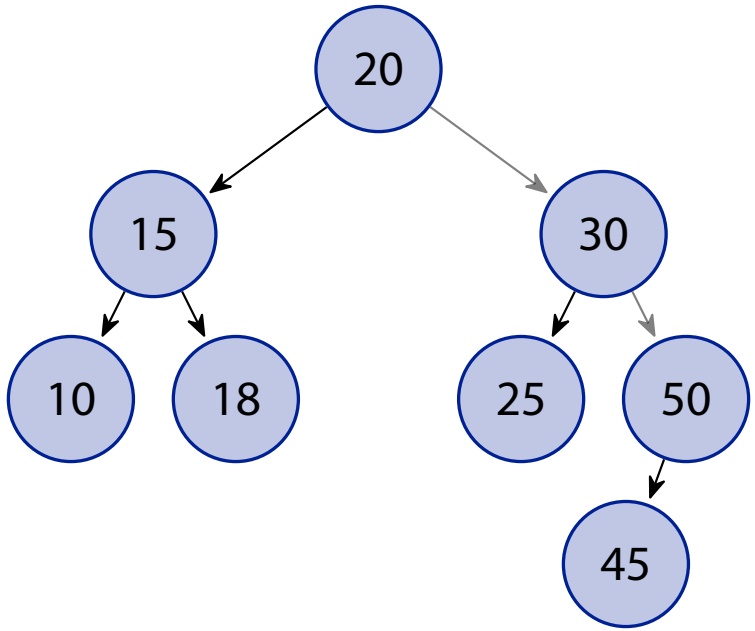

20
10
40
15
30
50

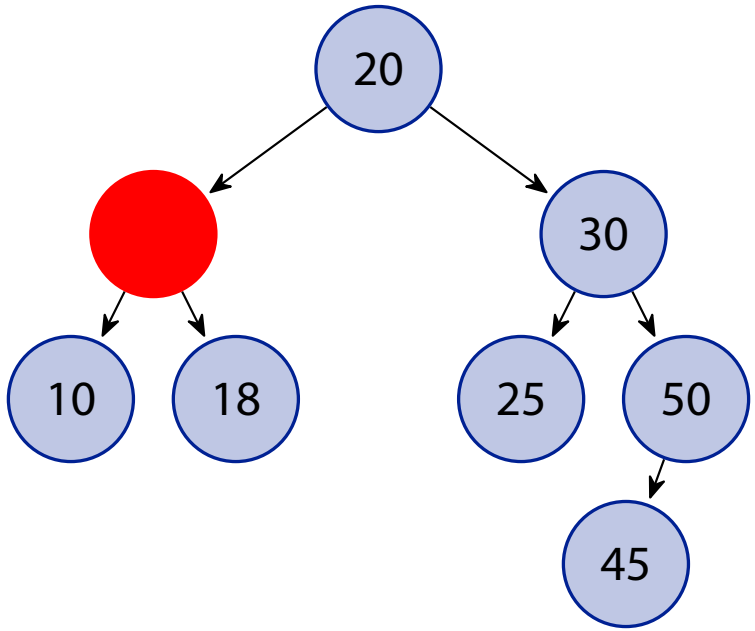

20
10
40
15
30
50
25

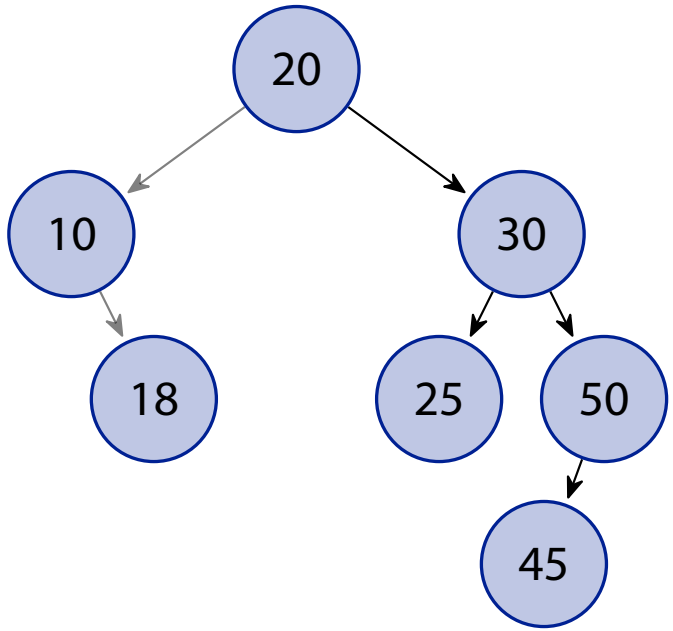

20
10
40
15
30
50
18
25

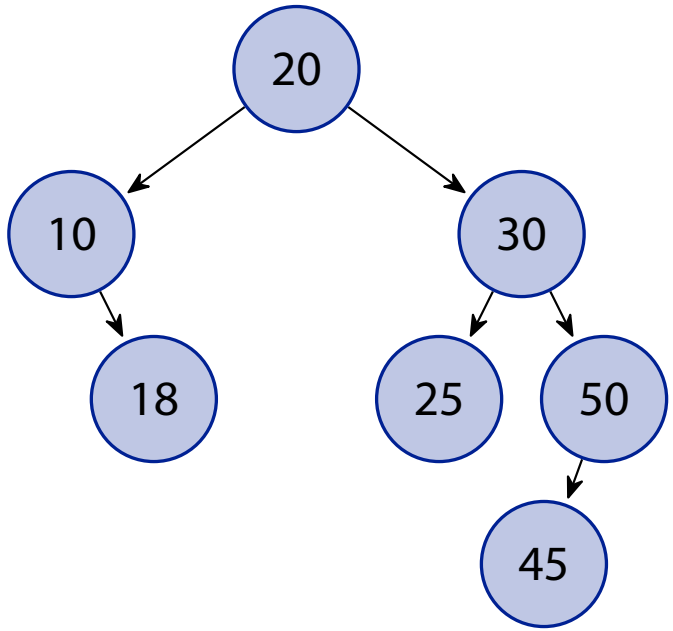

20
15
40
10
18
30
50
25
45

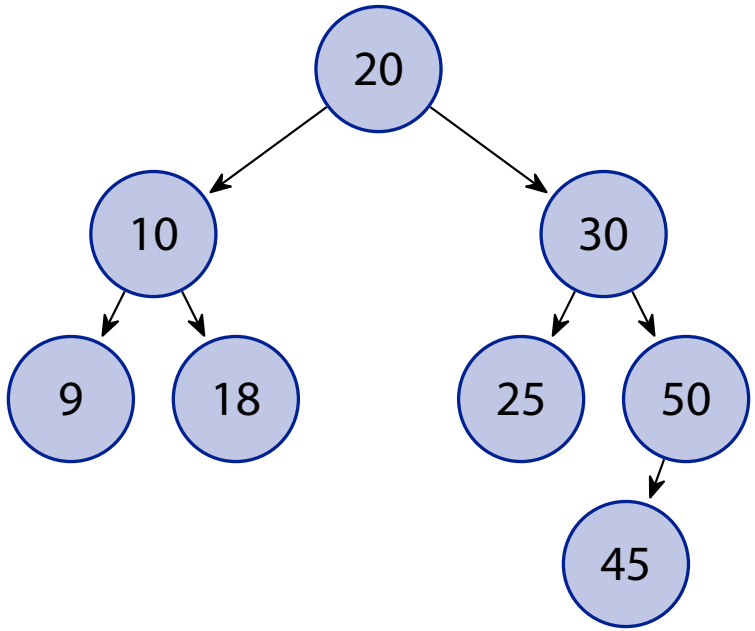

20
15
10
18
30
50
25
45

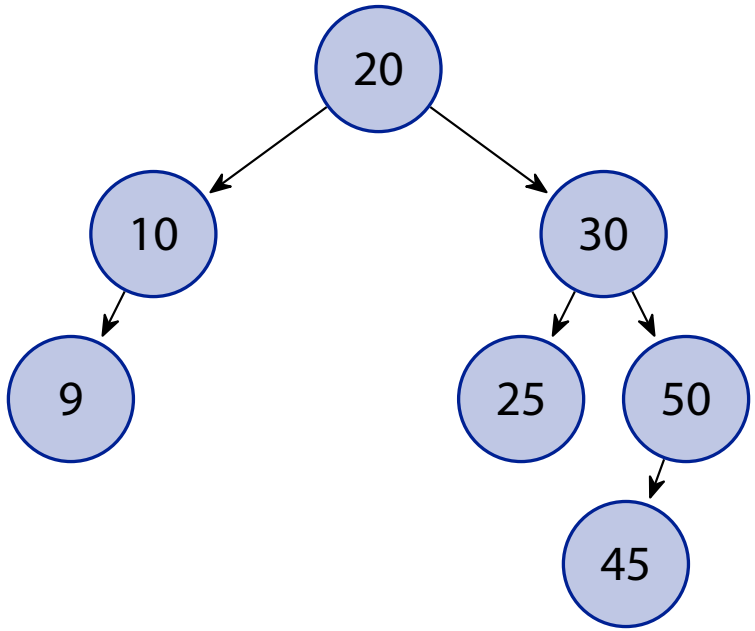

20
15
30
10
18
25
50
45

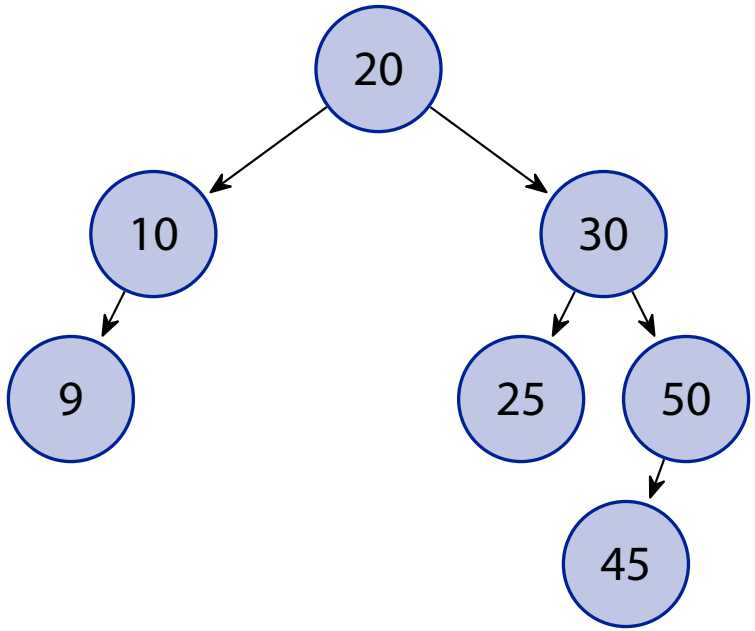

20
30
10
18
25
50
45


20
10
30
18
25
50
45

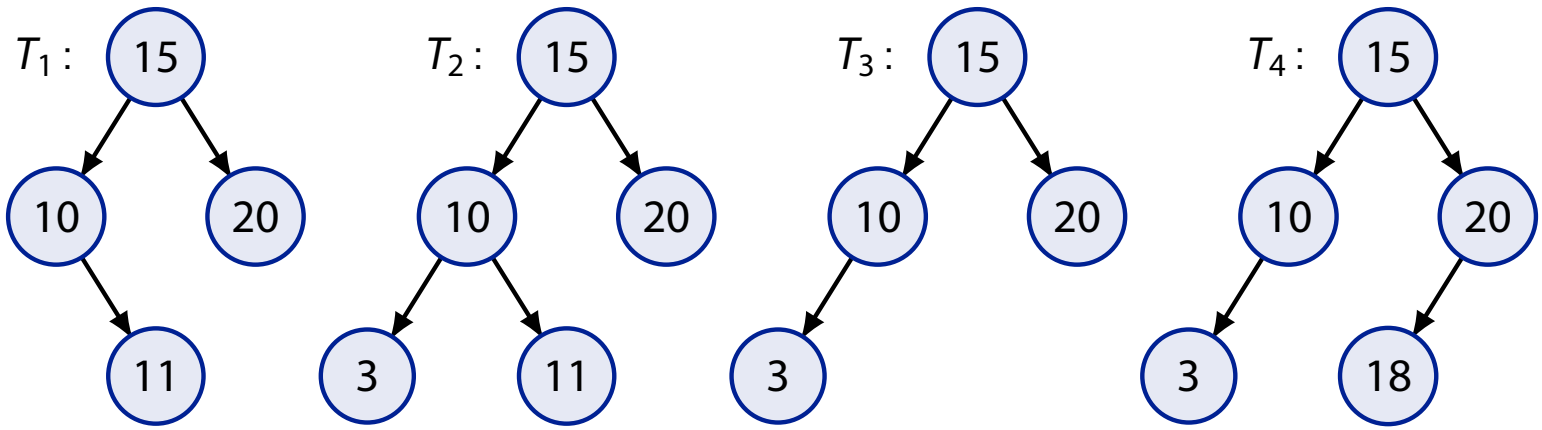

20
10
30
18
25
50
45


20
10
30
9
18
25
50
45


20
10
30
9
25
50
45


20
10
30
9
25
50
45

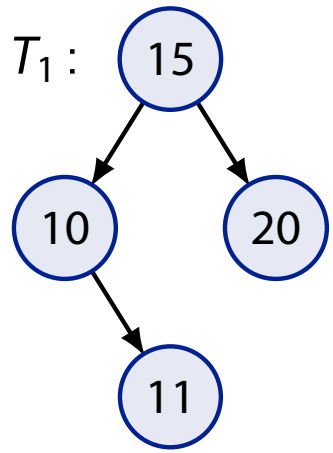
Animations in Documents: Why?



Animations. . .

- help seeing and recognizing temporal changes.
- save space.
- lead the attention in presentation.

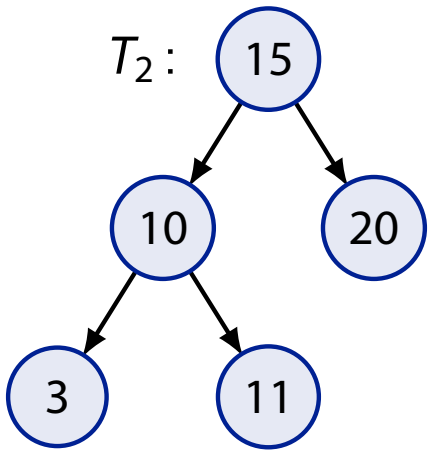
Animations in Documents: Why?



Animations. . .

- help seeing and recognizing temporal changes.
- save space.
- lead the attention in presentation.

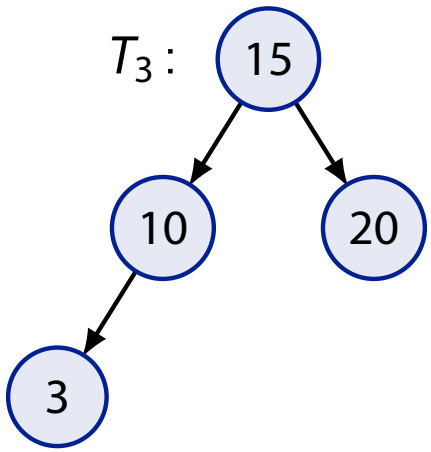
Animations in Documents: Why?



Animations. . .

- help seeing and recognizing temporal changes.
- save space.
- lead the attention in presentation.

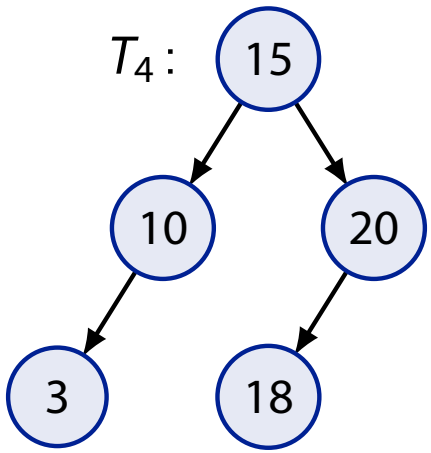
Animations in Documents: Why?



Animations. . .

- help seeing and recognizing temporal changes.
- save space.
- lead the attention in presentation.

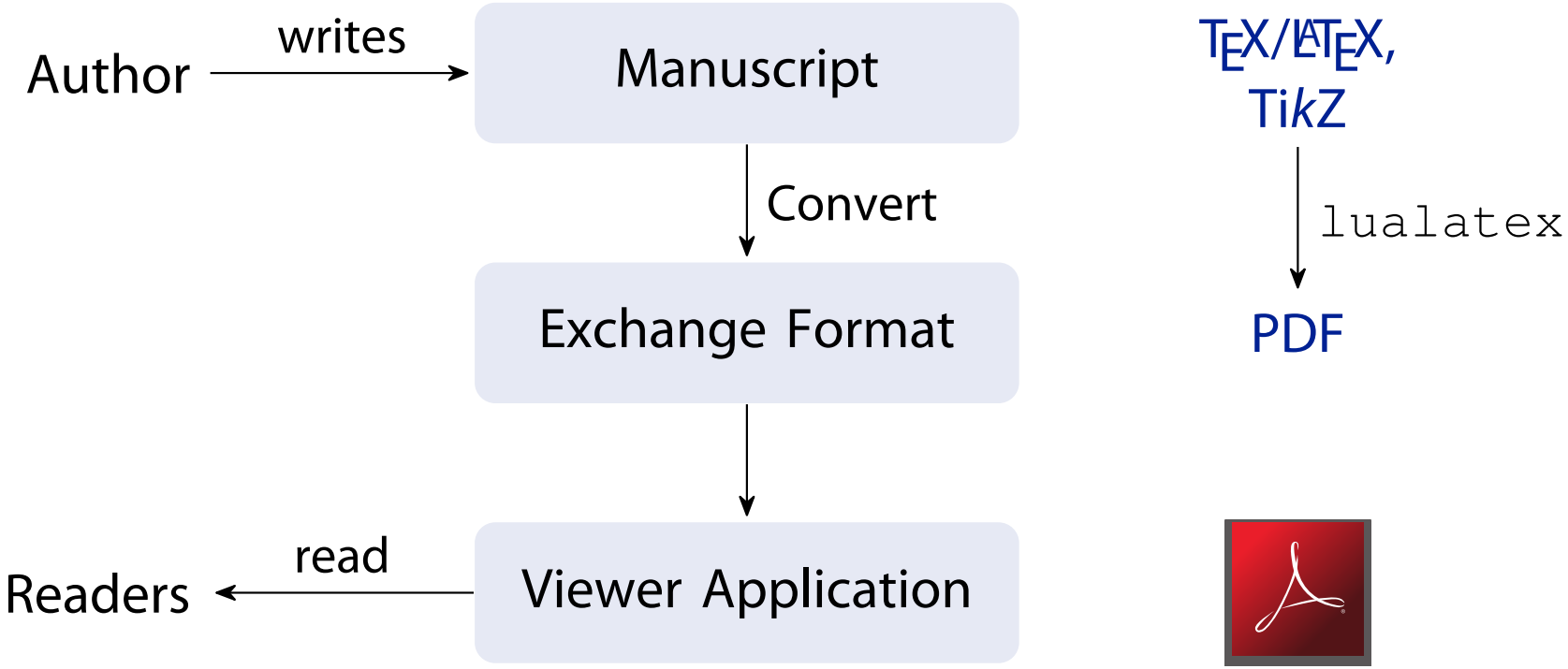
Animations in Documents: Why?



Animations. . .

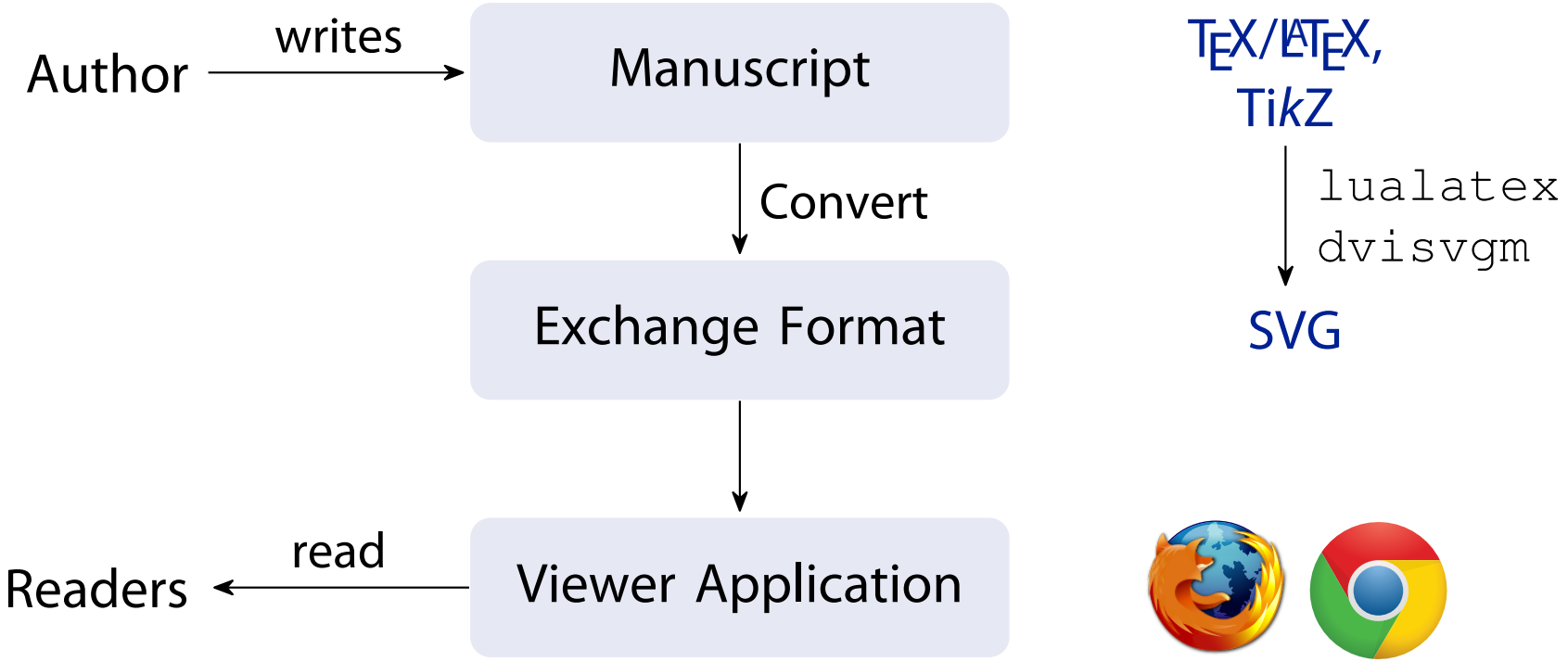
- help seeing and recognizing temporal changes.
- save space.
- lead the attention in presentation.

Animations in Documents: How?



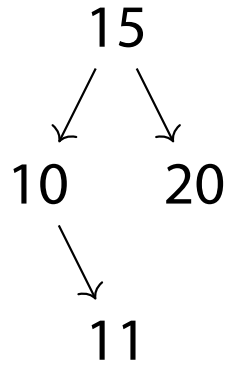
Authors prefer a single, powerful, familiar text processor.
Readers prefer a single, powerful, familiar viewer application.

Animations in Documents: How?

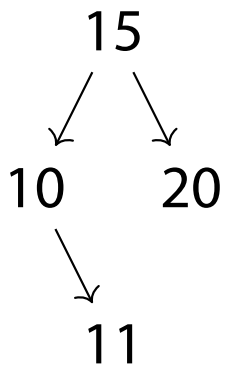


Authors prefer a single, powerful, familiar text processor.
Readers prefer a single, powerful, familiar viewer application.

Graph Drawing in Ti*k*Z

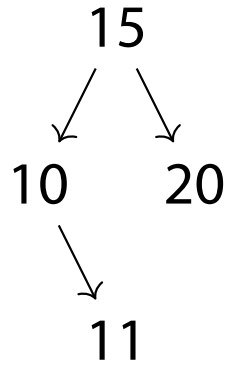



```
\graph[binary tree layout] {
  15 -> {
    10 -> { ,11},
    20}
};
```

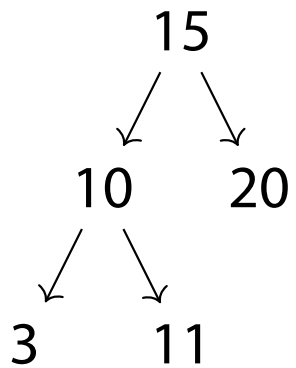



```
\graph[
  animated binary tree layout,
  ...] {
  {[when=1] 15->{10->{ ,11},20}},
  {[when=2] 15->{10->{3,11},20}},
  {[when=3] 15->{10->{3,  } 20}},
};
```

Graph Drawing in Ti*k*Z

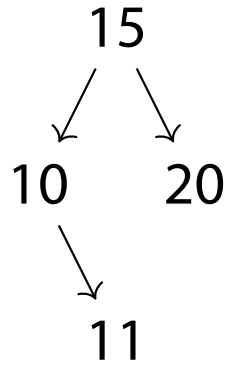



```
\graph[binary tree layout] {
  15 -> {
    10 -> { ,11},
    20}
};
```

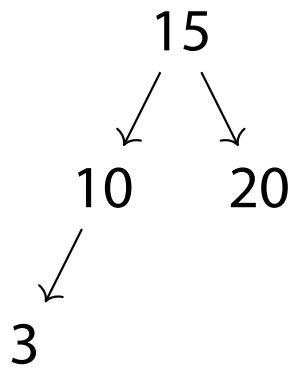



```
\graph[
  animated binary tree layout,
  ...] {
  {[when=1] 15->{10->{ ,11},20}},
  {[when=2] 15->{10->{3,11},20}},
  {[when=3] 15->{10->{3,  } 20}},
};
```

Graph Drawing in Ti*k*Z




```
\graph[binary tree layout] {
  15 -> {
    10 -> { ,11},
    20}
};
```



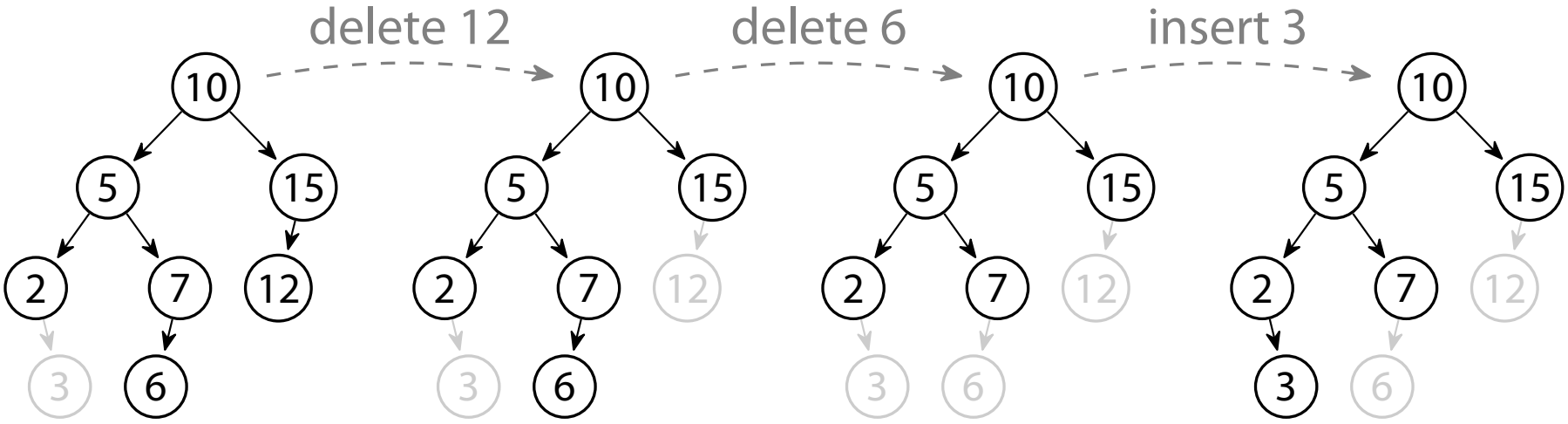

```
\graph[
  animated binary tree layout,
  ...] {
  {[when=1] 15->{10->{ ,11},20}},
  {[when=2] 15->{10->{3,11},20}},
  {[when=3] 15->{10->{3,  } 20}},
};
```

Challenges in Drawing *Dynamic* Graphs

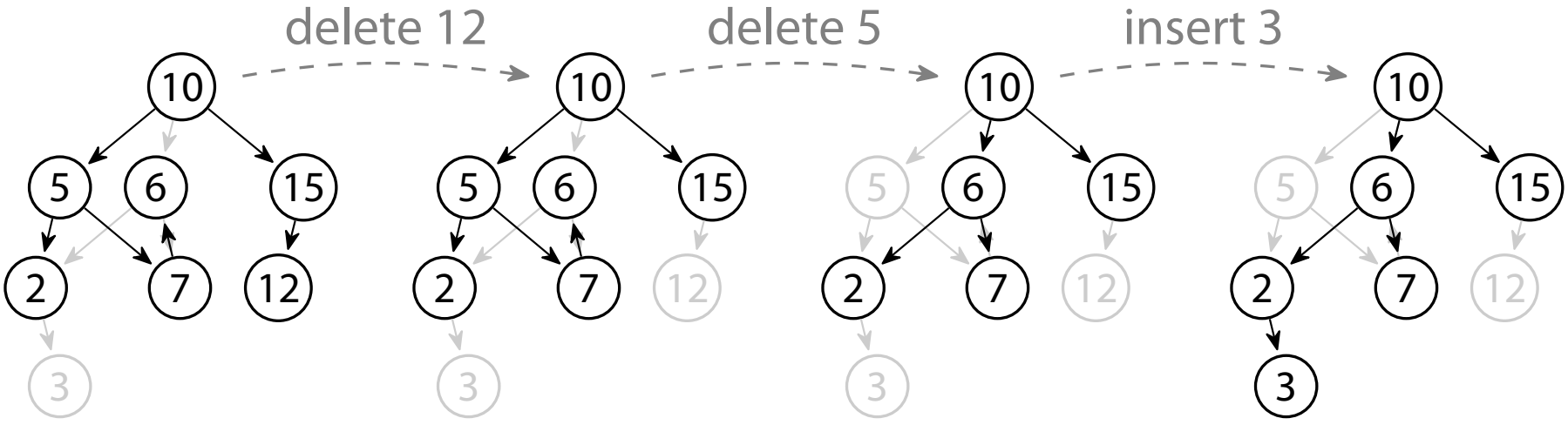
- Consistency: A dynamically drawn graph should be drawn in the same kind at each time.
- Stability:
  - Similar parts should be drawn similarly at all times.
  - No unnecessary movements.
- Existing approaches have severe drawbacks
  - Just drawing the *supergraph* fails even for trees.
  - Online drawing of dynamic graphs produce avoidable movements.

Drawing the Supergraph

Drawing the supergraph requires more space

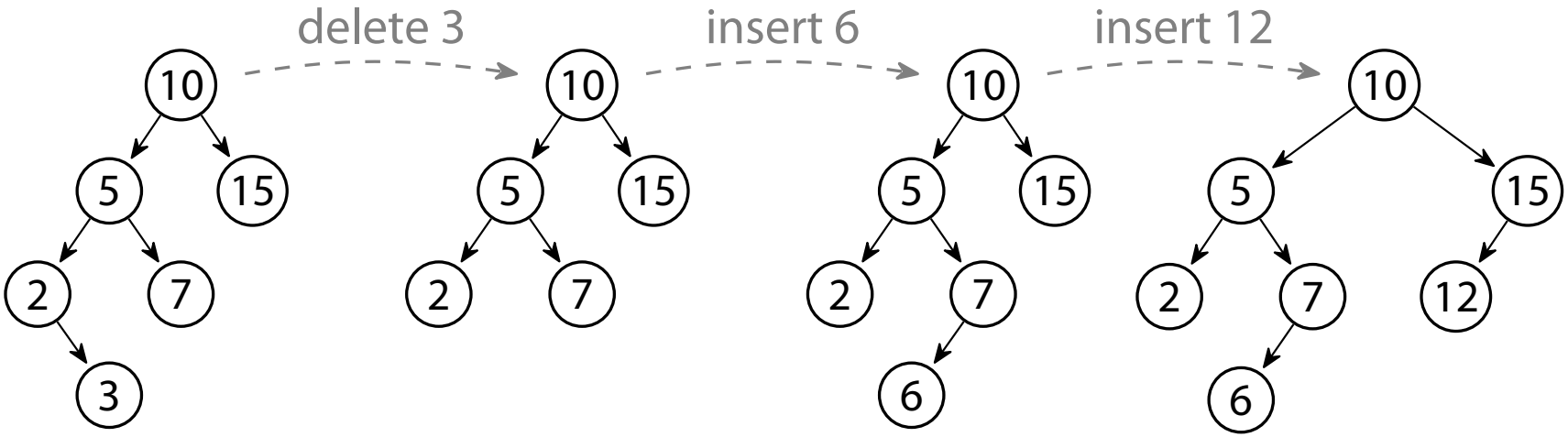


and may obscure the nature of the graph:



Online Drawing of Dynamic Trees

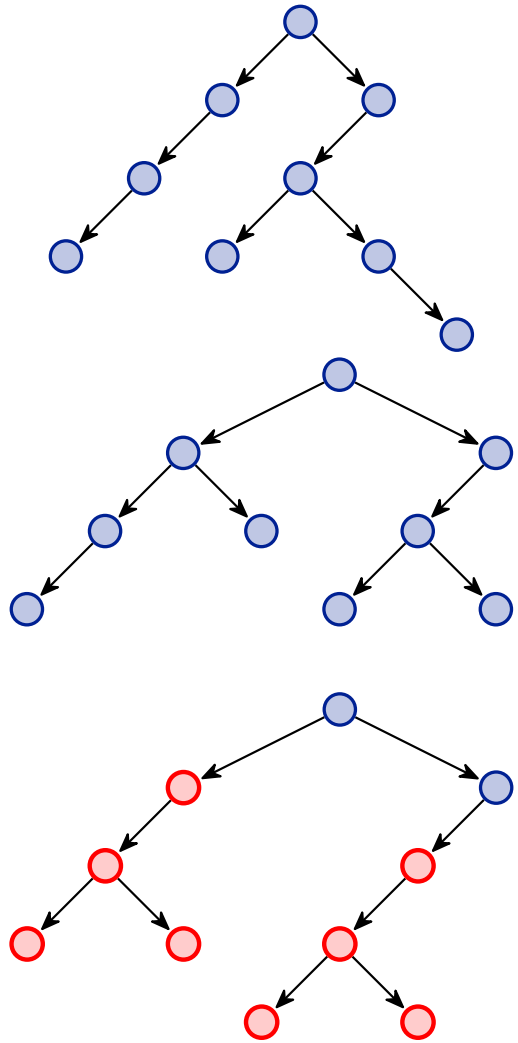
Online layout strategies create each layout only from the current graph and all previous layouts.



Tidy Drawings of Static (Binary) Trees

Aesthetic Criteria

- Ranking
- Ordering (Children from left to right)
- Centering
- Symmetry / Isomorphy



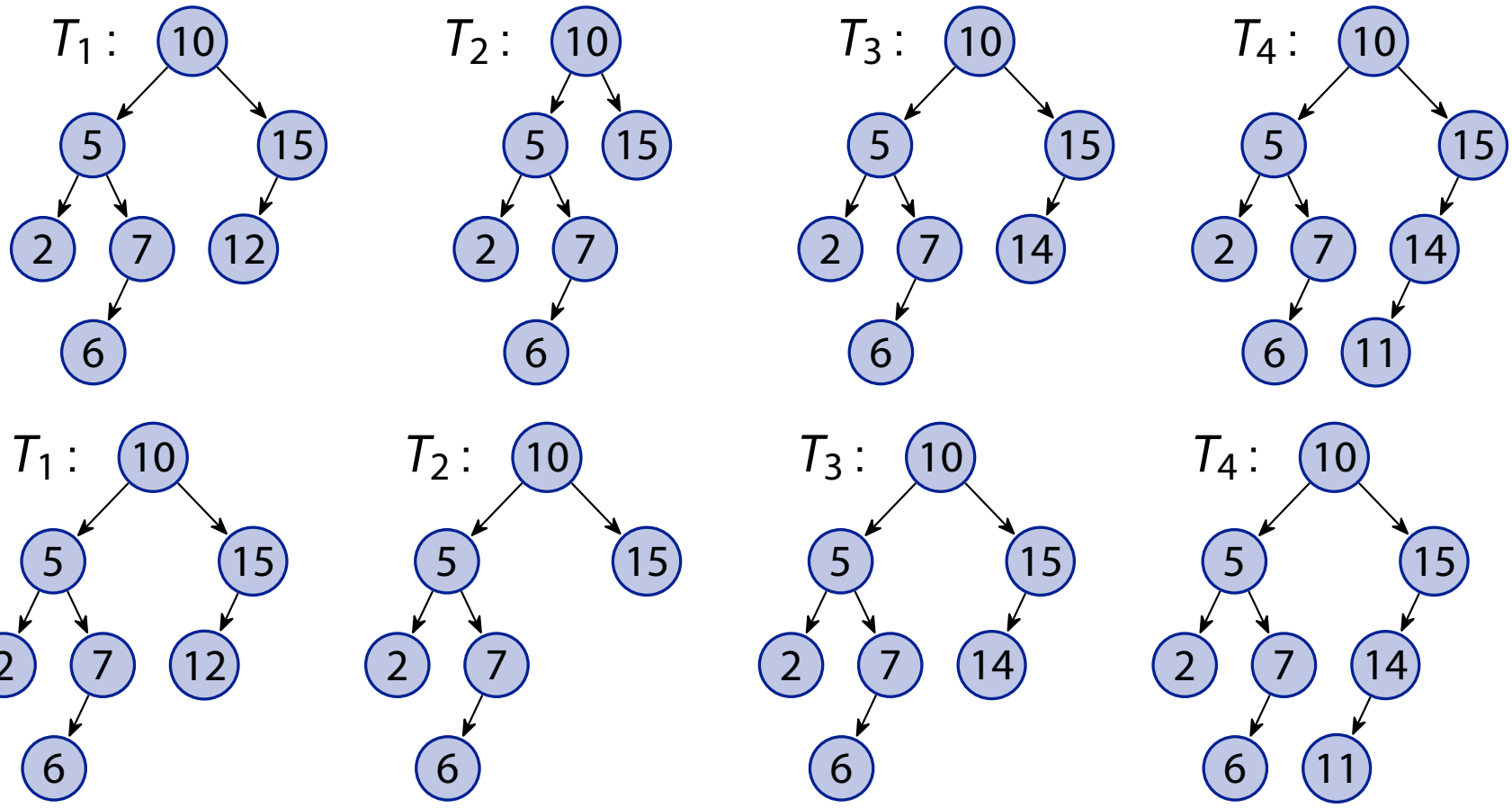
C. Wetherell and A. Shannon (1979).
Tidy Drawings of Trees.
IEEE Trans. on Softw. Eng., 5(5):514–520

E. M. Reingold and J. S. Tilford. (1981).
Tidier Drawings of Trees.
IEEE Trans. on Softw. Eng., 7(2):223–228

Stability

A New Criterion

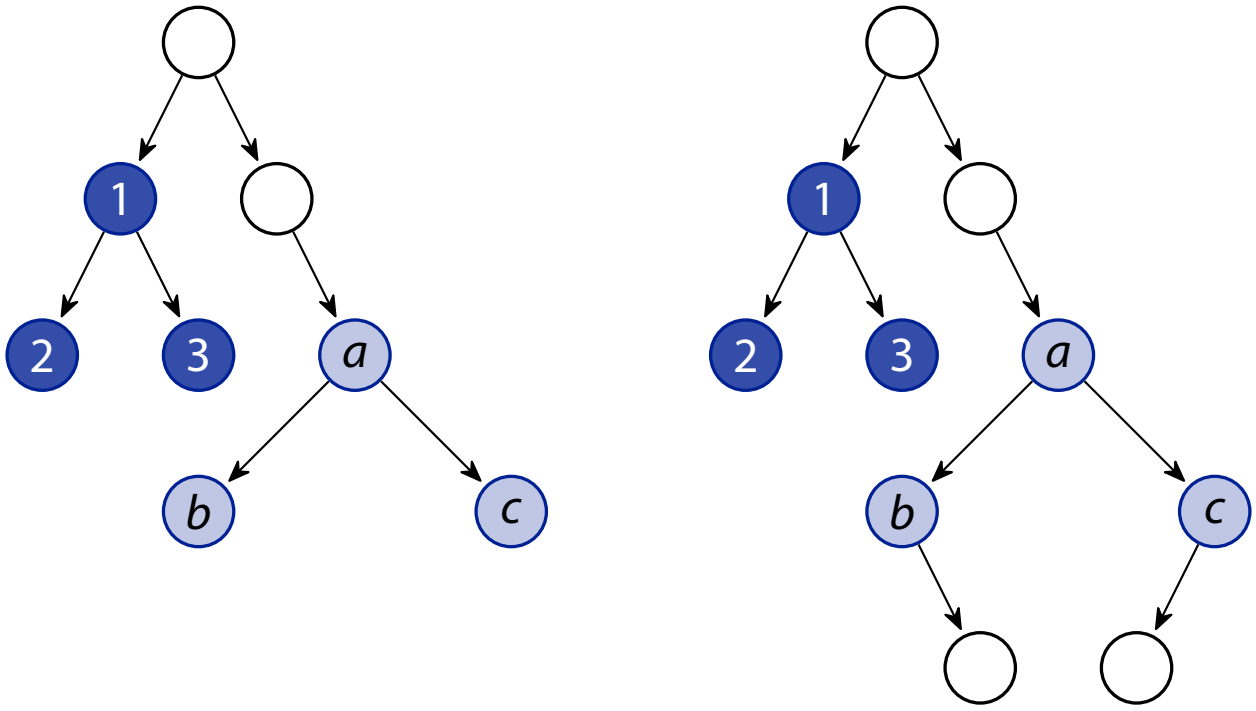
Child nodes may not change their offset before changing their parent or ordering position.



A Weaker Criterion for Symmetry

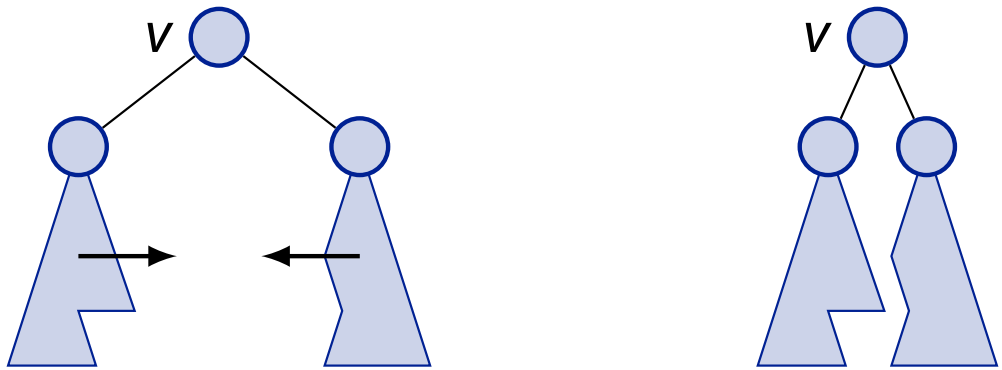
Weak Symmetry

Subtrees that are symmetric or isomorphic over time should have symmetric or the same layouts in each layout.



Drawing Static Trees

The Algorithm of Reingold and Tilford



Divide and conquer at root node v

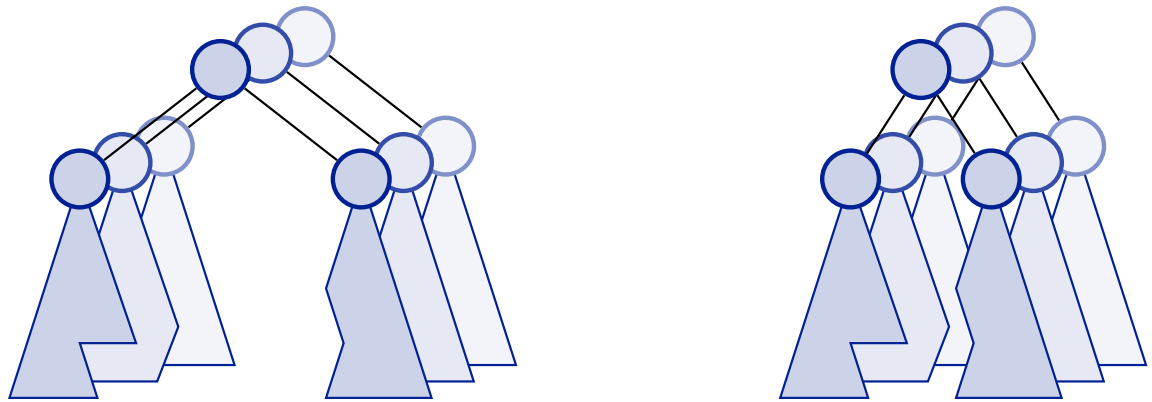
Divide Determine the layouts of the subtrees.

Conquer Make sublayouts tight.
Center the root node v above its children.

Runtime: $O(|V|)$

An Offline Algorithm for Drawing Dynamic Trees

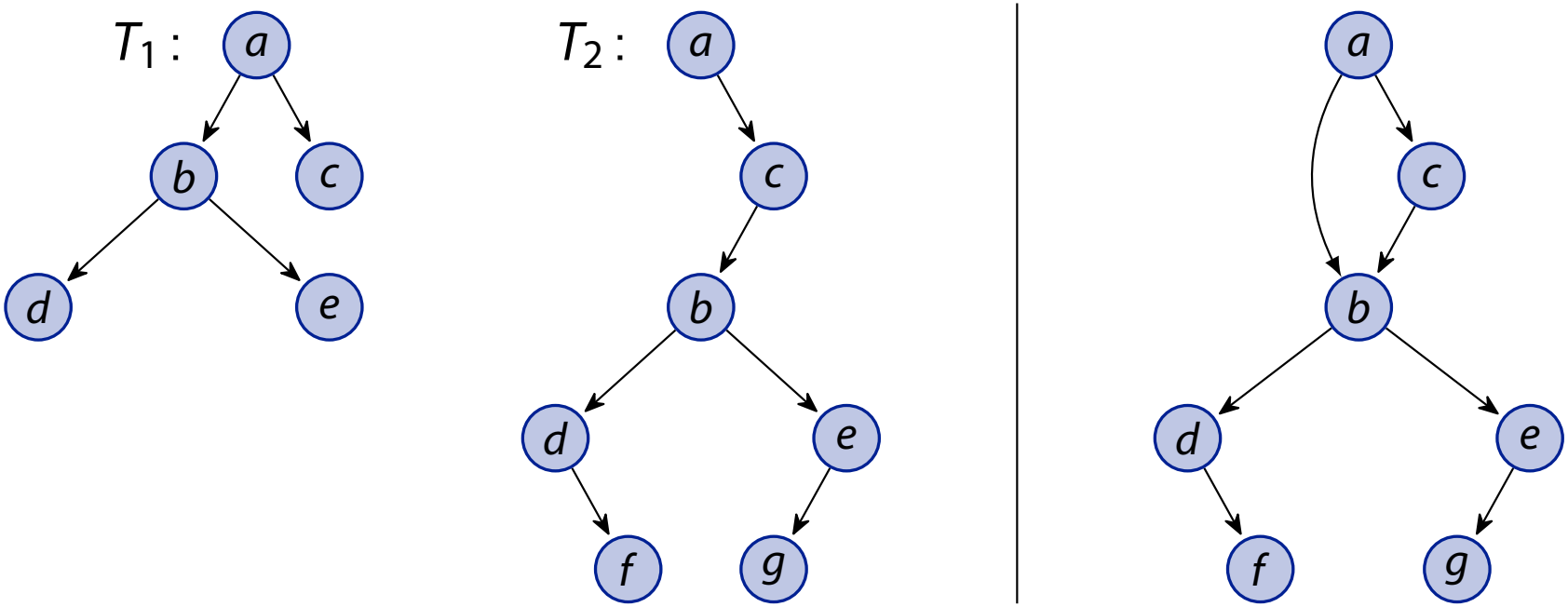
Case 1: The Supergraph is a Tree



- When the supergraph is a tree, nodes keep their parents over time.
- Do divide and conquer as in the static algorithm:
  1. Compute a layout for all subtrees of a node in all T_i.
  2. Compute the minimal distance of subtrees in T_i.
  3. Use the maximum distance in all T_i simultaneously.
- Runtime $O(|T| \cdot |V|)$

An Offline Algorithm for Drawing Dynamic Trees.

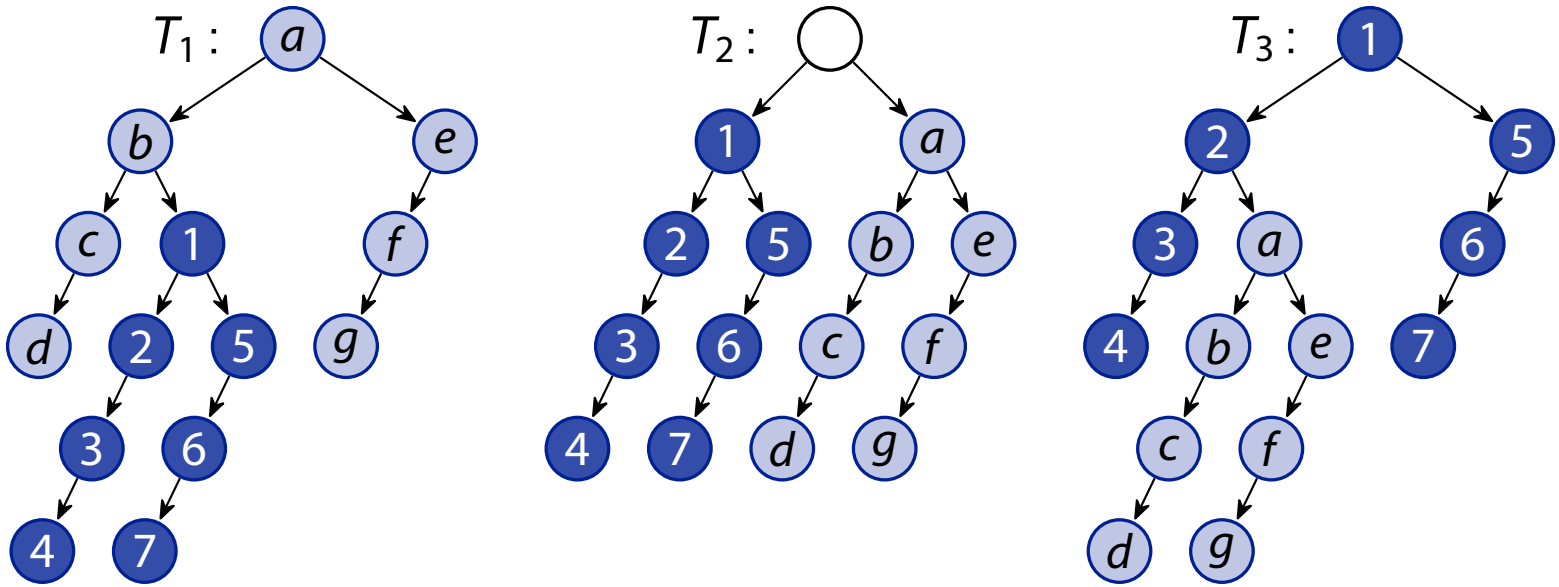
Case 2: The Supergraph is a DAG



- When nodes change parents over time, the supergraph is not a tree.
- But if it is at least acyclic, there exists a *topological ordering* of the nodes.
- Use the ordering to process the nodes as before.

An Offline Algorithm for Drawing Dynamic Trees.

Case 3: The Supergraph is Cyclic.



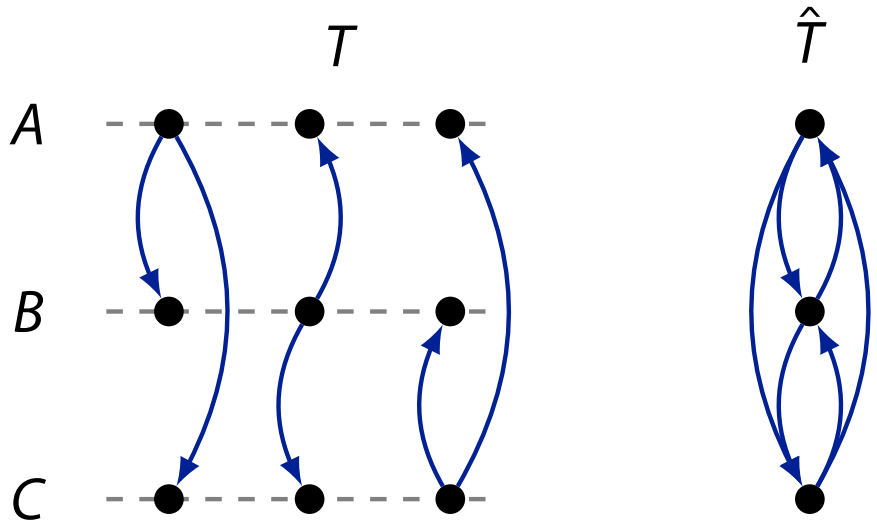
Lemma

No drawing of $T = (T_1, T_2, T_3)$ meets all of the criteria Ranking, Ordering, Centering, Weak Symmetry, and Stability.

Making the Supergraph Acyclic

Breaking Cycles

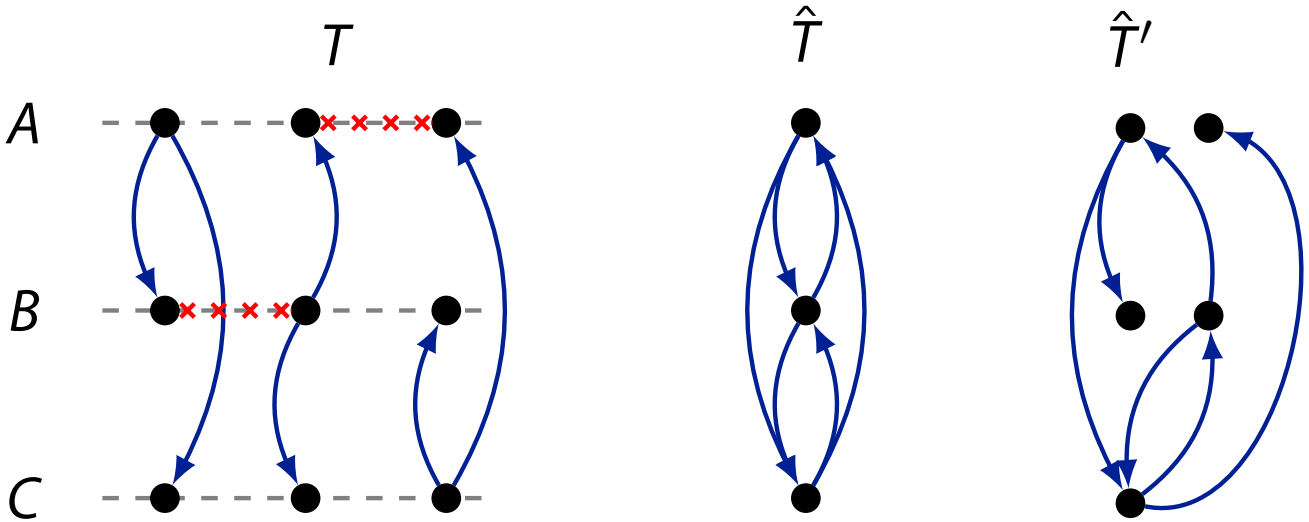
Renaming nodes at certain times can break the cycles.



Making the Supergraph Acyclic

Breaking Cycles

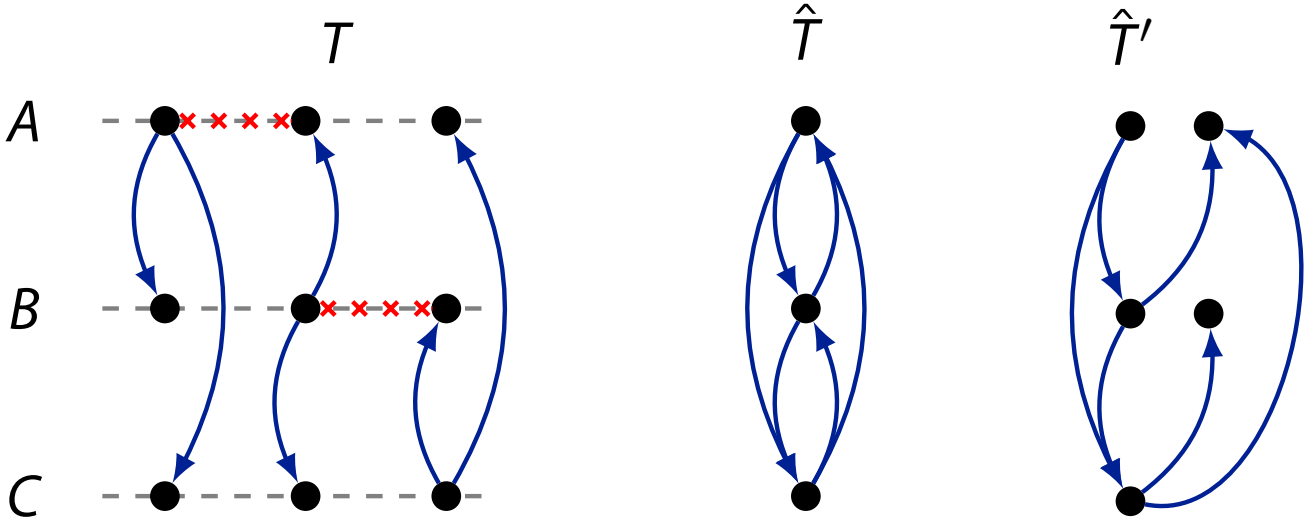
Renaming nodes at certain times can break the cycles.



Making the Supergraph Acyclic

Breaking Cycles

Renaming nodes at certain times can break the cycles.



Finding Temporal Cuts Is Hard

Temporal-Cut-Minimization

Input A dynamic graph G

Output A minimal number of temporal cuts that make G's supergraphs acyclic. that the resulting dynamic tree is acyclic.

Theorem

The decision version of Temporal-Cut-Minimization is NP-complete.

Conclusion

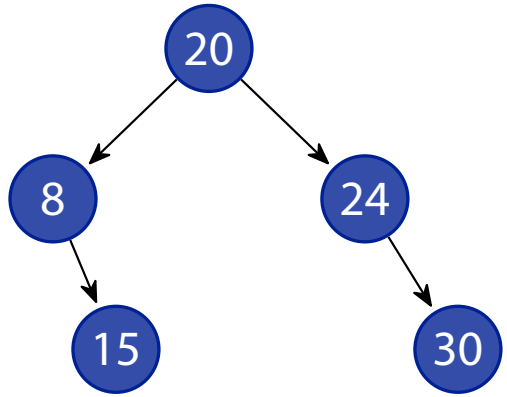
Dynamic graph drawing in Ti*k*Z allows you to

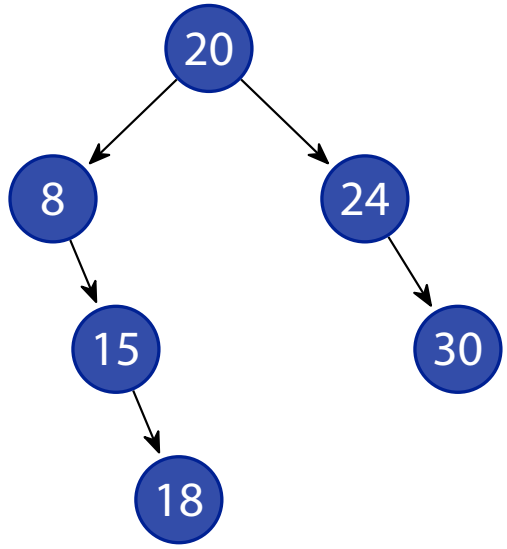
- integrate dynamic graphs in presentations and research papers,
- visualize sequences of trees.

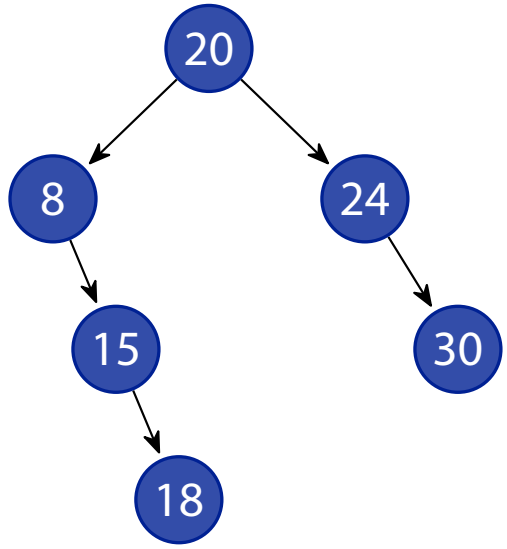
Offline drawing of dynamic graphs deserves further research:

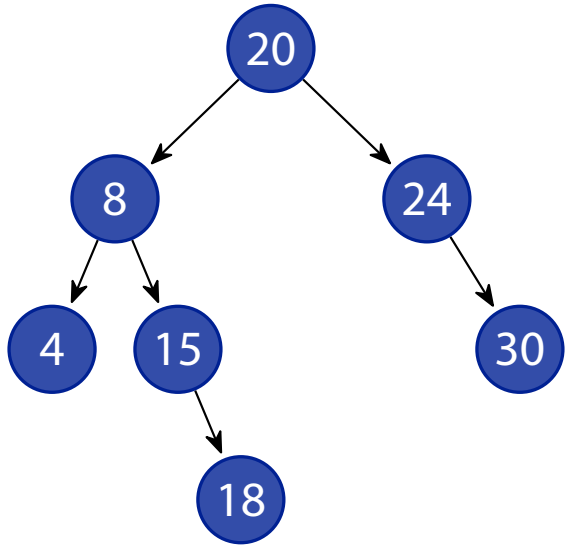
- There are "desirable" and "undesirable" movements in drawings. How do we identify them algorithmically?
- Trees are simple. What are aesthetic criteria and algorithms for series-parallel or layered or planar graphs?
- Even for trees NP-hard problem arise. What about more complicated graph classes?

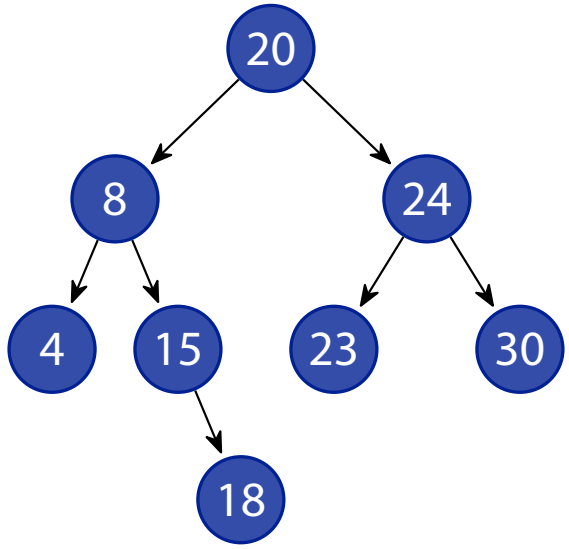
The library is available as part of (the development version of) Ti*k*Z.

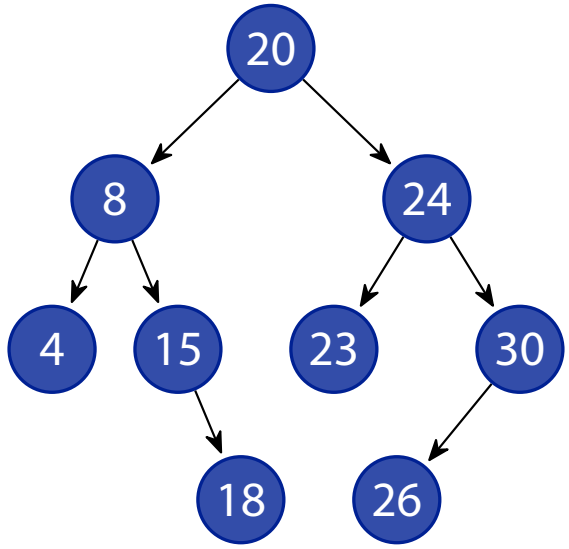

20
8
24
15
30

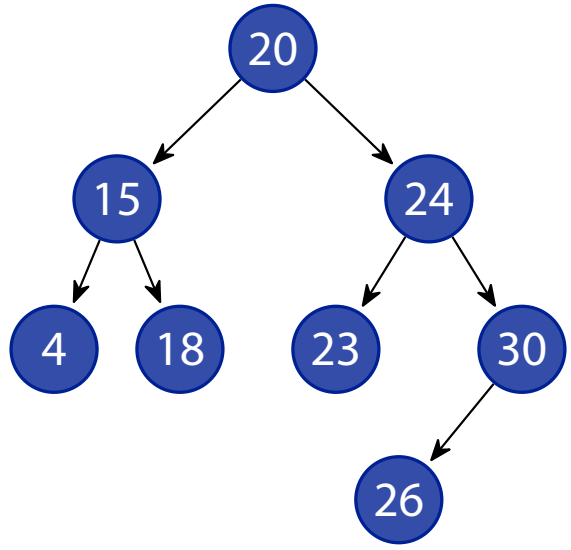

20
8
24
15
30
18

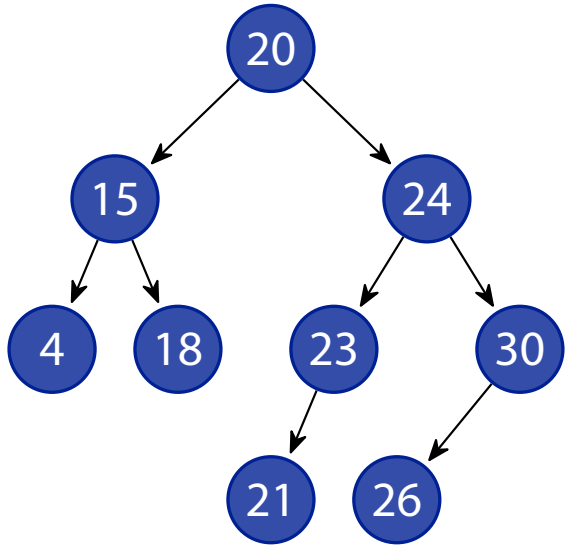

20
8
24
15
30
18

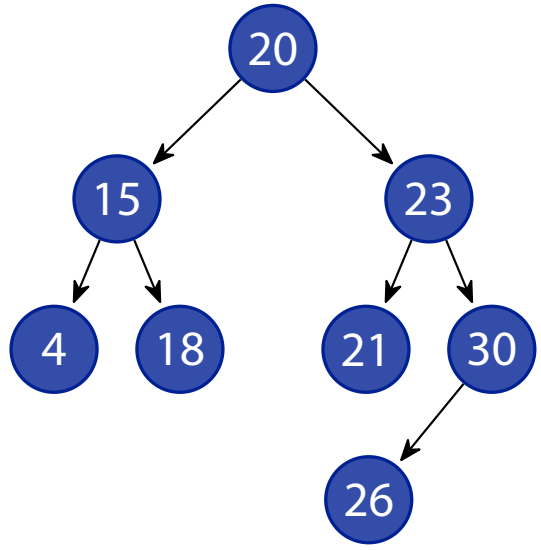

20
8
24
4
15
30
18

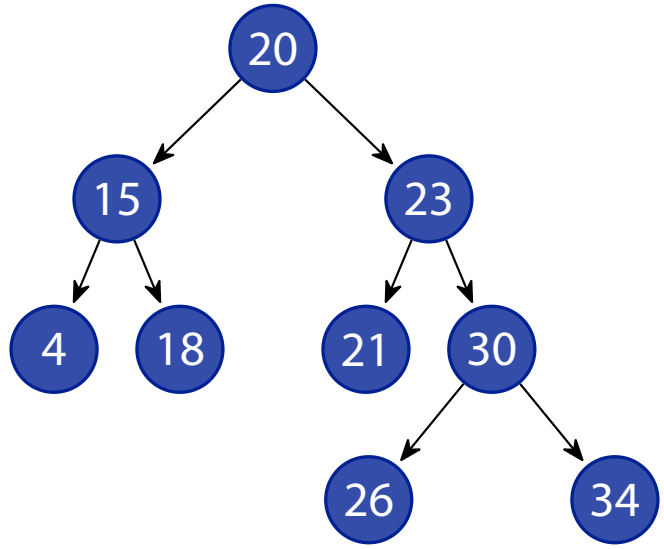

20
8
24
4
15
23
30
18

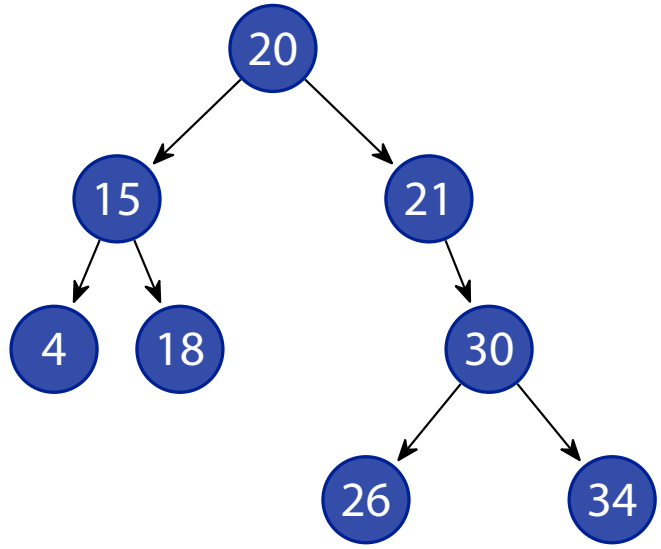

20
8
24
4
15
23
30
18
26

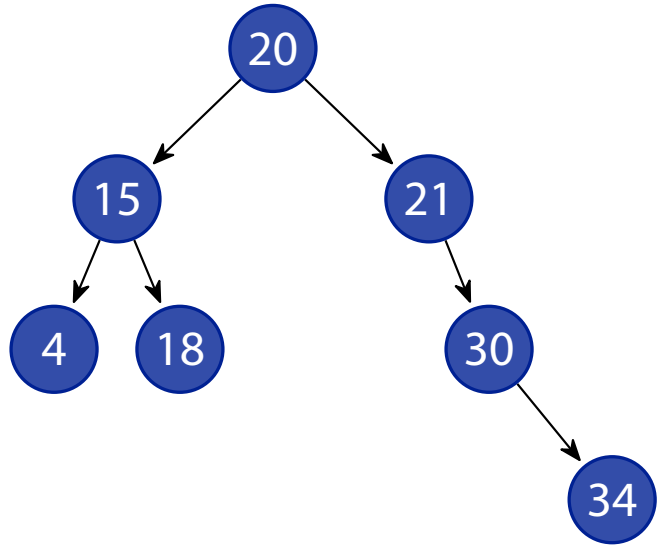

20
15
24
4
18
23
30
26

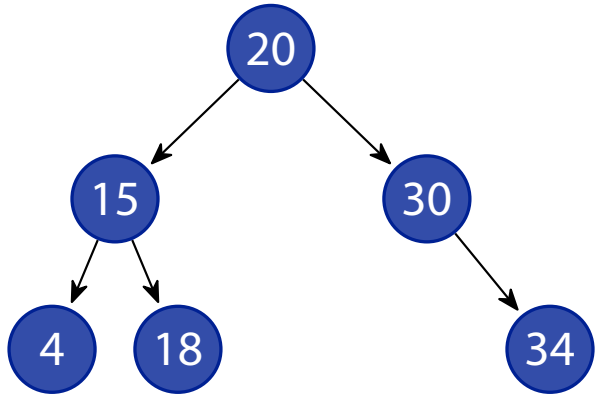

20
15
24
4
18
23
30
21
26

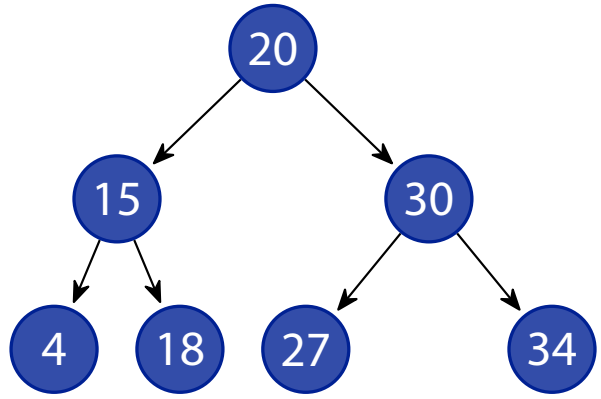

20
15
23
4
18
21
30
26

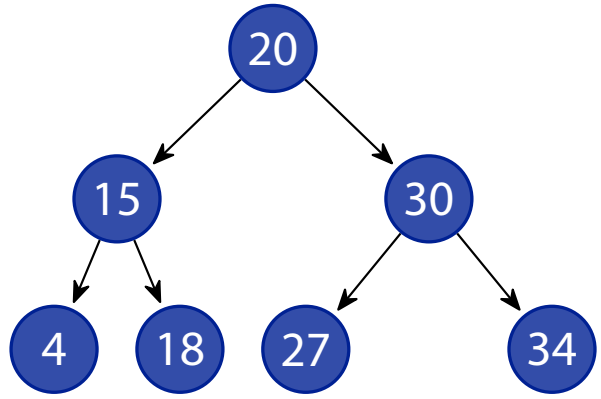

20
15
23
4
18
21
30
26
34

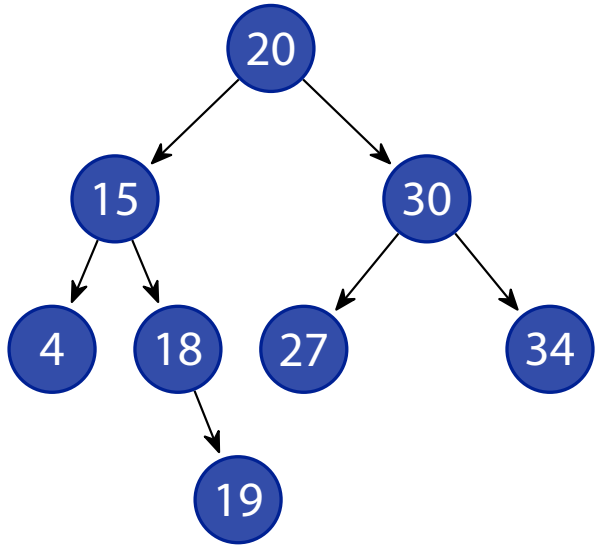

20
15
21
4
18
30
26
34

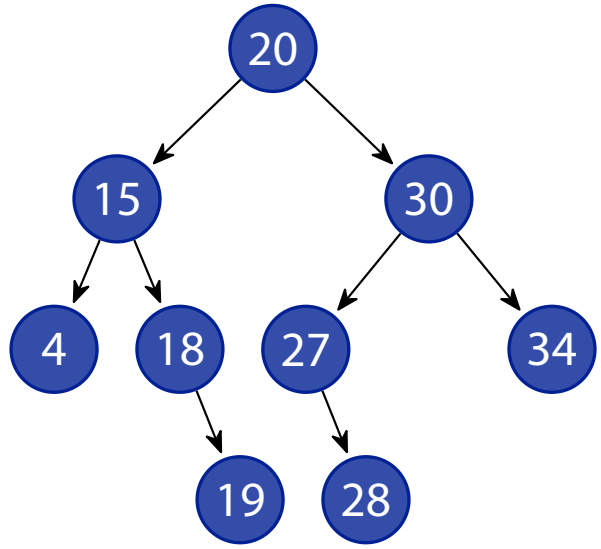

20
15
21
4
18
30
34

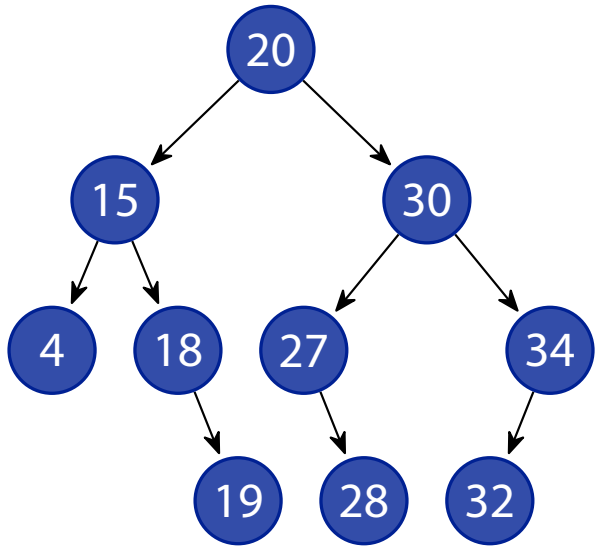

20
15
30
4
18
34

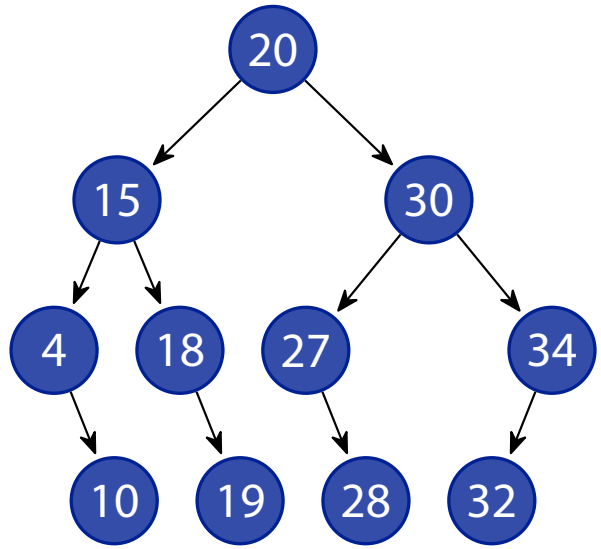

20
15
30
4
18
27
34


20
15
30
4
18
27
34


20
15
30
4
18
27
34
19


20
15
30
4
18
27
34
19
28


20
15
30
4
18
27
34
19
28
32


20
15
30
4
18
27
34
10
19
28
32