

Unique Rectification Targets in d -Complete Partial Orders

Rahul Ilango

Mik Zlatin

Work supported by the Rutgers Math Department

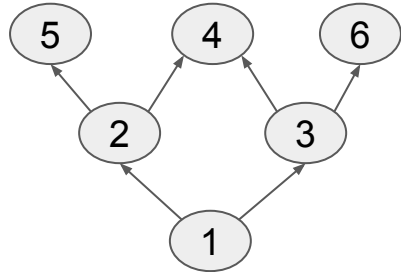
Background

Motivation

- K Theory and geometric spaces
 - associates a number system (ring) with a geometric space
 - Ring illuminates properties of the geometric space
- Our project helps determine multiplication in these number systems

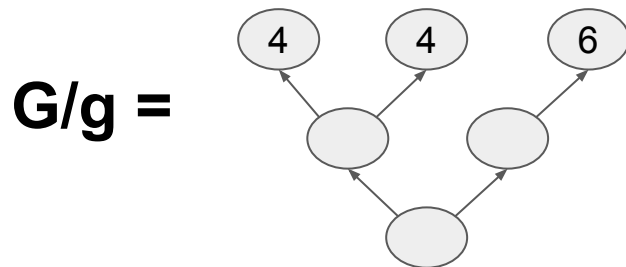
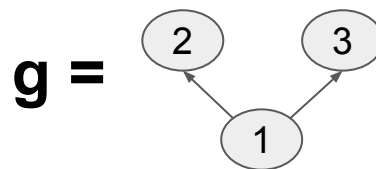
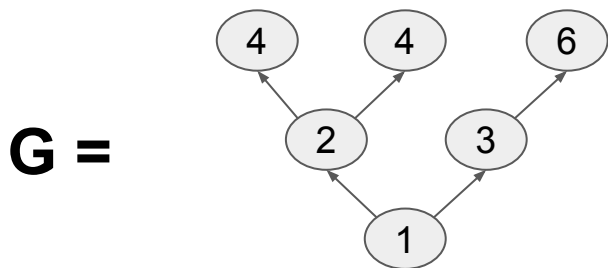
Partially Ordered Sets (Posets / shapes)

- Assigns hierarchy to elements in sets
- Ancestors of node are “less than” it



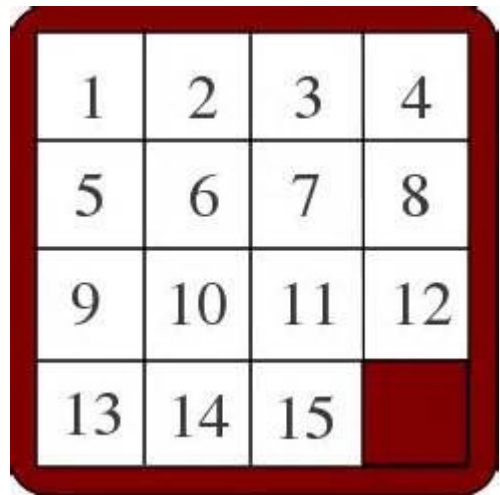
Skew shapes

- Remove downwardly closed subset



Rectification

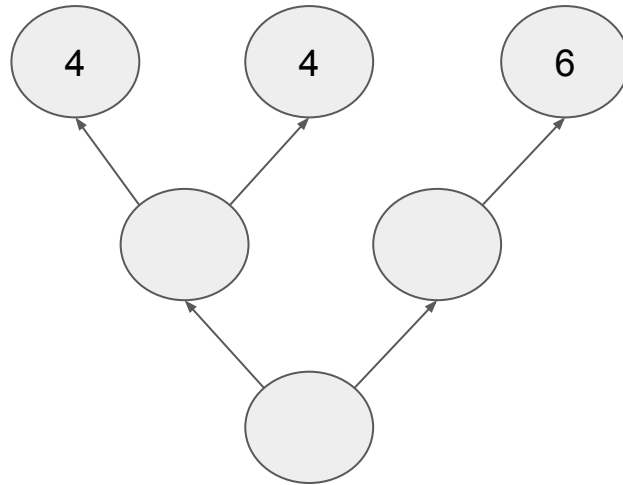
- Process to turn a skew shape into a poset of straight shape (an order ideal)
- Algorithm is called Jeu De Taquin - tile game
 - Dot an empty node which is an “inner corner”
 - Inner corner means all of a node's children are filled
 - Swap dot with smallest child (if tie, swap both)
 - Delete dot node when leaf
 - Repeat until no more empty nodes



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

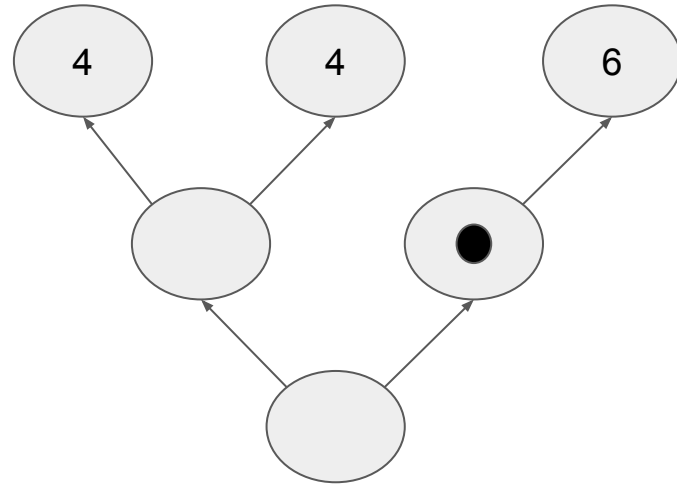
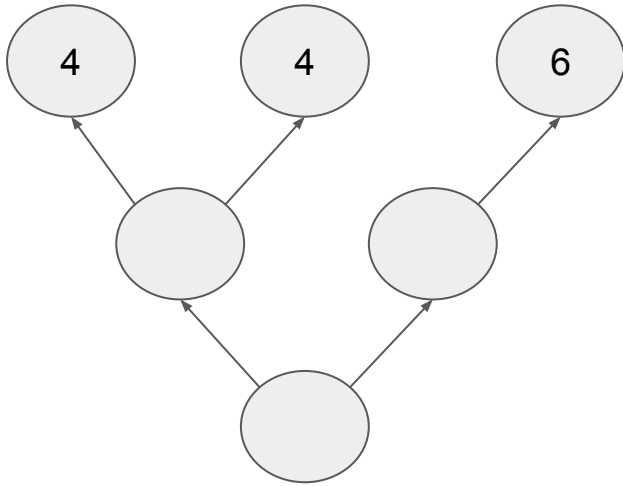
Rectification Example

- Begin with skewed graph



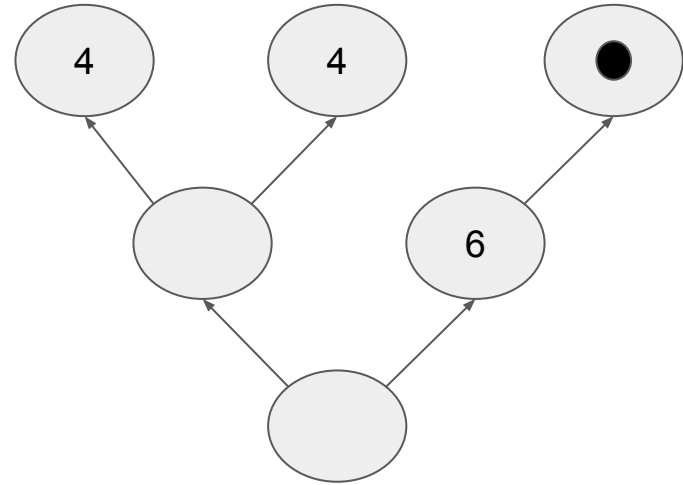
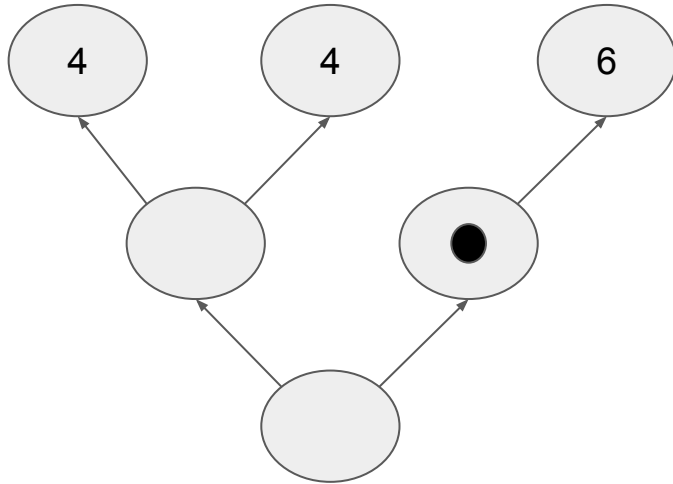
Rectification Example

- Fill external empty node with Dot
 - External = Node with children all filled
 - We have choice here



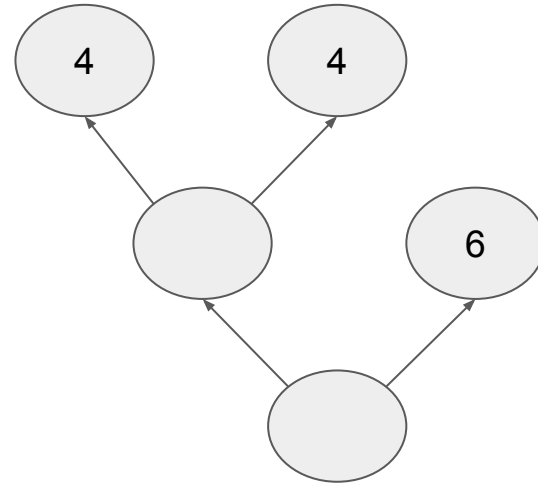
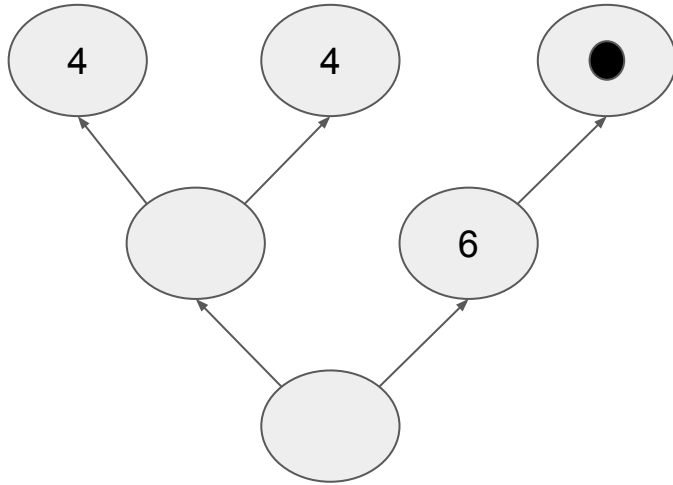
Rectification Example

- Swap dot with smallest child



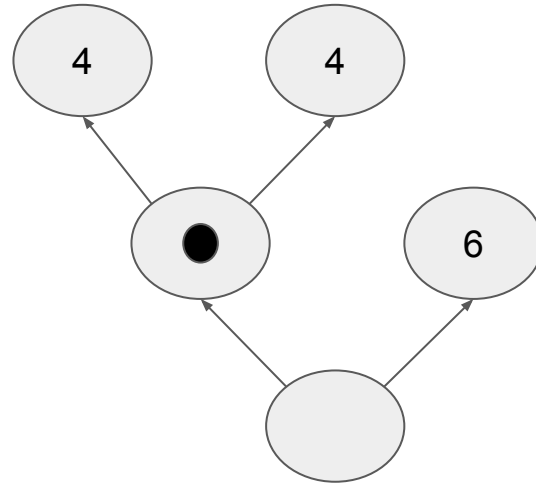
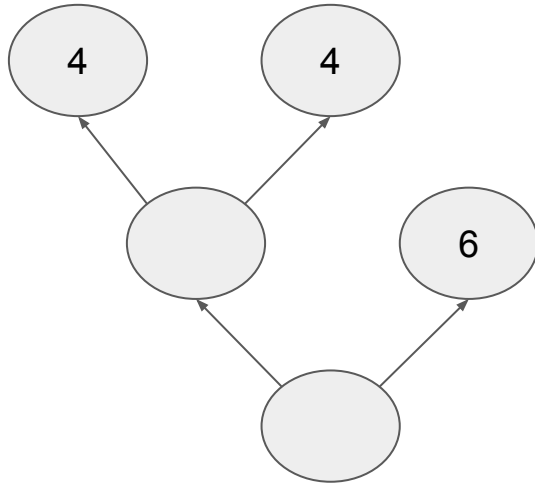
Rectification Example

- Delete dot with no children



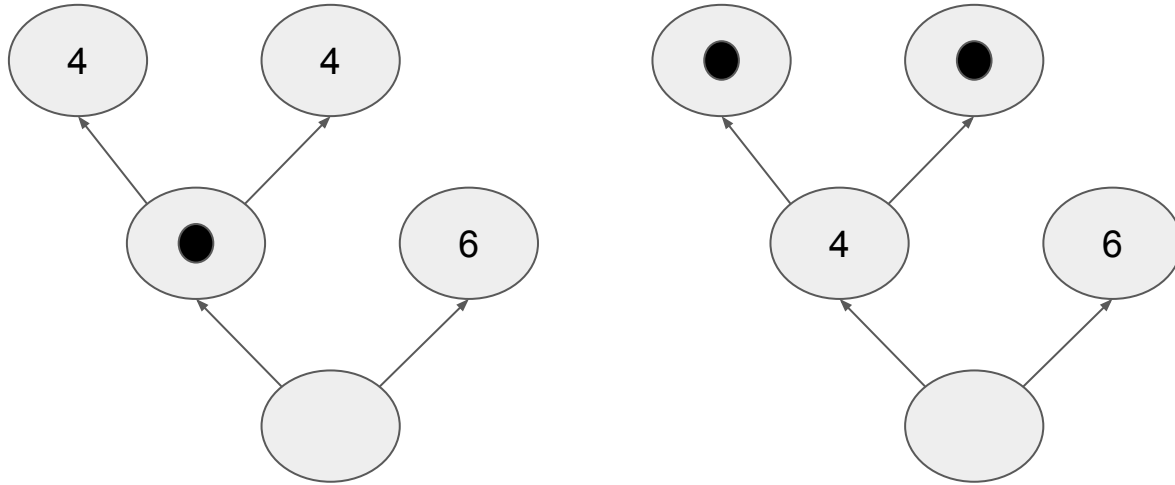
Rectification Example

- Add dot to inner corner



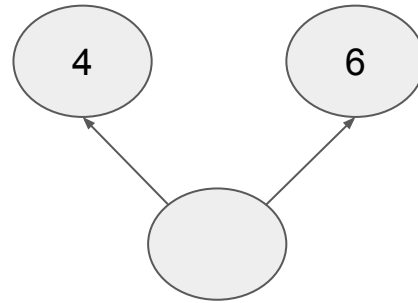
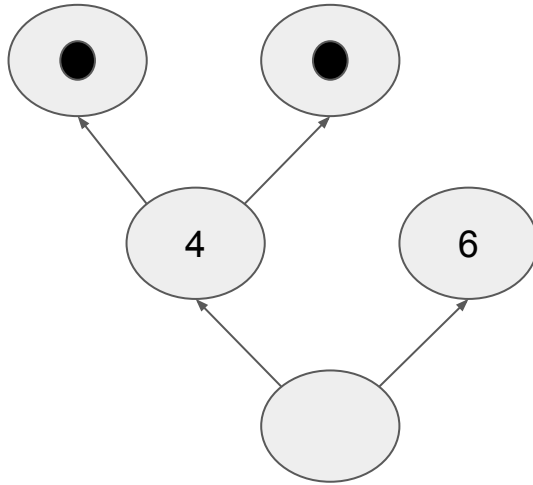
Rectification Example

- Swap dot with smallest child (if tie, swap both)



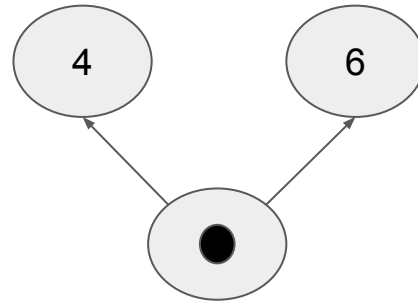
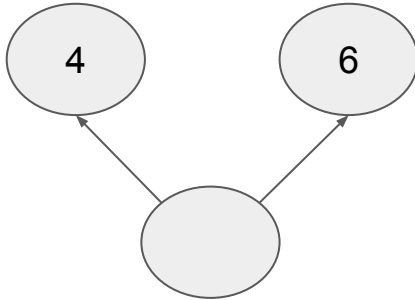
Rectification Example

- Delete dots with no children



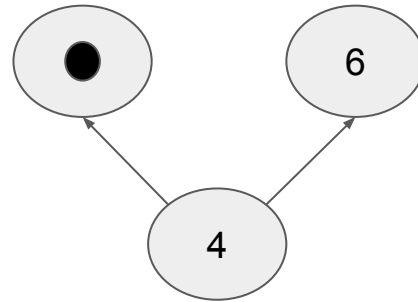
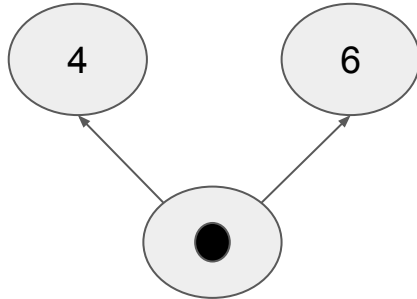
Rectification Example

- Add dot to inner corner



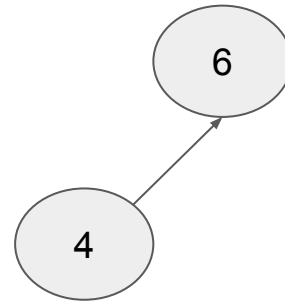
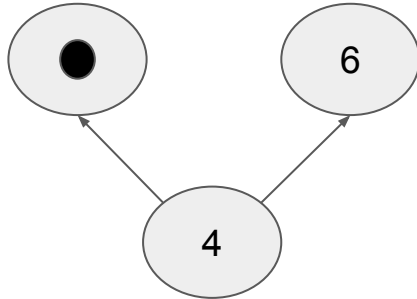
Rectification Example

- Swap dot with smallest child



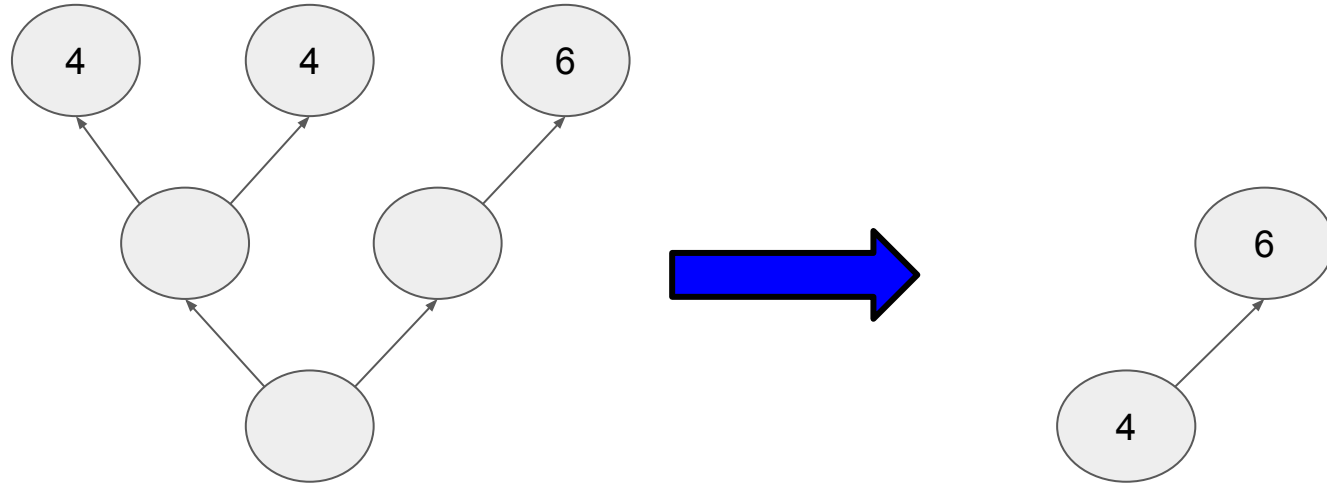
Rectification Example

- Delete dot with no children



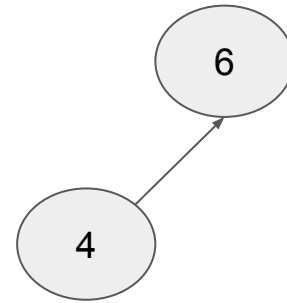
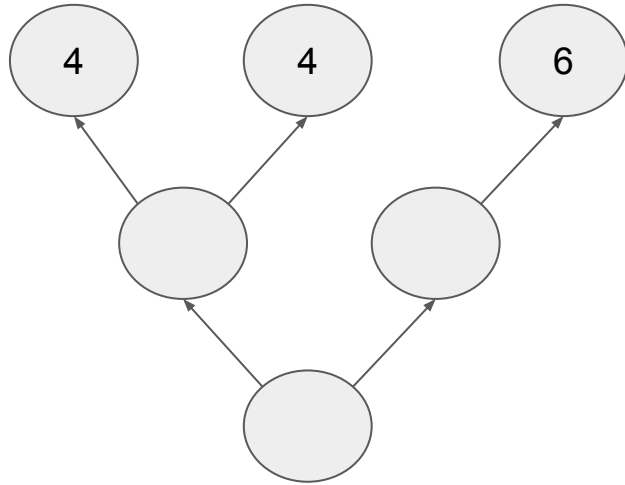
Rectification Example

- Done!

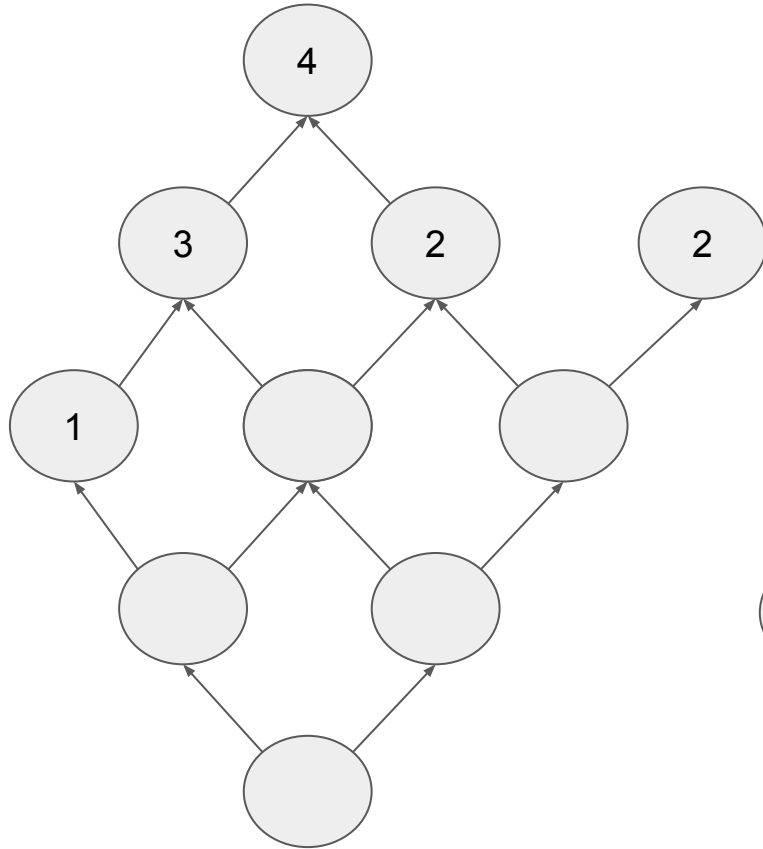


Rectification Example

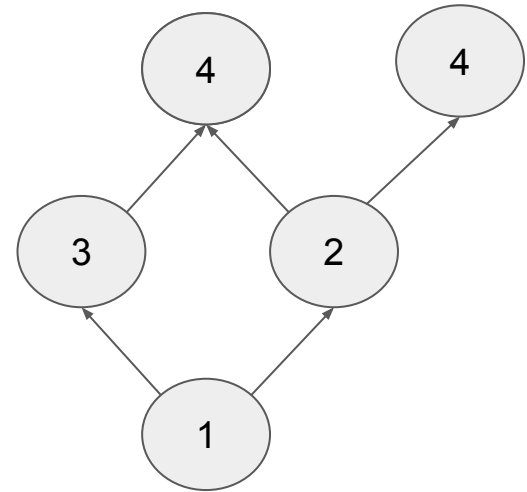
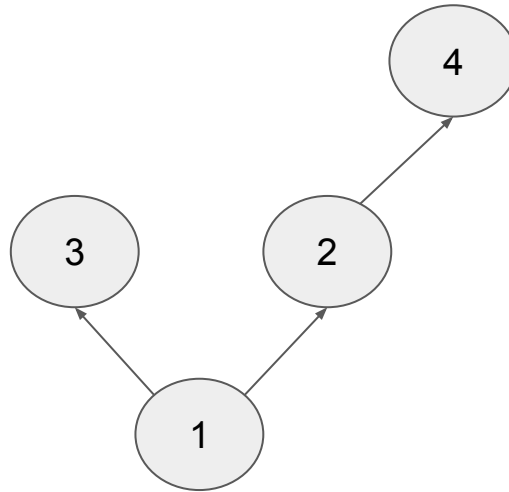
- Did our choices matter?



Order Matters?



- Didn't matter that time
- In general can matter



Objective

Definition

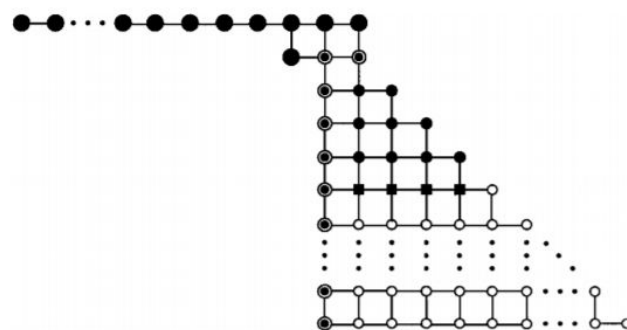
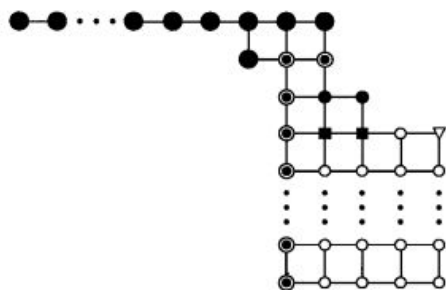
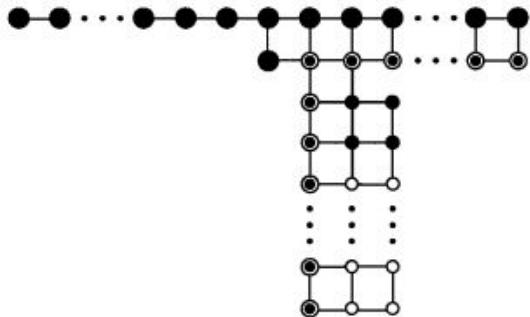
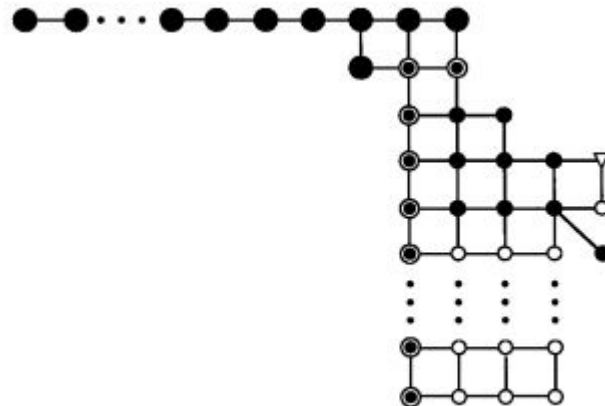
A Unique Rectification Target (URT) is a labeling T of an order ideal such that if some labeled skew poset rectifies to T , it will always rectify to T no matter the choices you make during Jeu de Taquin.

Research Goal

Investigate the existence of URTs on a specific family of posets (d-complete Posets)

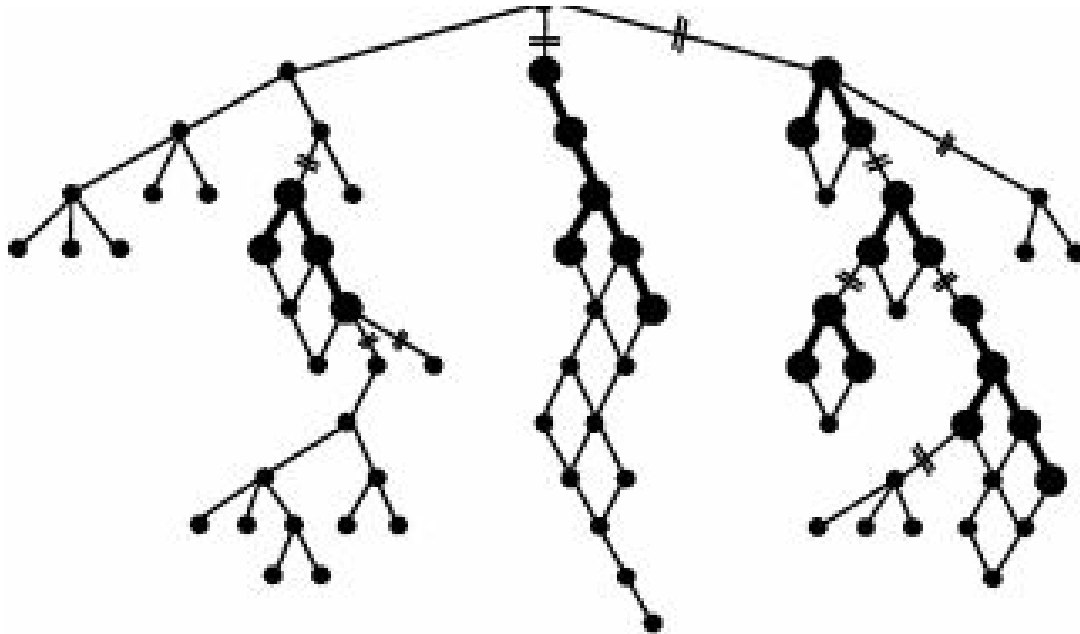
d-Complete posets

- Dictated by local constraints
- Composed of slant sum of “irreducible” posets



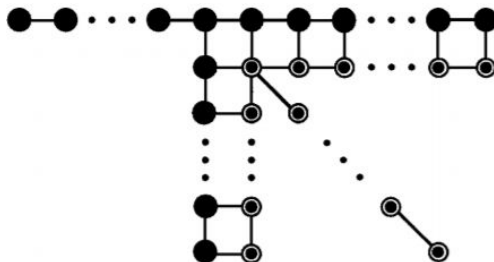
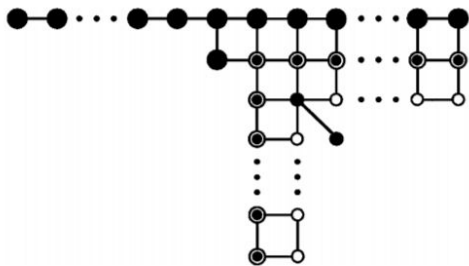
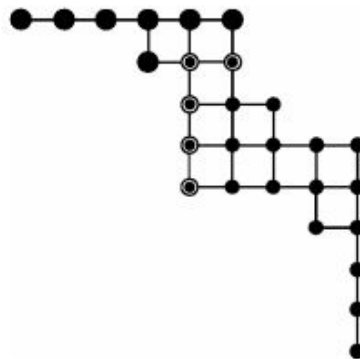
Slant sum

- Can attach d-Complete posets to each other at “acyclic” nodes



Why d-complete?

- Lambda-minuscule varieties
- Cohomology / standard labeling
- Seem to hold property empirically



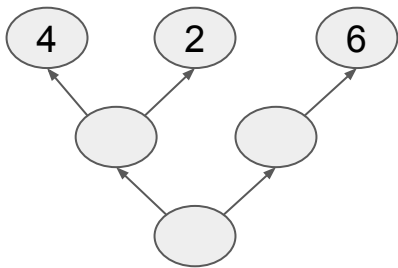
Ideas

- Look at URTs in each irreducible family
- See what we can say about how the slant sum operation preserves rectifications
- Computer

Results

Trees

- In some sense, the “simplest” d-complete posets
- We prove that everything is a unique rectification target in trees



Computer Program

- Wrote computer program to find URTs in given posets
- Faster version checks for just minimal URTs

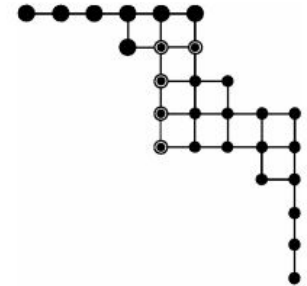
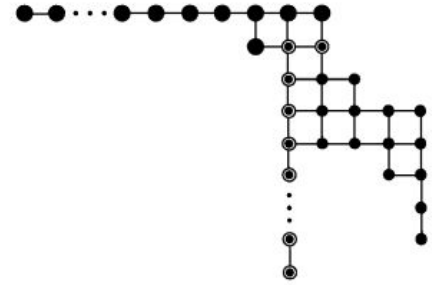
```
workspace: java - IntelliJ IDEA
File Edit Source Refactor Navigate Search Project Run Window Help
workspace: java - IntelliJ IDEA
1 import java.util.ArrayList;
2
3 public class Graph {
4
5     //Note: The zero element in the adjacency list MUST be the minimal one!
6     Integer [][] adjacencyList;
7     ArrayList<ArrayList<Integer>> parentList; //list of all the parents of a node
8
9     Graph (Integer[][] adjacencyList){
10         this.adjacencyList = adjacencyList;
11         setParentList();
12     }
13
14     //populates parentList with the corresponding parents of each node
15     void setParentList(){
16         //create empty arraylist of nodes
17         parentList = new ArrayList<ArrayList<Integer>>();
18         for (int v = 0; v < adjacencyList.length; v++){
19             parentList.add(new ArrayList<Integer>());
20         }
21
22         //populate with proper parents
23         for (int v = 0; v < adjacencyList.length; v++){
24             for (Integer c: adjacencyList[v]){
25                 parentList.get(v).add(c);
26             }
27         }
28     }
29
30 }
31
32
33 //MAY USE LOTS OF MEMORY, AVOID THIS
34 ArrayList<Ideal> allIdeals(){
35     IdealIterator it = new IdealIterator(this);
36     ArrayList<Ideal> ideals = new ArrayList<Ideal>();
37     while (true){
38         Ideal i = it.next();
39         if (i == null) break;
40         ideals.add(i);
41     }
42     return ideals;
43 }
44
45
46 HashSet<ArrayList<Integer>> minimalFillings(){
47     ArrayList<Ideal> ideals = allIdeals();
48     HashSet<ArrayList<Integer>> minFillings = new HashSet<ArrayList<Integer>>();
49     for (Ideal ideal : ideals){
50         minFillings.add(ideal.minimalFilling());
51     }
52     return minFillings;
53 }
54
55
56 Integer idealCount(){
57     int i = 0;
58     IdealIterator it = new IdealIterator(this);
59     while (it.next() != null){
60         i += 1;
61     }
62     return i;
63 }
```

Adding “tails”

How does adding tails to elements affect a poset’s URTs?

- Adding below a minimum element preserve URTs
- Adding a new minimum below multiple posets preserves URTs
- Attaching posets to the nodes of a tree preserves URTs

- Adding above elements does not preserve URTs in general

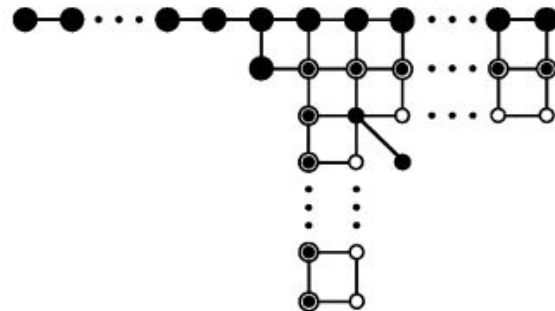
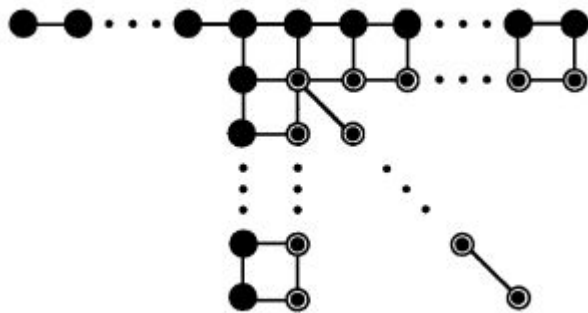
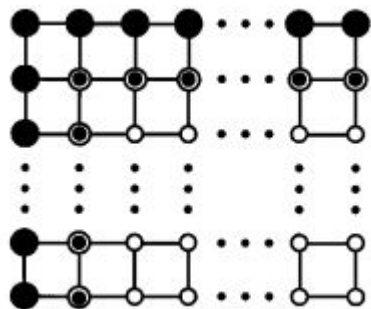


Slant Sum

- Adding above does not preserve all URTs
- We define p -chain URTs, which are exactly the URTs which are preserved when adding tails above
- p -chain URTs are preserved in slant sums (at a point p)

p-chain URTs in d-complete posets

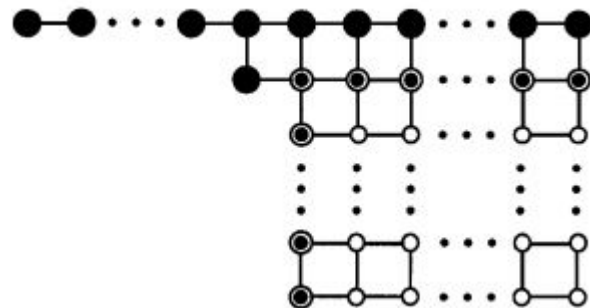
- 15 irreducible components
- 10 of these can be slant summed onto
- For 5 of these, we have shown that a poset is a URT in that shape if and only if it is a p-chain URT



Current Work

URT's in irreducible poset

- Work so far has concentrated on gluing together posets and d-complete posets in general
- Now we will focus in on individual posets and investigate their URTs and p-chain URTs
- Identify “reading word” which is slide invariant



Sources

Buch, A. S., Samuel, M. J. (2016). K-theory of minuscule varieties. Journal für die reine und angewandte Mathematik (Crelles Journal), 2016(719), 133-171.

Pechenik, O. A. (2016). K-theoretic Schubert calculus and applications (Doctoral dissertation, University of Illinois at Urbana-Champaign).

Proctor, R.A (1999). Dynkin Diagram Classification of λ -Minuscule Bruhat Lattices and of d-Complete Posets. Journal of Algebraic Combinatorics 9: 61.

Proctor, R.A (1999). Minuscule elements of Weyl groups, the numbers game, and d-complete posets. Journal of Algebra 213: 272-303

Thomas H., Yong A (2009). A jeu de taquin theory for increasing tableaux, with applications to K-theoretic Schubert calculus. Algebra Number Theory 3, no. 2, 121--148.

Acknowledgement

Work supported by the Rutgers Math Department