Lattice Drawing

Survey of Approaches, Geometric Method and Existing Software

Jan OUTRATA

Dept. Computer Science, Palacký University, Olomouc, Czech Republic

Binghamton University—SUNY, Binghamton, NY, USA (recreation member)

Invited lecture

SSIE Dept., T. J. Watson School, Binghamton University—SUNY

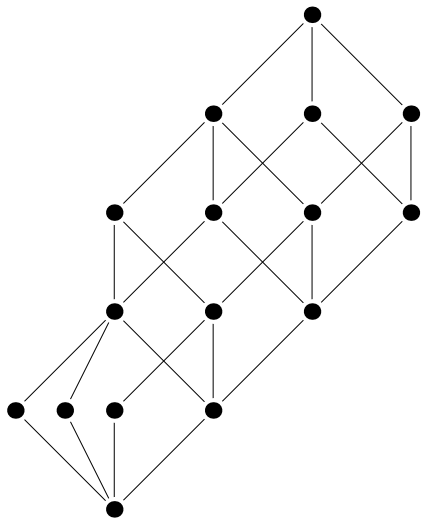
March 2009

Outline

1. introduction and motivation
2. existing approaches and methods
   - drawing by hand
   - layer approach and force directed approach
   - new approaches for lattices
   - level method (my own)
   - nested diagrams
3. geometric heuristic
   - geometric method: intro
   - rules of parallelograms and lines
   - evaluation & comparison
   - open questions and problems
   - forthcoming research
4. software for drawing lattices
   - existing software
   - our software: LatVis and EllenaArt

Introduction and motivation

- **important role of lattices** in computer science and applied math. (data analysis, information retrieval, machine learning, intelligent systems, industrial engineering, . . .)
- information usually represented by hierarchical structures, often described by graphs or lattices
- **need to visualize (draw) lattices** – (commonly) by drawings of **Hasse (upward, linear) diagram**



= **oriented graph** $\langle V, E \rangle$, where nodes V = lattice elements and edges E = lattice cover relation $\prec$

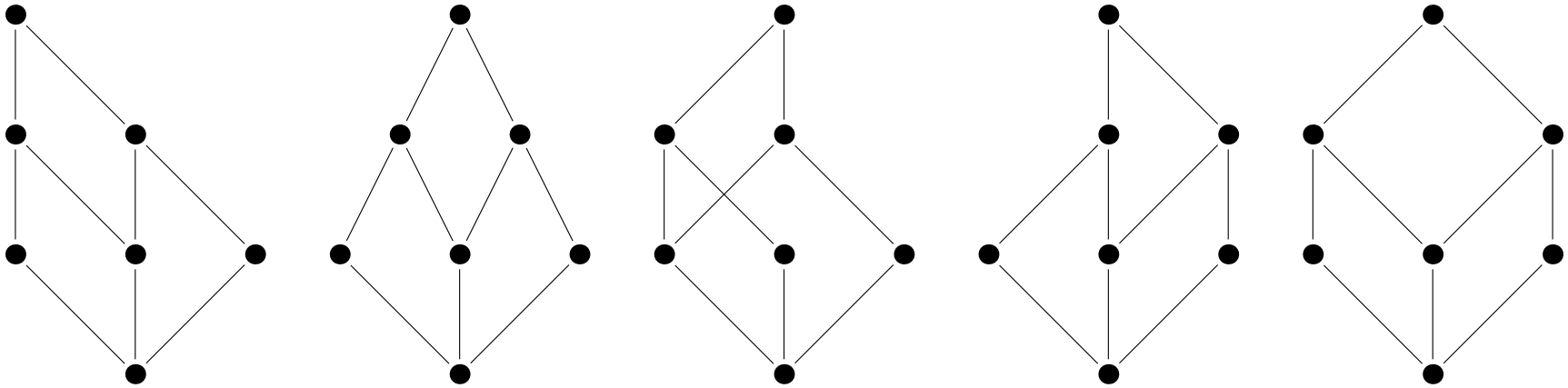
\+ **drawing conventions**:

1. node for x is drawn (as a dot or a circle) below node for $y \iff x < y$
2. nodes for x and y are connected by a straight line $\iff x \prec y$ (i.e. no lines for transitive edges and no cycles)

Introduction and motivation (con't)

Problem:

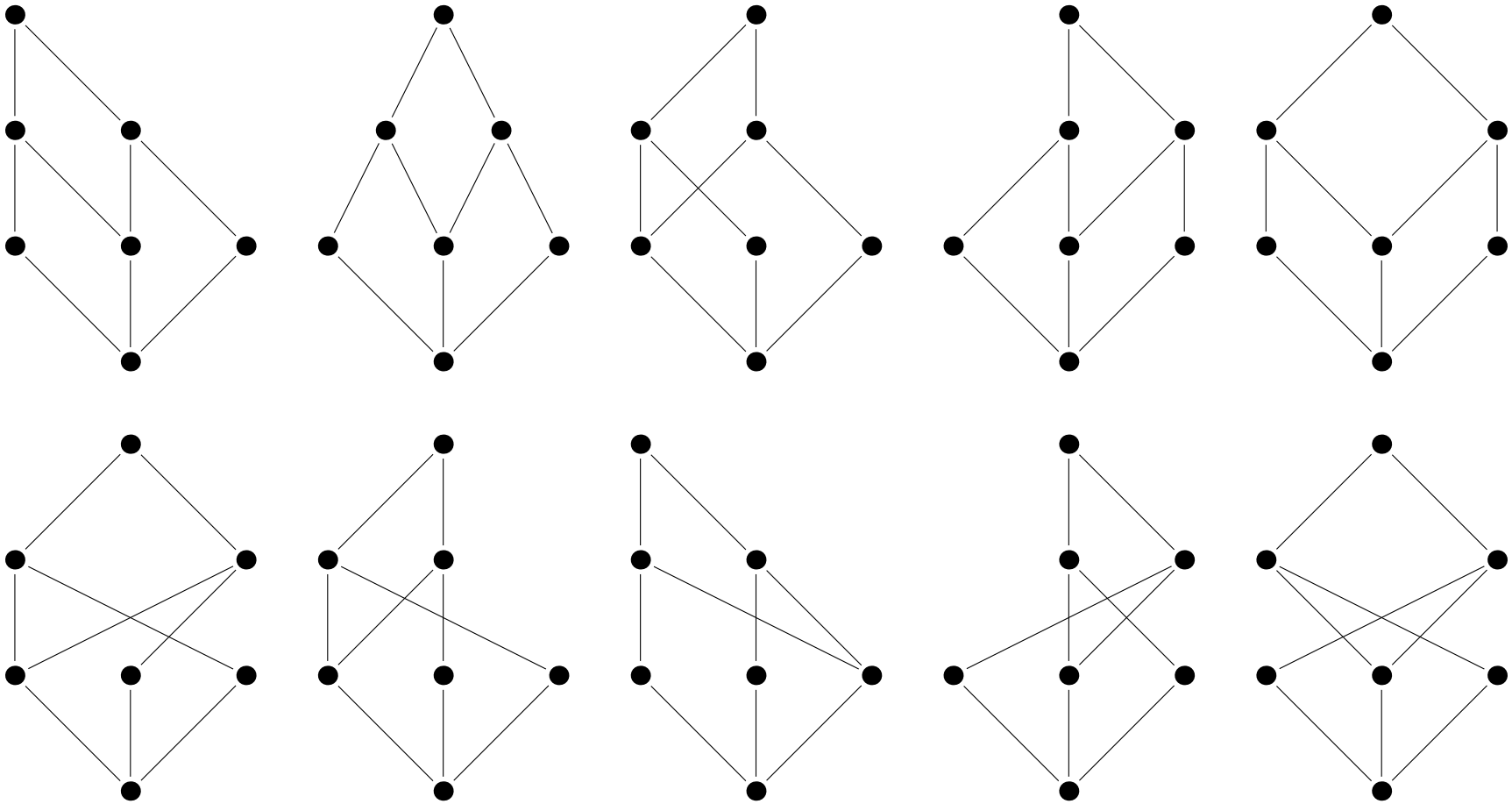
We can draw many different Hasse diagram drawings of (the Hasse diagram of) a given lattice.



Introduction and motivation (con't)

Problem:

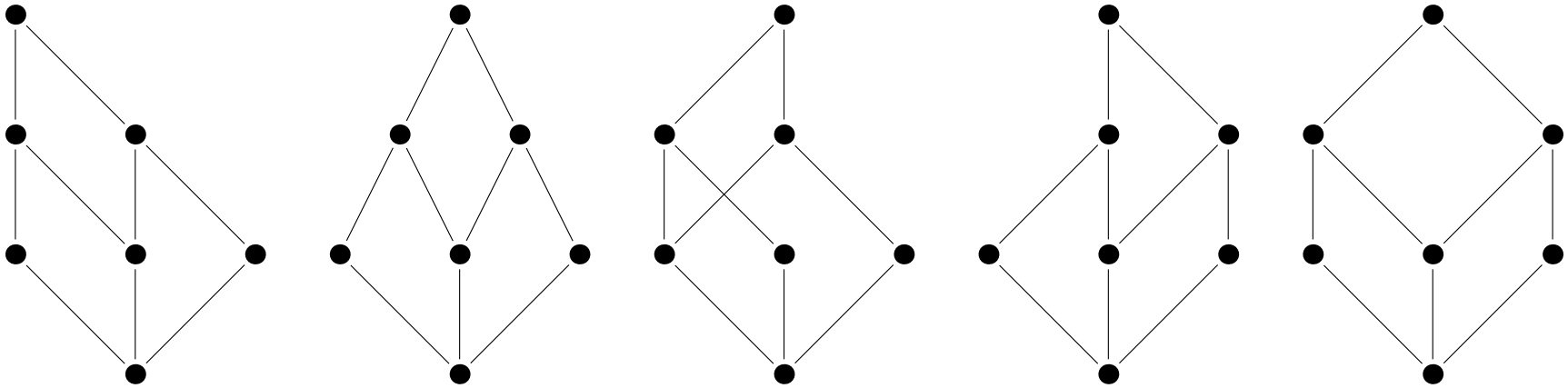
We can draw many different Hasse diagram drawings of (the Hasse diagram of) a given lattice.



Introduction and motivation (con't)

Problem:

We can draw many different Hasse diagram drawings of (the Hasse diagram of) a given lattice.



Task:

Arrange the nodes and lines of the lattice diagram drawing in order to achieve the **best visual quality, readability**, etc. . . . and do it fast and **automatically**.

Lattice drawing

- evolved from **graph drawing** (well-elaborated)
- several **subjective human aesthetics criteria**: *minimizing line crossings*, eliminating line breaks (produced by e.g. layer approach), maximizing conflict distance, angle between incident lines, *symmetries*, compactness etc.

$=$ optimization criteria used when drawing by hand

- however, **what makes the best readable diagram?**
- criteria are often contradictory and lead to computationaly difficult (NP-complete) problems

$\rightarrow$ **heuristic approaches** to drawing, but the task remains difficult (how to precisely mathematize the criteria?)

$=$ several automated drawing methods, but none universal, the best

- drawing by hand is traditionally better, but slow and tedious
- automated drawing by computer is at least a good starting point

Note: We also have criteria for labelling diagram nodes (e.g. in concept lattices, depends on application area).

Lattice drawing (con't)

How large lattices one can draw by a computer?

Up to about a hundred of elements.

There is no point in drawing whole larger lattices.

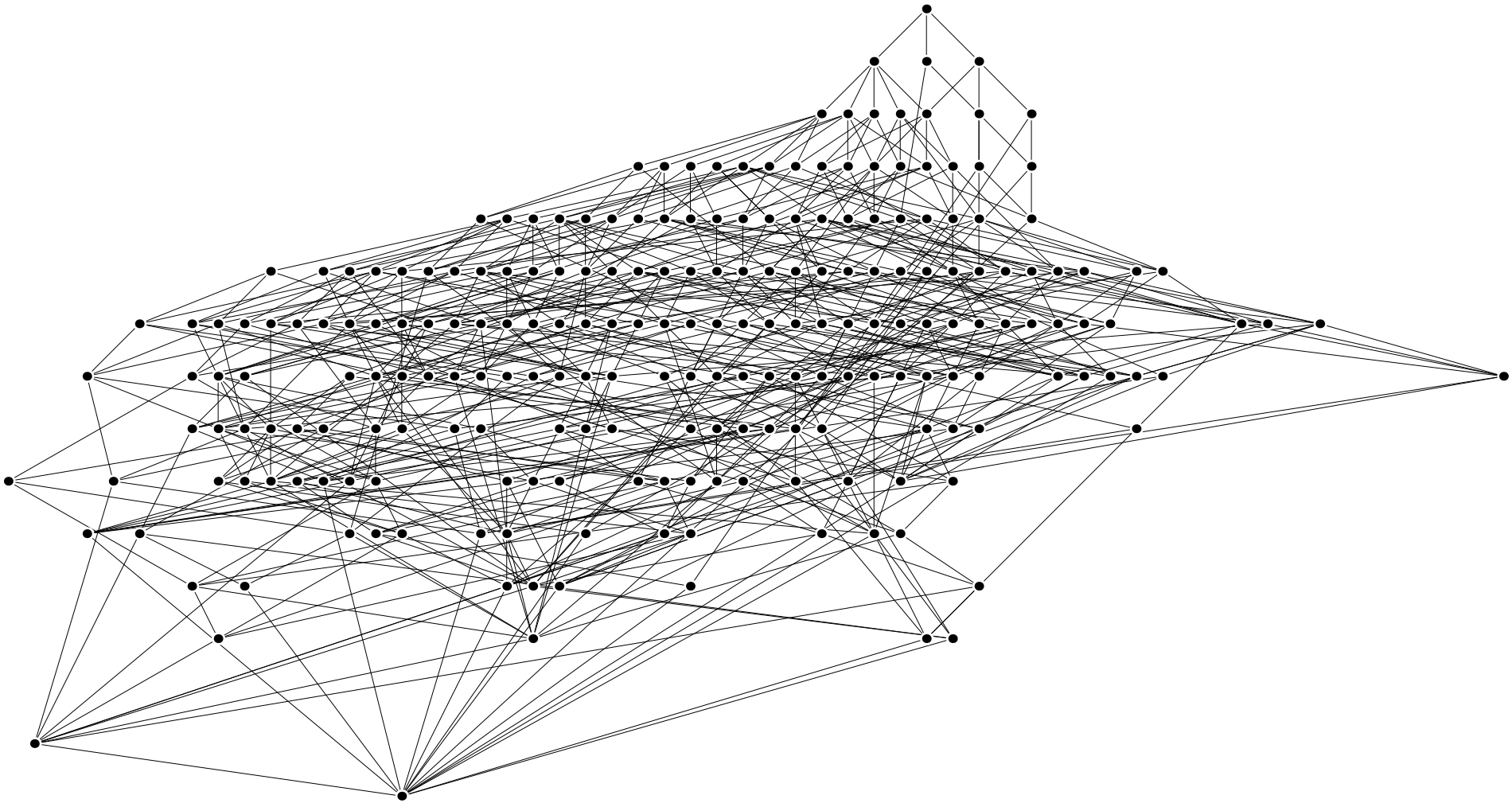
Lattice drawing (con't)

How large lattices one can draw by a computer?

Up to about a hundred of elements.

There is no point in drawing whole larger lattices.

Lattice drawing (con't)



$\longrightarrow$ divide and draw substructures (only)

Existing approaches and methods

Presumptions:

- drawing a lattice top-down, i.e. downwards from the top element
- (usually) no need of initial drawing, the input is the underlying order relation only

Drawing by hand ("intuitively")

= arranging nodes of lower **neighbors** of actual node followed by placing nodes of **infima** of the neighbors or further neighbors and so on

- **problem:** concrete placement of nodes $\longrightarrow$ "intuitively" in iterations

EXERCISE:

Draw the Hasse diagram of the following lattice:

elements:	a	b	c	d	e	f	g	h
lower neighbors:		a	a	a	$b\ c$	$b\ d$	$c\ d$	$e\ f\ g$

Existing approaches and methods (con't)

Drawing graphs

G. Di Battista, P. Eades, R. Tamassia, I.–G. Tollis:
Graph Drawing: Algorithms for the Visualisation of Graphs,
Prentice-Hall, New Jersey, 1999.

- many elaborate methods
- some can be used and adapted to draw Hasse diagrams

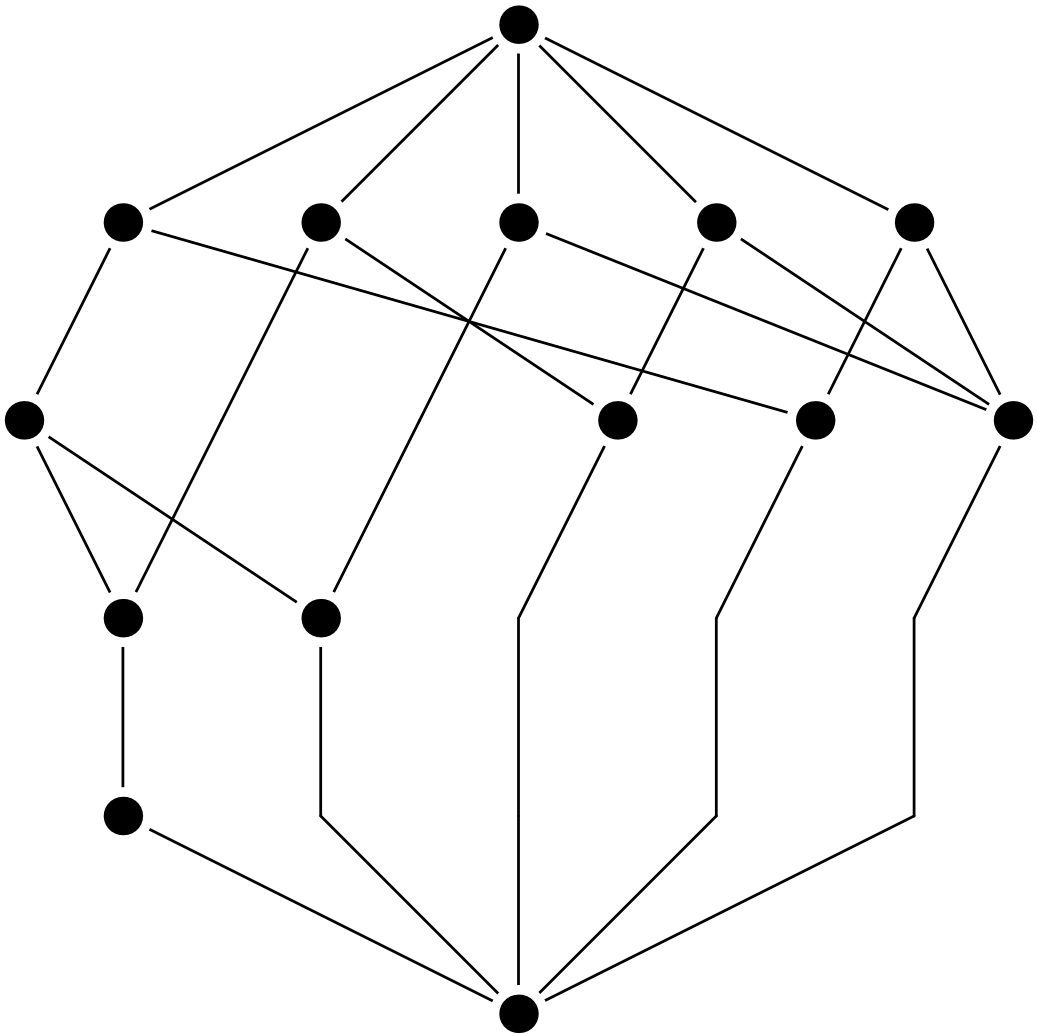
Layer approach

= nodes in **layers** based on their distance from the top node, **sorted** to achieve minimal line crossings

- steps:
  1. layer assignment (determining y-coordinates): longest path layering, Coffman-Graham layering, usage of broken lines
  2. crossing reduction: layer-by-layer node sweep by solving two-layer crossing problem (using sorting, averaging, linear programming methods etc.)
  3. x-coordinate assignment: e.g. straightening broken lines

Existing approaches and methods (con't)

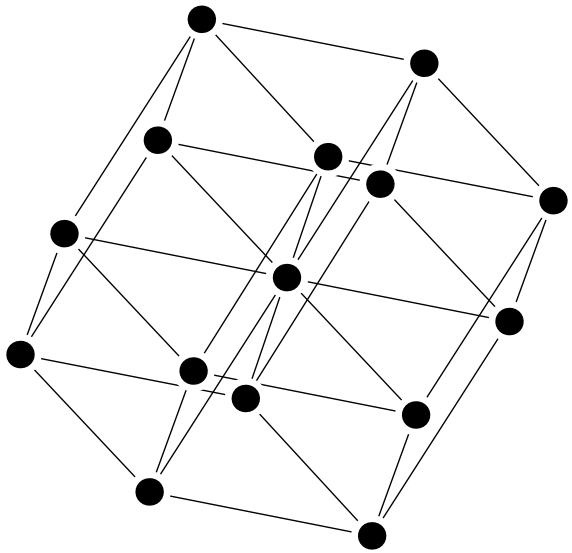
Layer approach – example



Existing approaches and methods (con't)

Force directed approach

- balancing imaginary repulsive and attractive **forces** between nodes and lines, based on **spring models**
- resulting drawing depends on **initial arrangement of nodes**
- works in iterations, results are "unpredictable" (methods are quite difficult to adapt to Hasse diagrams)
- several variants: force placement, edge-edge repulsion and others



Existing approaches and methods (con't)

New approaches for lattices

- **attribute additivity** = node position $\vec{p}$ determined by the node positions $\vec{x}$ of greater **inf-irreducible elements** $x \in M(V)$

$$\vec{p} := \vec{a} + \sum_{x \in M(V) \mid p \leq x} \vec{x}$$

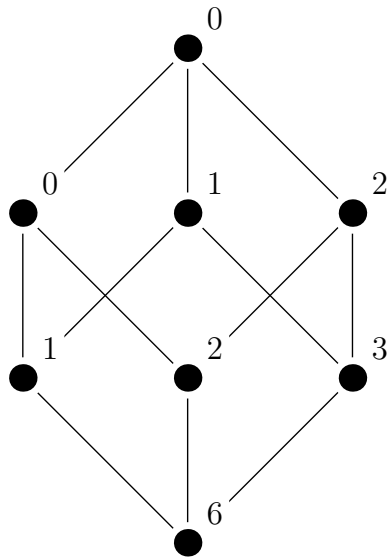
- combinations of methods, e.g. **hybrid method** = layer + attribute additivity
- special methods, due to visualizing **concept lattices** in FCA, e.g. **grid method** = **projection** of the lattice placed into a **multidimensional grid** onto a suitable plane

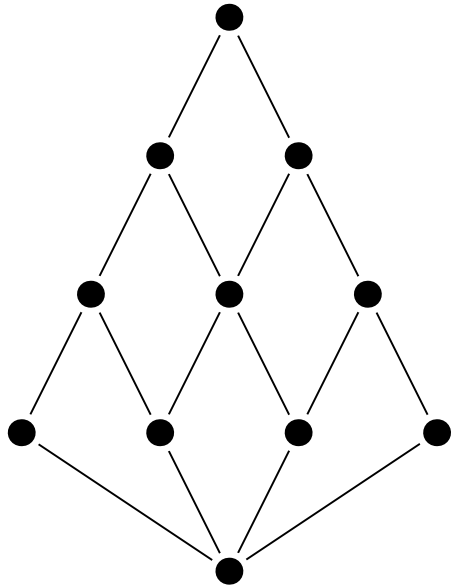
Existing approaches and methods (con't)

Level method (my own)

- similar to drawing by hand and layer approach
- solves concrete placement of nodes minimizing line crossings and maximizing compactness of the diagram

= arranging nodes of lower neighbors of nodes from previous **level**/layer in a new level/layer, **evenly** below the nodes and **ordered** by ordering of the nodes (based on non-decreasing numbering)





Existing approaches and methods (con't)

Present methods:

- produce quite good, readable, diagrams, however, for **smaller lattices only** (around 30 elements at max) – because of **global optimization**?
- attribute additivity based win
- **performance** of drawing or interactive altering is not a problem (up to a hundred of elements)

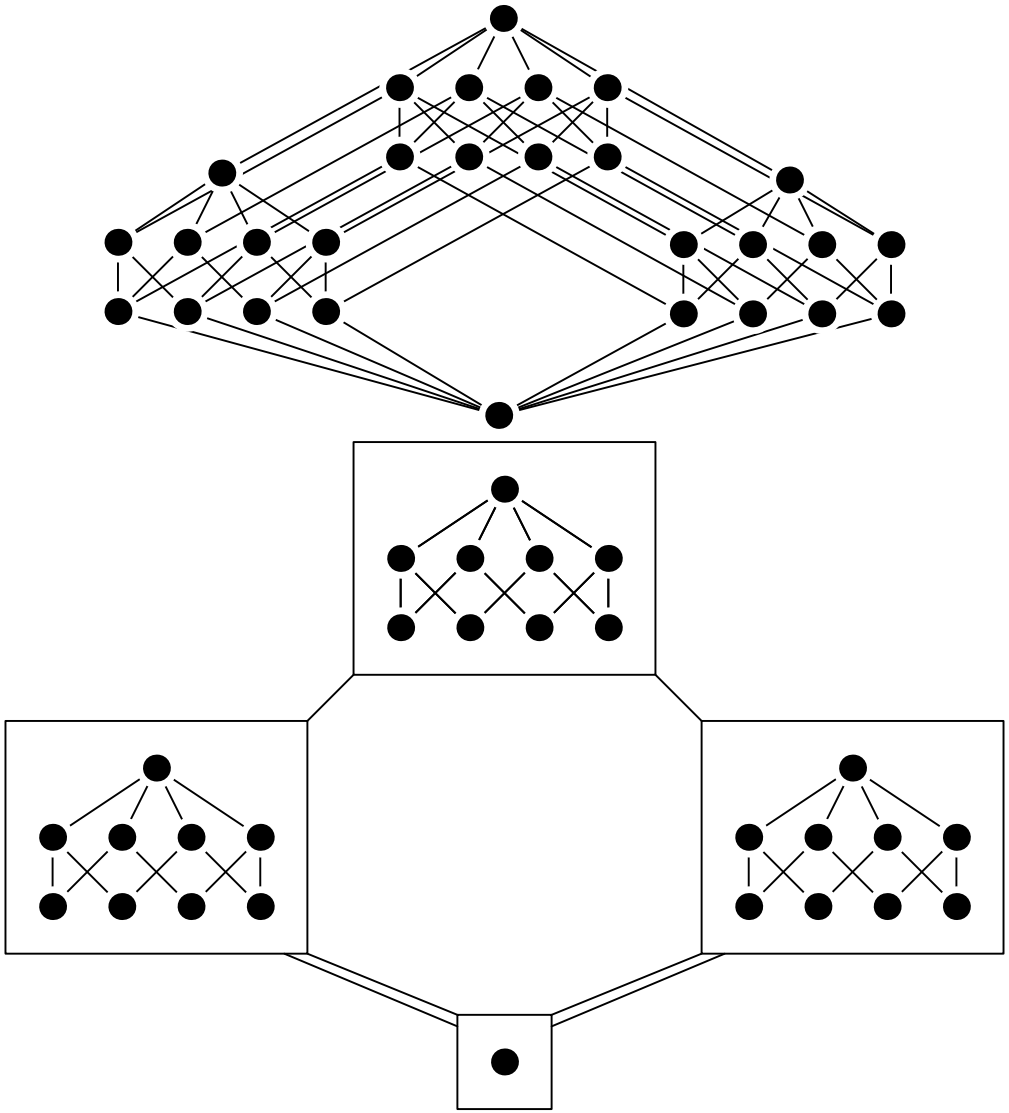
Nested diagrams

= separated **parts** of the whole diagram drawn as **nested diagrams**, bunches of parallel lines replaced by a single (or double) line

- used by Wille et al. in FCA
- interesting idea, well-readable drawings (even for tens of elements), but unusual and not very used
- **problem:** identifying parts – needs structural analysis of the lattice (hard!)

Existing approaches and methods (con't)

Nested diagrams – example



Geometric method: intro

- proposed by Wille et al. in 1989, re-introduced in 1993

  R. Wille: Geometric representation of concept lattices. In: Opitz O. (Ed.): Conceptual and numerical analysis of data, pp. 239–255, Springer-Verlag, Berlin-Heidelberg, 1989.

  G. Stumme, R. Wille: A geometric heuristic for drawing concept lattices. In: Tamassia R., Tollis I. G. (Eds.): Graph drawing, pp. 85–98, Springer, Berlin-Heidelberg-New York, 1993.

- originally developed for drawing **concept lattices** in FCA
- (originally) based on a **geometric interpretation** of the lattice:

– finding a best possible diagram layout with the help of a **geometrical diagram** (auxiliary picture when drawing by hand)

Geometric method: intro (con't)

Geometrical diagram

= a look at the 3D visualization of the lattice from its top element:

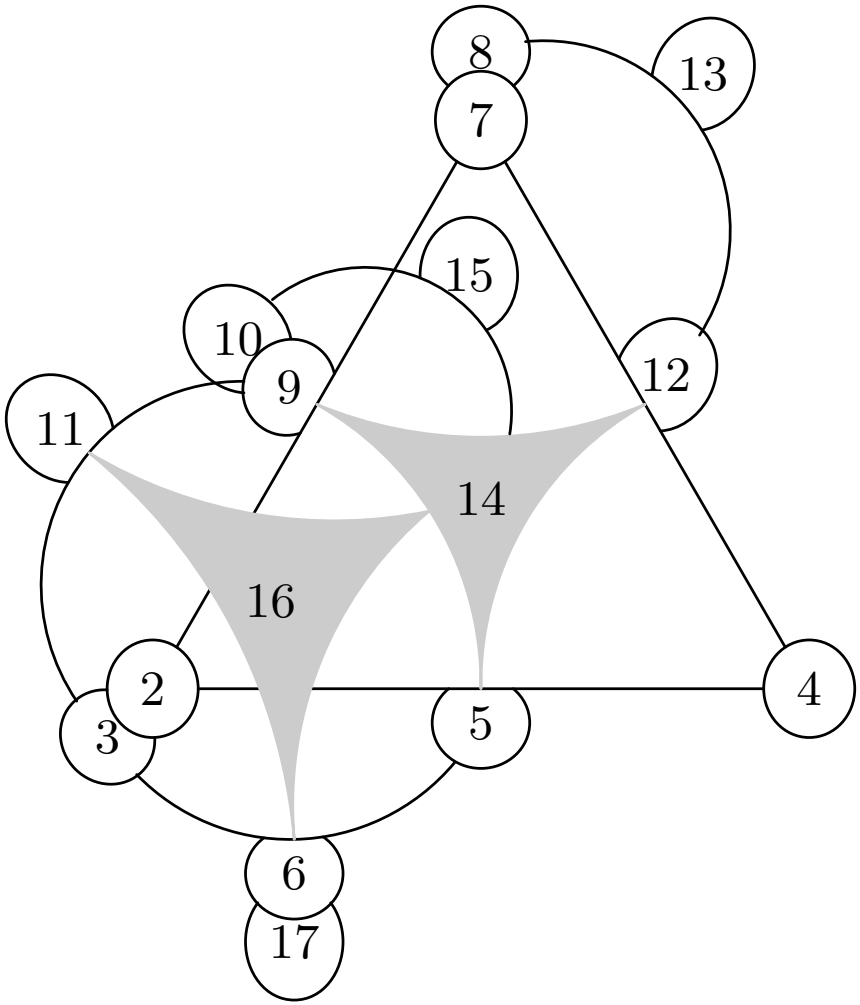
- lower neighbor of top element $\longrightarrow$ circled label
- element with one upper neighbor $\longrightarrow$ circled label partly covered by the neighbor's label
- elements with two upper neighbors $\longrightarrow$ circled label partly covered by a line connecting the neighbors' labels
- elements with three upper neighbors $\longrightarrow$ label inside a (sloped filled) triangle connecting neighbors' labels
- . . . (the top and the least element are omitted)

EXERCISE: Draw the geometrical diagram for the following lattice:

elements	upper neighbors
1	
2	1
3	2
4	1
5	2 4
6	3 5
7	1
8	7
9	2 7
10	9
11	3 9
12	4 7
13	8 12
14	5 9 12
15	10 14
16	6 11 14
17	6
18	13 15 16 17

Geometric method: intro (con't)

Geometrical diagram – example



Geometric method: intro (con't)

→ drawing of the Hasse diagram by **recognizing and realizing geometrical patterns**

– using mainly two **geometric rules**:

1. rule of parallelograms
2. rule of lines

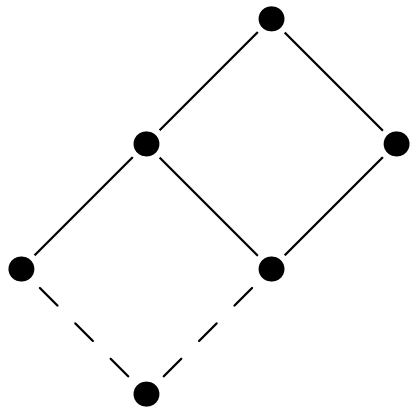
Rule of parallelograms

Definition (Rule of parallelograms)

A new node should be placed in such a way that the node together with some already placed nodes and lines forms a **parallelogram** (the geometric shape with parallel lines, e.g. diamond or rhomboid).

- rather a general rule, looks simple, but
- immediate problem: vague formulation "some already placed nodes"

$\rightarrow$ most commonly selected nodes = of **pair of upper neighbors + their common upper neighbor (supremum)**



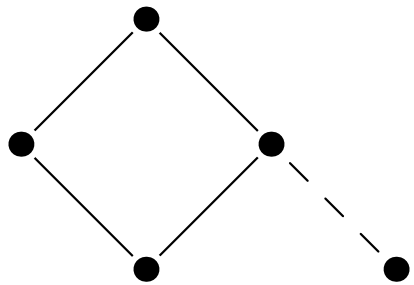
Rule of lines

Definition (Rule of lines)

A new node should be placed on a **prolonged line** connecting some already placed nodes, preferably at the same distance as the distance between the nodes.

- again, rather a general simple looking rule, but
- again, immediate problem: vague formulation "some already placed nodes"

$\rightarrow$ most commonly selected nodes = of a **single upper neighbor + its upper neighbor**



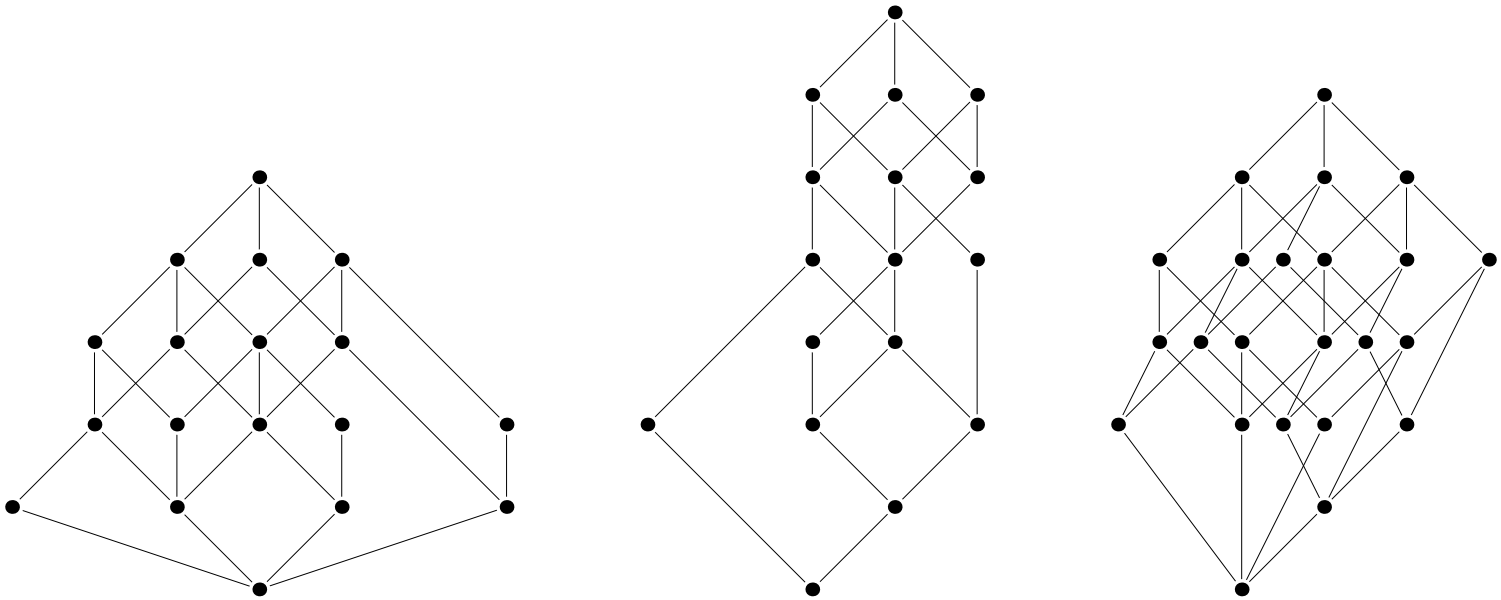
Geometric method (con't)

Application of rules

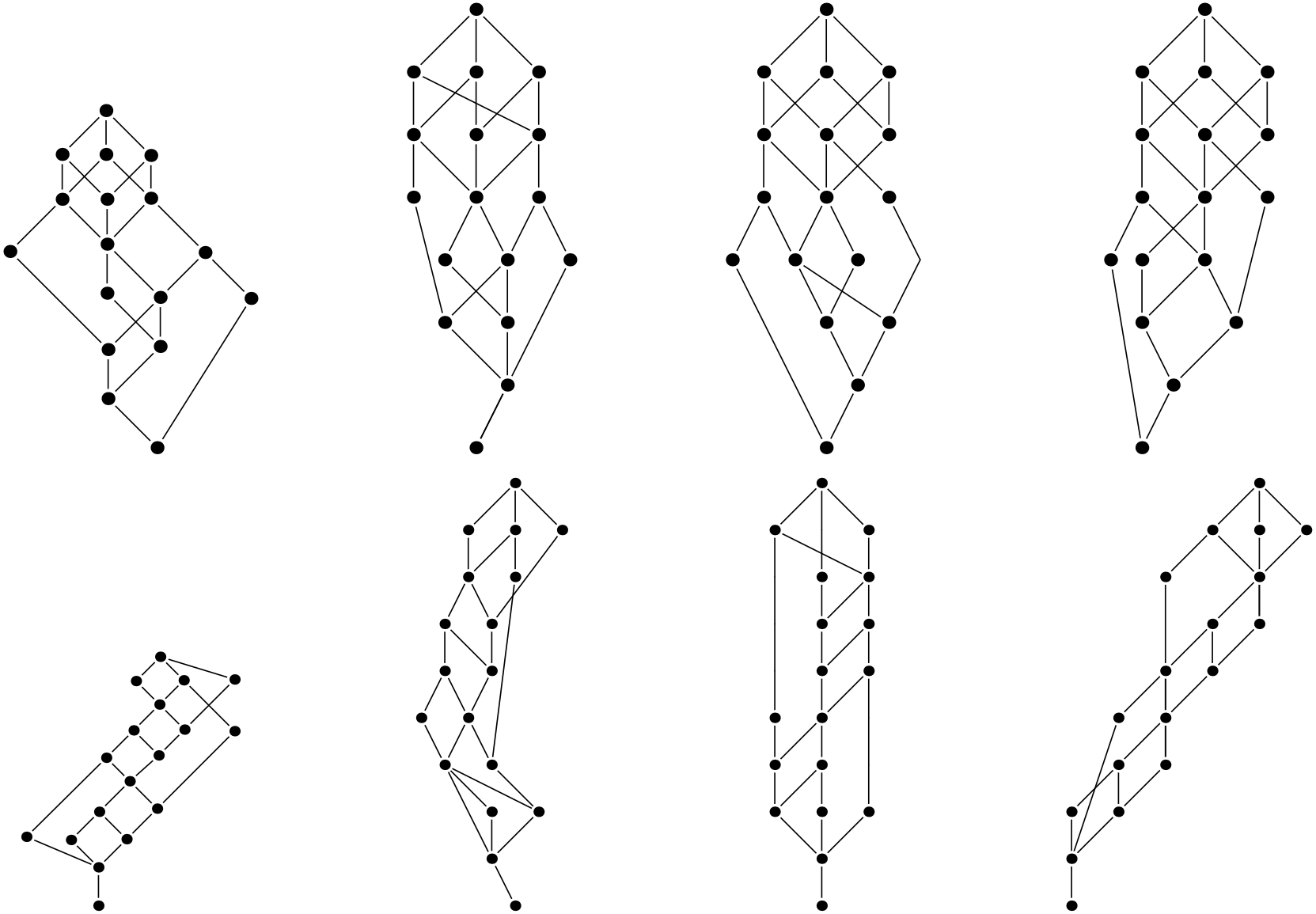
- results in **many parallel lines and regular geometric shapes** in the diagram (fulfilling aesthetics criteria) $\longrightarrow$ good level of readability even for medium sized lattices (30–50 elements)

= essential part of discovering regular geometrical shapes, structures and patterns

- aim: best possible **overall geometric regularity of the diagram**



Comparison of methods



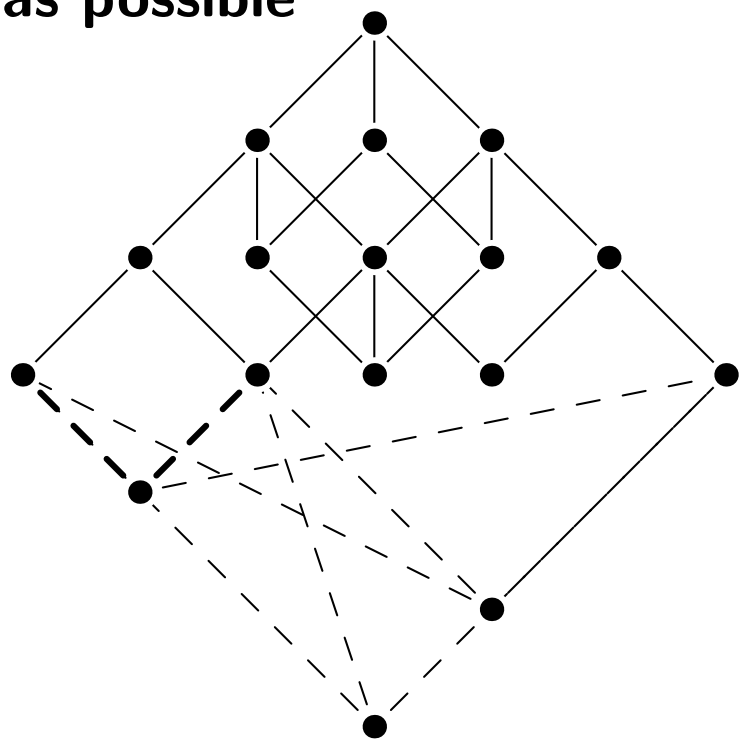
hand drawn level layer geometric

Open questions and problems (1)

However, application of rules is not a straightforward action at all, there are many **decision points** in a new node placement:

Rule of parallelograms

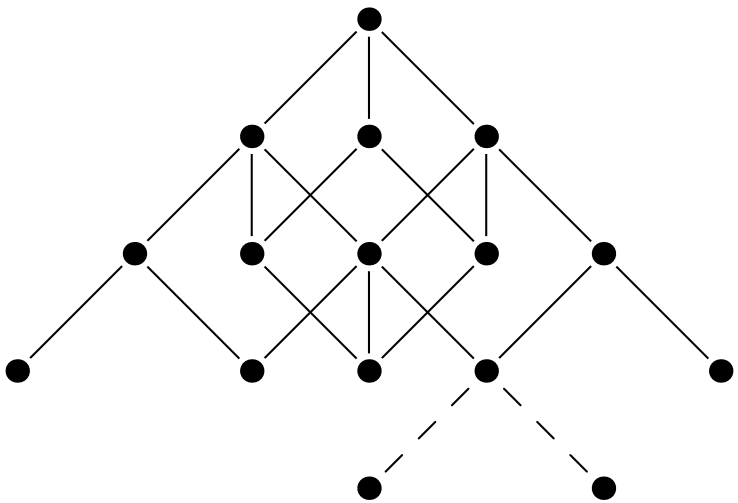
- there can be more than one pair of upper neighbors $\rightarrow$ the supremum of a pair should be as "close" as possible (ideally an upper neighbor) = make the **parallelogram as small as possible**
- if the supremum is not an upper neighbor, *should we place the new node in the middle bellow its upper neighbors?* (violating the rule, e.g. the least element)
- ...



Open questions and problems (2)

Rule of lines

- there can be more than one upper neighbor of the single upper neighbor $\longrightarrow$?
- *should we place the new node straight bellow its upper neighbor?* (violating the rule)
- ...



The final choices in the the decisions often depends on suggested placements for **other elements**.

$\downarrow$

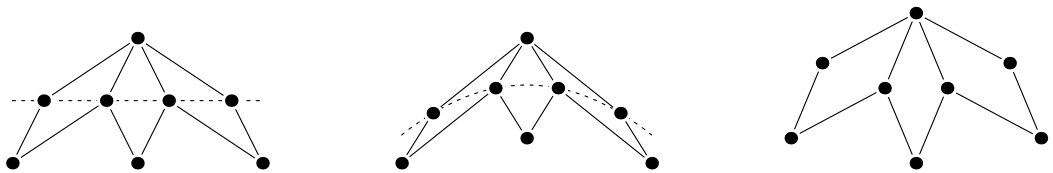
complex heuristic on (semi)local optimization problems

Open questions and problems (3)

Arrangement of co-atoms (inf-irreducibles)

- important **starting point** in drawing the diagram!
- or whenever the rule of lines suggests the same location

$\rightarrow$ we can place them on an imaginary horizontal line, a parabola, using a force directed approach, ...

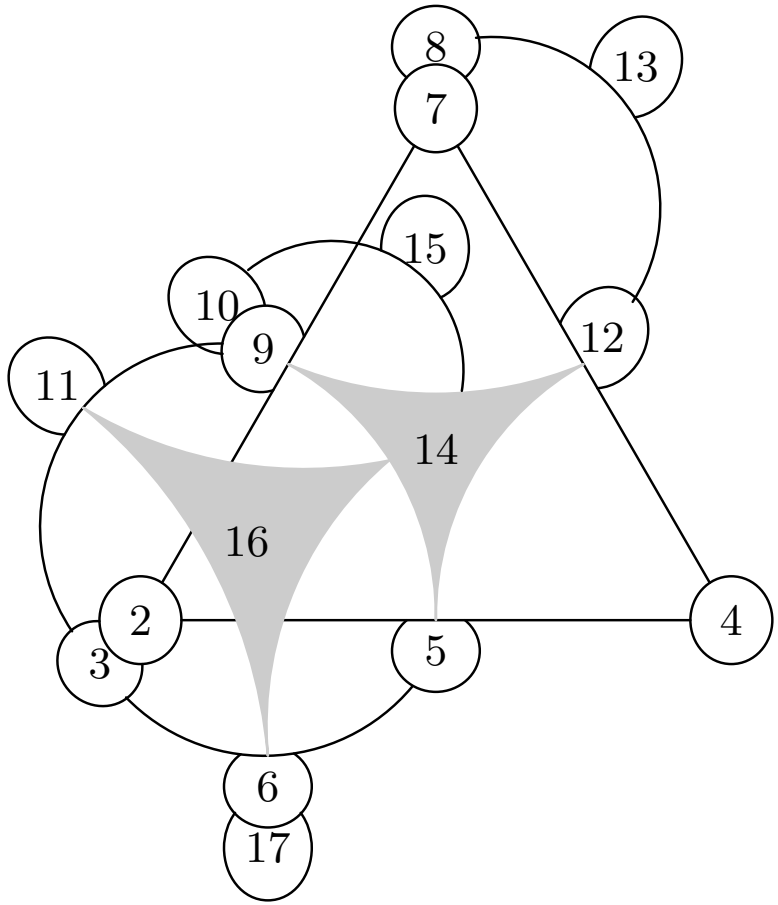


- *which order of the elements?* $\rightarrow$ should place elements which have more lower neighbors **aside** (since we will need a space for the neighbors)
- the same applies for elements with equal suggested placement (by the rules or other way)

Geometric method (con't)

EXERCISE:

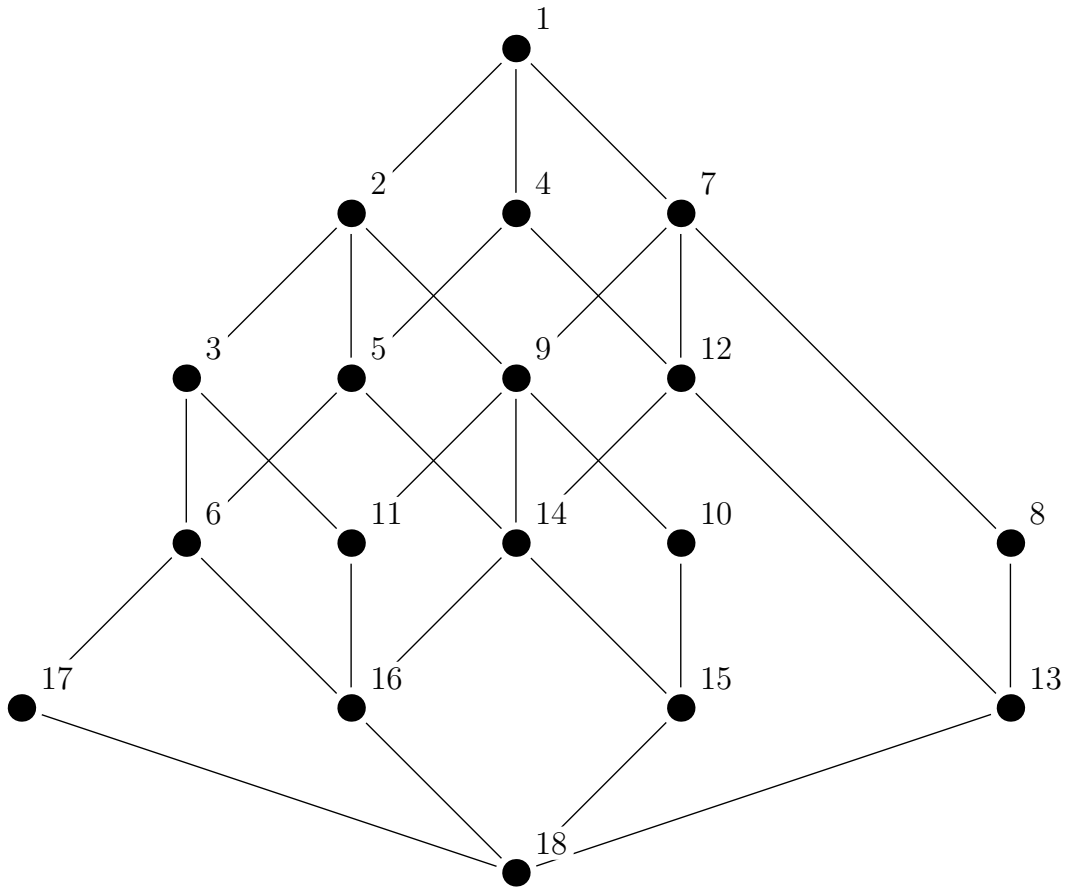
Draw the Hasse lattice diagram for the geometrical diagram from the previous exercise.



Geometric method (con't)

EXERCISE:

Draw the Hasse lattice diagram for the geometrical diagram from the previous exercise.



Forthcoming research

- no further papers on the geometric method (since the initial two or three in 1989-93)!
- we find the method very **promising**
- we have explored (some of) the problems and proposed (some) ideas and solutions

→ we are developing a **new method for automated lattice drawing** inspired by and further refining the geometric method

Main idea: (intermediate) **logical diagram description**

- similar to geometrical diagram used in the original geometric method, but more general
- **contains:** constraints of Hasse diagram, space constraints, node placements suggested by the geometric rules and other principles, evaluations of the suggested placements, . . .
- obtaining final diagram = heuristic solutions to both local and global optimization problems aimed at producing the best possible diagram

Software for drawing lattices: requirements

. . . to test, evaluate, develop, fine-tune and . . . of course **use** the drawing method (by end users)

Requirements

- producing the **best possible diagram drawing** of any given lattice (of course)
- **fine-tuning** the diagram drawing by hand (or additional methods): moving nodes or parts, grid aligning, hiding/folding parts, zooming, rotating, etc.
- **displaying parts** of the lattice: lower/upper neighbors/cones, infs/sups, paths, etc.
- **exporting** the diagram drawing (or part of it) to the picture in a paper, look customization
- editing the lattice (underlying order relation)
- . . .
- usable by **end user** (i.e. graphical)

Software for drawing lattices: concept lattice

Software for FCA (Formal Concept Analysis)

– original purpose: a tool for FCA

= drawing the resulting concept lattice only

Toscana, Anaconda, Diagram

- FCA tools from the (former) FCA group of TU Darmstadt, Germany
- not available anymore

ToscanaJ

- Java reimplementation of Toscana, open source, part of Tockit FCA framework (`http://tockit.sourceforge.net`)
- lattice diagram viewer only, displaying lower/upper cones, look customization
- drawing methods?, nested diagrams
- `http://toscanaj.sourceforge.net/`

Software for drawing lattices: concept lattice (con't)

Galicia

- (rich) FCA platform
- layer and force directed approaches (including 3D variants)
- node moving, rotating, zooming
- written in Java, open source
- `http://www.iro.umontreal.ca/~galicia/`

Concept Explorer

- FCA tool
- layer, force directed (including the one in LatDraw) and grid approaches
- node moving, displaying lower/upper cones, grid aligning, zooming
- written in Java, open source
- `http://conexp.sourceforge.net/`

Software for drawing lattices: concept lattice (con't)

GaloisExplorer

- FCA tool, lattice diagram viewer only
- force directed approach? (3D variant)
- features?
- written in C++ (MS Windows, Apple Mac OS), free software
- `http://galoisexplorer.sourceforge.net`

JaLaBa

- online FCA web application
- uses LatDraw for drawing the concept lattice
- `http://maarten.janssenweb.net/jalaba/JaLaBA.pl`

Software for drawing lattices: any lattice

Software for drawing (arbitrary) lattices

LatDraw

- online Java applet or Java application by Ralph Freese from the University of Hawaii
- used by several other tools (e.g. JaLaBa, JavaMath plugin to Maple), source upon request (API for free)
- combined layer and force directed approach
- lattice diagram viewer only, rotating
- `http://www.math.hawaii.edu/~ralph/LatDraw/`, `http://www.latdraw.org`

Software for drawing lattices: any lattice (con't)

GAP – poset visualization part

- online (only) Java applet by Peter Jipsen from the Chapman University, CA, open source
- (simplified) combined layer and force directed approach
- limited node moving
- `http://www1.chapman.edu/~jipsen/gap/posets.html`

Conclusion

- some FCA tools, limited in lattice drawing features
- **JUST TWO** Java applets for drawing arbitrary lattices!, yet very limited
- there is quite a lot of graph drawing tools (`http://www.graphviz.org` `http://graphdrawing.org`), but none of them with lattice Hasse diagram drawing feature

Software for drawing lattices: any lattice (con't)

GAP – poset visualization part

- online (only) Java applet by Peter Jipsen from the Chapman University, CA, open source
- (simplified) combined layer and force directed approach
- limited node moving
- `http://www1.chapman.edu/~jipsen/gap/posets.html`

Conclusion

– some FCA tools, limited in lattice drawing features

– **JUST TWO** Java applets for drawing arbitrary lattices!, yet very limited

– there is quite a lot of graph drawing tools (`http://www.graphviz.org` `http://graphdrawing.org`), but none of them with lattice Hasse diagram drawing feature

Our software for drawing lattices

LatVis (Jan Outrata, 2003)

- lattice and poset editor and visualizer
- developed with my MSc. thesis at **Dept. Computer Science, Palacký University, Czech Rep.** in 2003
- **layer** approach, (author's own) **level** and (original) **geometric** methods
- fine-tuning: user selected node moving (with coordinates displayed), node hiding
- displaying and selecting parts: lower/upper neighbors/cones, infs/sups, min/max, paths
- editing: copy&paste, undo/redo
- export: Metapost, Encapsulated Postscript (and PDF), look customization, saving to a XML document
- written in C++ (MS Windows, GNU/Linux), free software (GNU GPL)
- `http://phoenix.inf.upol.cz/~outrata/latvis/`

Our software for drawing lattices (con't)

EllenaArt (Lukas Hostalek, 2007)

- lattice and poset drawing tool
- developed with MSc. thesis of Lukas Hostalek at **Dept. Computer Science, Palacký University, Czech Rep.** in 2007
- **force directed** approach (three variants), (author's own) heuristic and (original) **geometric** methods
- fine-tuning: node moving (with coordinates displayed), grid aligning, zooming
- export: Encapsulated Postscript and PDF, look customization, saving to a XML document
- written in Java, open source
- `http://phoenix.inf.upol.cz/~hostalel/en/ellenart/ellenart.html`

Thank you!

LatVis `http://phoenix.inf.upol.cz/~outrata/latvis/`

EllenaArt `http://phoenix.inf.upol.cz/~hostalel/en/ellenart/ellenart.html`