Lattices and classification

SET07106 Mathematics for Software Engineering

School of Computing
Edinburgh Napier University
Module Leader: Uta Priss

2010

Outline

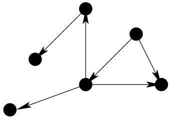
Reminder: properties of binary relations

- antisymmetric: $a \neq b : (a, b) \implies \text{ not } (b, a)$
- reflexive: for all elements: (a, a)
- transitive: $(a, b), (b, c) \implies (a, c)$

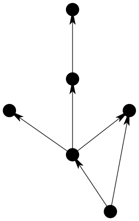
A **partially ordered set (poset)** is a reflexive, antisymmetric and transitive binary relation.

Directed acyclic graphs

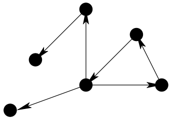
antisymmetric:



directed acyclic graph

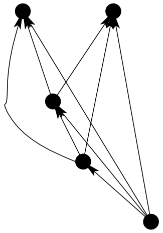


directed cyclic graph:

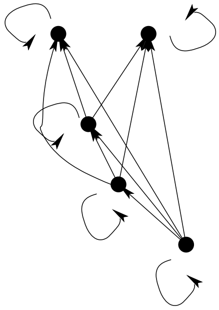


Partially ordered set (poset)

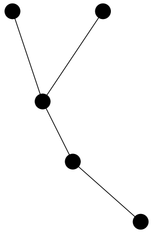
antisymmetric
and transitive:



antisymmetric,
reflexive and transitive:



Hasse diagram:



Order relation $\leq$

The binary relation of a partially ordered set is written as $\leq$.

- antisymmetric: $a \neq b : a \leq b \Longrightarrow$ not $b \leq a$
- reflexive: for all elements: $a \leq a$
- transitive: $a \leq b, b \leq c \Longrightarrow a \leq c$

Order relation $\leq$

The binary relation of a partially ordered set is written as $\leq$.

- antisymmetric: $a \neq b : a \leq b \implies$ not $b \leq a$
- reflexive: for all elements: $a \leq a$
- transitive: $a \leq b, b \leq c \implies a \leq c$

For example:

- antisymmetric: $5 \leq 7 \implies$ not $7 \leq 5$
- reflexive: $6 \leq 6$
- transitive: $3 \leq 4, 4 \leq 7 \implies 3 \leq 7$

Order relation $\leq$

The binary relation of a partially ordered set is written as $\leq$.

- antisymmetric: $a \neq b : a \leq b \Longrightarrow$ not $b \leq a$
- reflexive: for all elements: $a \leq a$
- transitive: $a \leq b, b \leq c \Longrightarrow a \leq c$

For example:

- antisymmetric: $5 \leq 7 \Longrightarrow$ not $7 \leq 5$
- reflexive: $6 \leq 6$
- transitive: $3 \leq 4, 4 \leq 7 \Longrightarrow 3 \leq 7$

If $a \neq b$, one can also write $a < b$ or $a > b$.

Examples of posets

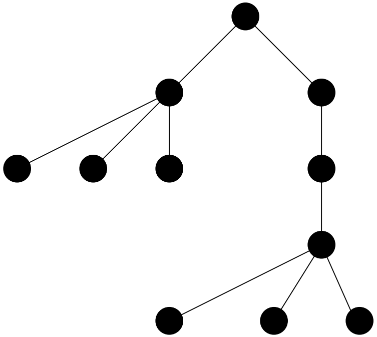
- Scheduling problems: PERT charts, flow charts
- Dependency graphs: software installers, compilers, variable dependencies
- C++ class hierarchy (a hierarchy where every node can have multiple parents)
- Part-whole relationships (e.g. food and ingredients)

Tree order

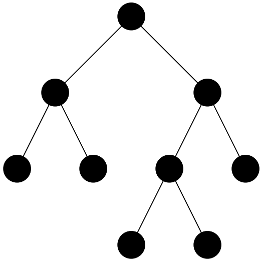
A tree is a poset where each child node has exactly one parent node. A tree has a single root node.

A binary tree is a tree where each node has either exactly two children or no children.

tree:



binary tree:



Examples of tree orders

- B-tree search structures in operating systems and databases
- Directory structures in operating systems (without symbolic links)
- Java class hierarchy
- XML document structure
- Library classification systems
- Genealogy

Exercise

Find all trees with 6 nodes.

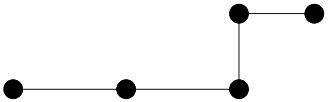
Hint: there are 6 different ones.

Linearly (or totally) ordered set

A linearly ordered set is a poset where each two nodes are comparable, i.e. for a, b, either $a = b$ or $a < b$ or $b < a$.

For example:
$1 < 2 < 3 < ... < 15 < 16 < 17 < ....$

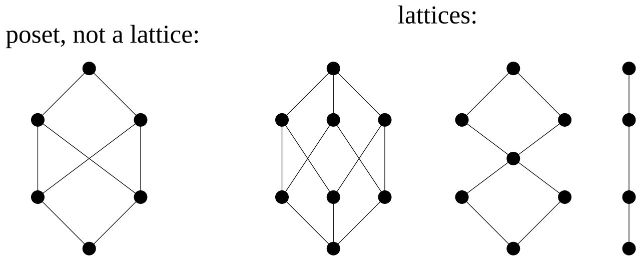
$A < B < C < D < ... < Z < a < b < ... < z$



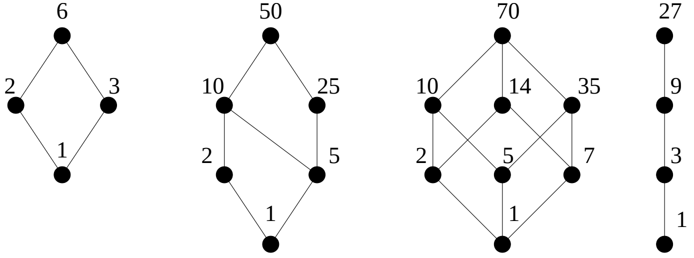
Lattice

A lattice is a poset where each two nodes have a greatest common child node and a least common parent node.

Lattices and posets



Example: divisor lattices



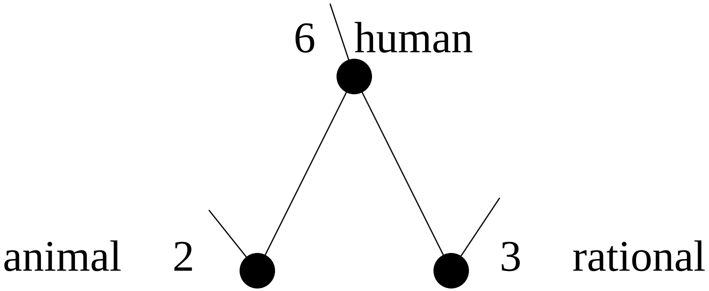
Exercise

Draw the divisor lattices for 18, 36 and 108.

Examples of lattices

- The subset relation $\subset$ among sets.
- The integers: $1 < 2 < 3 < 4 <$ .... (This lattice is infinite.)
- Boolean logic.
- Concept lattices (see the next slides).

Leibniz: Characteristica universalis (17th century)



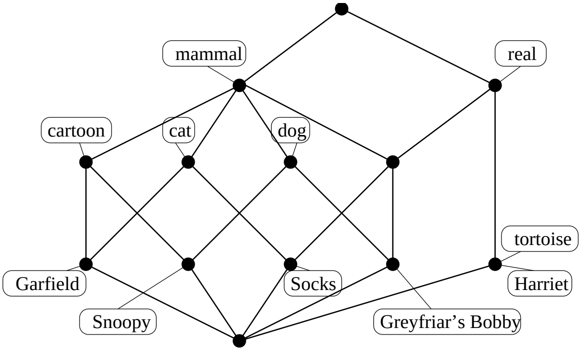
Formal concept analysis

Formal concept analysis (FCA) is a mathematical method for data analysis and knowledge representation which uses lattice theory. A binary relation (or **formal context**) is converted into a **concept lattice**.

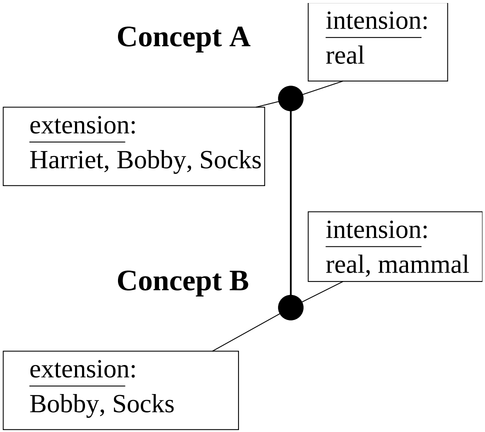
FCA is based on philosophical notions of the duality of concepts (extension and intension). It was invented in the 1980s and has since grown into an international field of research with applications in many domains.

Formal context and concept lattice

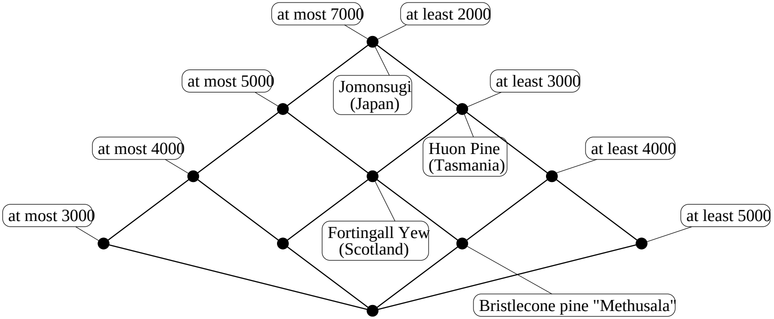
	cartoon	real	tortoise	dog	cat	mammal
Garfield	×				×	×
Snoopy	×			×		×
Socks		×			×	×
Greyfriar's Bobby		×		×		×
Harriet		×	×			



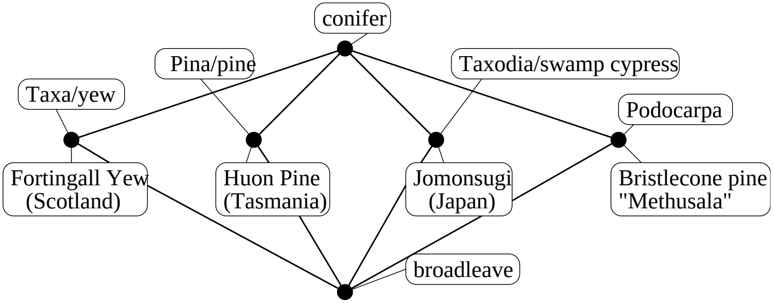
Subconcept-superconcept relation



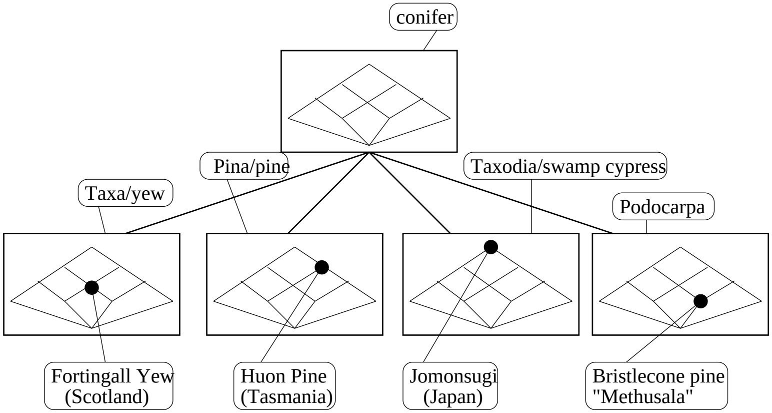
An interval scale



A nominal scale



A nested line diagram



FCA applications in software engineering

- Visualise and analyse Java class hierarchies
- Detect variable or code dependencies
- Analyse dependencies and vulnerabilities in Linux
- Re-engineering, code analysis, module restructuring