Why

We want to talk about optimally eliminating variables in a system of linear equations.[1]

Definition

An *ordered* undirected graph is *filled* or *monotone transitive* if all higher neighborhoods induce complete subgraphs. An ordering σ of an undirected graph (V, E) is a *perfect elimination ordering* if the ordered undirected graph $((V, E), \sigma)$ is filled.

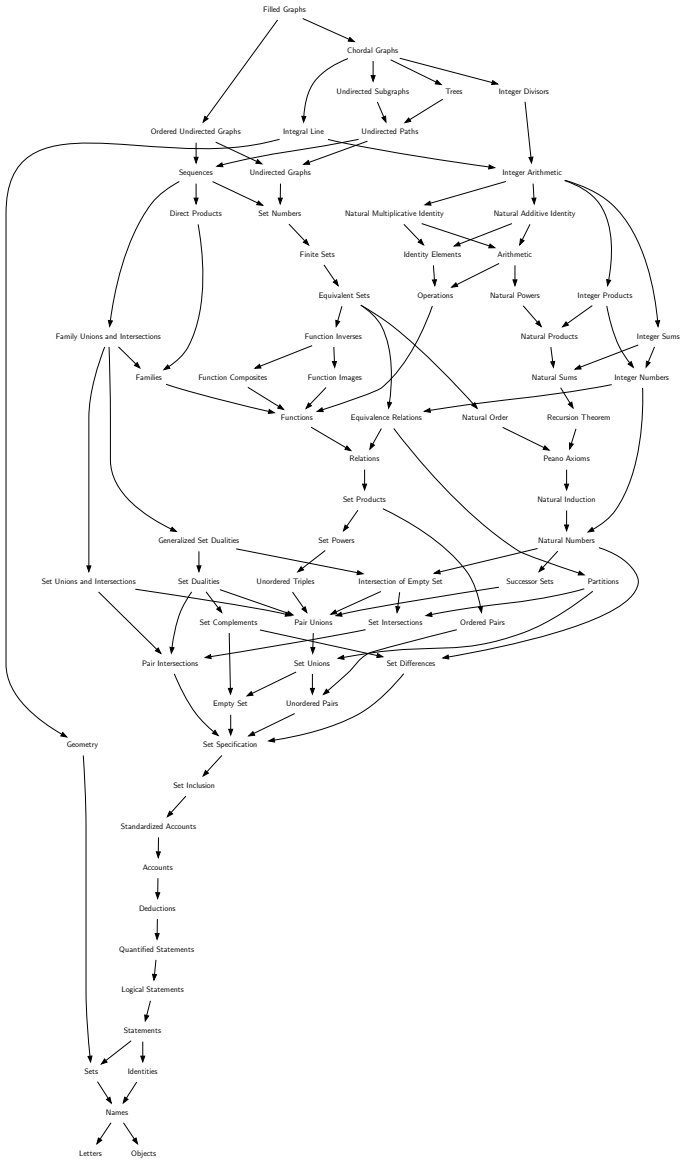
Let $G = ((V, E), \sigma)$ be an ordered undirected graph. G is filled if, for all $v \in V$, $w, z \in \mathrm{adj}^+(v) \longrightarrow \{w, z\} \in E$. Equivalently stated, G is filled if, for all $i < j < k$, $\{\sigma(i), \sigma(j)\} \in E$ and $\{\sigma(i), \sigma(k)\} \in E$ imply $\{\sigma(j), \sigma(k)\} \in E$.

Chordality

Prop. 1. *If (V, E, σ) is a filled graph, then (V, E) is chordal.*

Proof. Take the vertex with the lowest index on a cycle of length greater than three. Take $\square$

[1] Future editions will expand. For example, this sheet is needed for perfect elimination orderings.


Filled Graphs
Chordal Graphs
Undirected Subgraphs
Trees
Integer Divisors
Ordered Undirected Graphs
Integral Line
Undirected Paths
Sequences
Undirected Graphs
Integer Arithmetic
Direct Products
Set Numbers
Natural Multiplicative Identity
Natural Additive Identity
Finite Sets
Identity Elements
Arithmetic
Equivalent Sets
Operations
Natural Powers
Integer Products
Family Unions and Intersections
Function Inverses
Natural Products
Integer Sums
Families
Function Composites
Function Images
Natural Sums
Integer Numbers
Functions
Equivalence Relations
Natural Order
Recursion Theorem
Relations
Peano Axioms
Set Products
Natural Induction
Generalized Set Dualities
Set Powers
Natural Numbers
Set Unions and Intersections
Set Dualities
Unordered Triples
Intersection of Empty Set
Successor Sets
Partitions
Set Complements
Pair Unions
Set Intersections
Ordered Pairs
Pair Intersections
Set Unions
Set Differences
Empty Set
Unordered Pairs
Geometry
Set Specification
Set Inclusion
Standardized Accounts
Accounts
Deductions
Quantified Statements
Logical Statements
Statements
Sets
Identities
Names
Letters
Objects