Decision Procedures in First Order Logic

Decision Procedures for Equality Logic

Part III – Decision Procedures for Equality Logic and Uninterpreted Functions

- Algorithm I – From Equality to Propositional Logic
  - Adding transitivity constraints
  - Making the graph chordal
  - An improved procedure: consider polarity
- Algorithm II – Range-Allocation
  - What is the small-model property?
  - Finding a small adequate range (domain) to each variable
  - Reducing to Propositional Logic

Decision Procedures for Equality Logic

- We will first investigate methods that solve Equality Logic. Uninterpreted functions are eliminated with one of the reduction schemes.

- Our starting point: the E-Graph $G^E(\phi^E)$

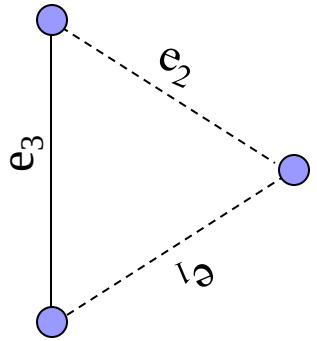
- Recall: $G^E(\phi^E)$ represents an abstraction of $\phi^{E:}$
  It represents ALL equality formulas with the same set of equality predicates as ϕ^E

From Equality to Propositional Logic

Bryant & Velev 2000: the *Sparse* method

$$\phi^{E} = x_1 = x_2 \wedge x_2 = x_3 \wedge x_1 \neq x_3$$

$$\phi_{enc} = e_1 \wedge e_2 \wedge \neg e_3$$

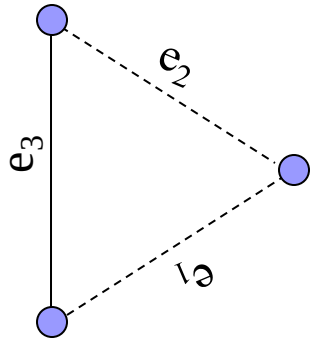


- Encode all edges with Boolean variables
  - (note: for now, ignore polarity)
  - This is an abstraction
  - Transitivity of equality is lost!
  - Must add transitivity constraints!

From Equality to Propositional Logic

$$\phi^{E} = x_1 = x_2 \wedge x_2 = x_3 \wedge x_1 \neq x_3$$

$$\phi_{enc} = e_1 \wedge e_2 \wedge \neg e_3$$



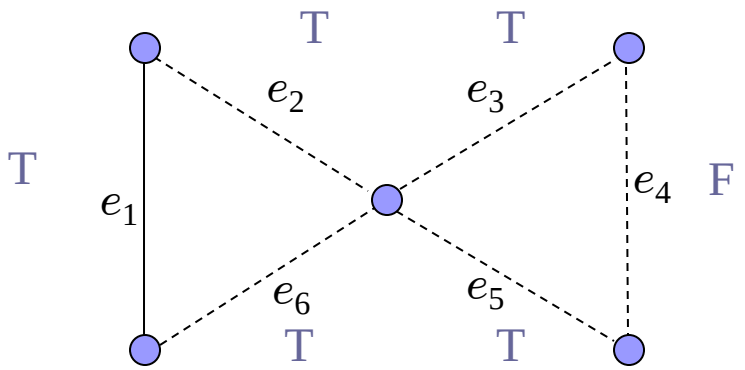
- For each cycle add a transitivity constraint

$$\phi_{trans} = (e_1 \wedge e_2 \rightarrow e_3) \wedge (e_1 \wedge e_3 \rightarrow e_2) \wedge (e_3 \wedge e_2 \rightarrow e_1)$$

Check: $\phi_{enc} \wedge \phi_{trans}$

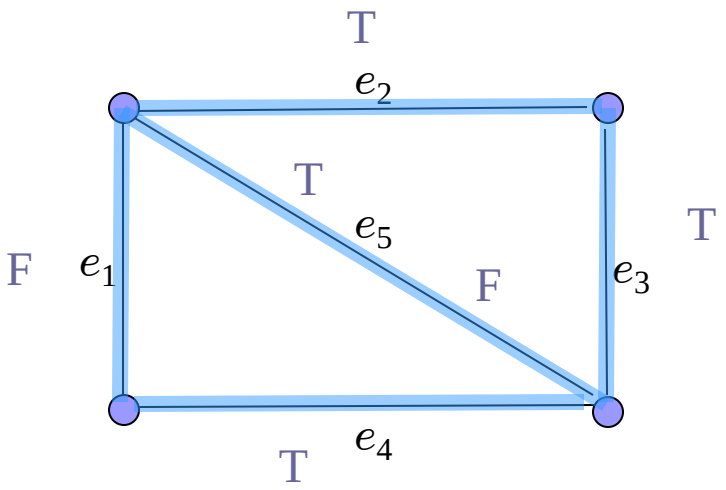
From Equality to Propositional Logic

- There can be an exponential number of cycles, so let's try to make it better.
- Thm: *it is sufficient to constrain simple cycles only*



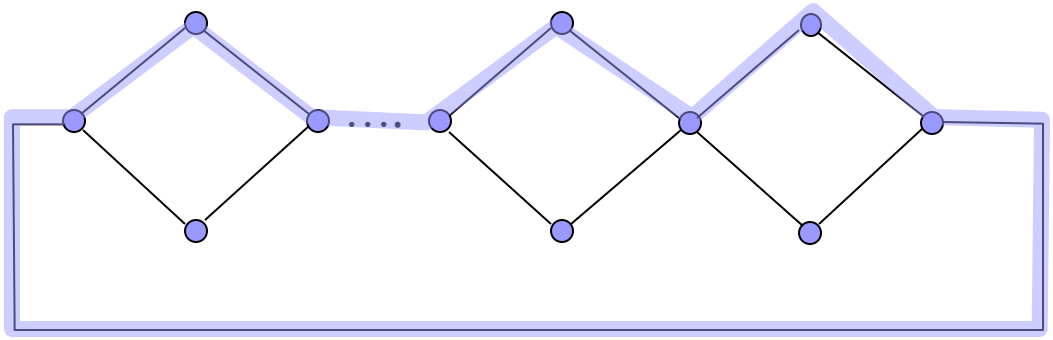
From Equality to Propositional Logic

- Still, there is an exponential number of simple cycles.
- Thm [Bryant & Velev]: *It is sufficient to constrain chord-free simple cycles*



From Equality to Propositional Logic

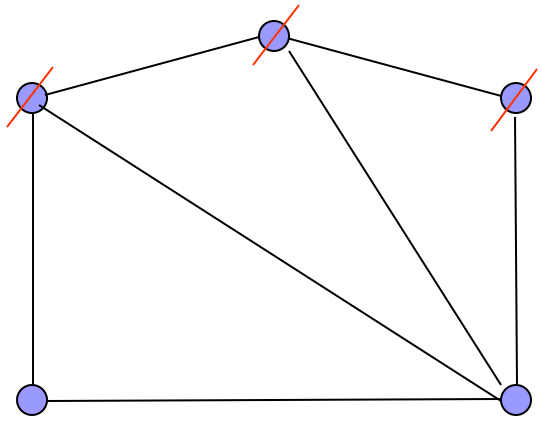
- Still, there can be an exponential number of chord-free simple cycles…



- Solution: make the graph ‘chordal’ by adding edges.

From Equality to Propositional Logic

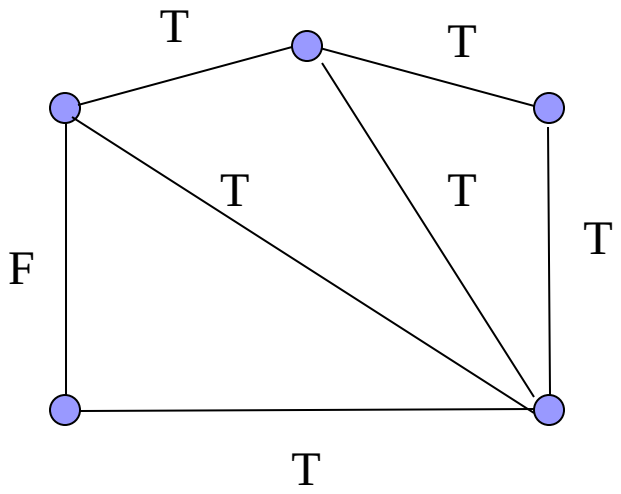
- Dfn: A graph is chordal iff every cycle of size 4 or more has a chord.
- How to make a graph chordal ? eliminate vertices one at a time, and connect their neighbors.



From Equality to Propositional Logic

- Once the graph is chordal, we can constrain only the triangles.

Contradiction!



- Note that this procedure adds not more than a polynomial # of edges, and results in a polynomial no. of constraints.

Suggested Readings

- Chapter 5, DP, Linear Arithmetic.
  - Each chapter in the book, has a different theory.
  - Couple this with the z3 solvers theories.
  - We will look at questions from EUF both in theory and solver.