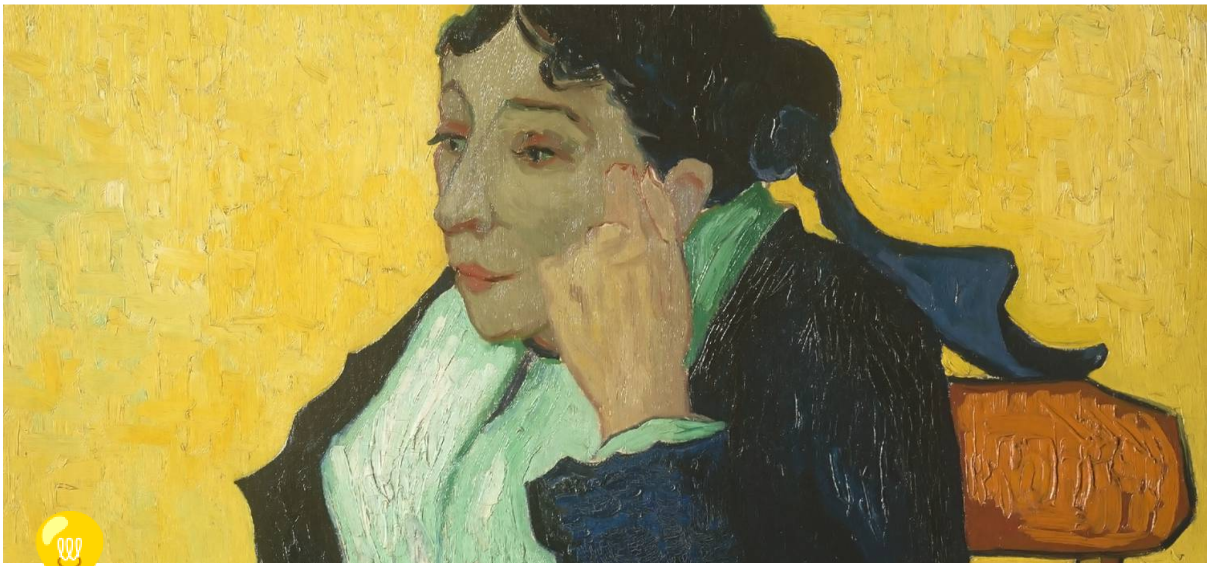


HOW-TO: Tasks in Eager SMT solving (a short selection)

> *In the following the story of a equality logic formula, that wanted to become a propositional logic formula so bad, is told. Featuring E-Graphs.*
>
> — @Piccola Radge

Table of Contents

Chapter 1: Simplifying equality logic formula with E-Graphs

Given a formula φ^{EQ} in Equality Logic, simplify the formula φ^{EQ} using the method presented in the lecture, based on ***polar*** E-graphs (equality graphs).

HOW-TO:

1. Draw the polar E-graph $G^E(\varphi^{EQ})$ of φ^{EQ} .

   (*Polar (or "with polarity") meaning there are two types of edges in the E-Graph distinguishing between equality and disequality: Notation for equality edge = straight line; notation for disequality edge = line with small vertical dashes through it or simply a dashed line.*)

2. Simplify the formula by simplifying its E-graph as follows:

   - If an *equality edge* is not part of a contradictory cycle*, remove that edge from the E-Graph. Removing that edge now corresponds to setting the corresponding **equation** in the formula φ^{EQ} to $true$.

     (**Those edges are mostly the ones "sticking out" of the E-Graph.*)

   - If an *disequality edge* (notation = line with vertical dashes) is not part of a contradictory cycle, remove that edge from the E-Graph. Removing that edge now corresponds to setting the corresponding **equation** in the formula φ^{EQ} to $false$.

     (*Here you need to really take care of the fact that you do not set the "whole" disequation corresponding to an disequality edge to $false$, since it's only a "negation of an equation". Meaning you do not set $a \neq b$ to $false$ but $a = b$ "in" $\neg(a = b)$. So essentially you'd get $a = b \equiv false \Rightarrow \neg(a = b) \equiv \neg false$*)

Chapter 2: Constructing satisfiability-equivalent propositional formula for equality logic formula using E-Graphs

1. Given a formula φ^{EQ} in Equality Logic and its **polar** E-Graph, construct the satisfiability-equivalent propositional logic formula for φ^{EQ} using the polar E-Graph.

HOW-TO:

For this task we need to use the **Sparse method** *based on polar E-Graphs* (**Algorithm 1** in lecture 08), which works in the following way:

1. Construct φ_{sk}, the *propositional **sk**eleton* of φ^{EQ}:

Replace all equalities in the formula by Boolean variables.

(E.g. for $a = b$ write "e_1" or "ab" or something like that; However **DO NOT FORGET**, that for the negation of $a = b$, i.e. $a \neq b$ or better $\neg(a = b)$ you need to write "$\neg e_1$" or "$\neg ab$")

2. For each simple contradictory cycle in the equality graph add a transitivity constraint:

$$\varphi^{pol}_{trans} = \bigwedge_{\text{simple contradictory cycle with edges } e_1,\ldots,e_n,\neg e} \left(\left(\bigvee_{i=1}^{\overset{n}{\cancel{k}}} \neg e_i \right) \vee e \right)$$

3. Conjunct φ_{sk} and φ^{pol}_{trans} , which then build the satisfiability-equivalent formula (lets call it φ_{prop}):

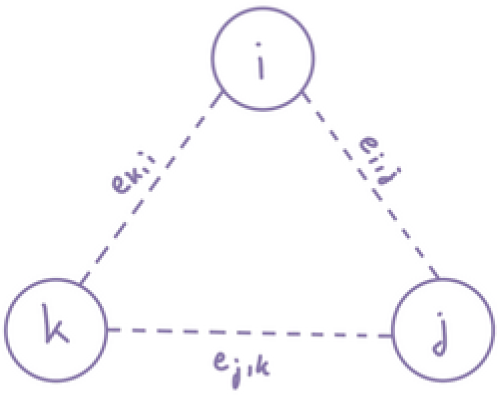
$$\varphi_{prop} = \varphi_{sk} \ \wedge \ \varphi^{pol}_{trans}$$

2. Given a formula φ^{EQ} in Equality Logic and its **non-polar** E-Graph, construct the satisfiability-equivalent propositional logic formula for φ^{EQ} using the polar E-Graph.

HOW-TO:

For this task we need to use the Algorithm *based on non-polar E-graphs* (**Algorithm 2** in lecture 08), which works in the following way:

1. Construct φ_{sk}, the *propositional* ***sk**eleton* of φ^{EQ}, by replacing each equality $t_i = t_j$ in φ^{EQ} by a fresh Boolean variable $e_{i,j}$.
2. (If not given in the task, first construct the non-polar E-Graph $G^E(\varphi^{EQ})$ for φ^{EQ})
3. Make $G^E(\varphi^{EQ})$ *chordal*.
4. Set $\varphi_{trans} = true$.
5. **For each** **triangle** “$(e_{i,j}, e_{j,k}, e_{k,i})$”



in $G^E(\varphi^{EQ})$:

$$\begin{aligned}\varphi_{trans} := \varphi_{trans} \quad & \wedge (e_{i,j} \wedge e_{j,k}) \to e_{k,i} \\ & \wedge (e_{i,j} \wedge e_{i,k}) \to e_{j,k} \\ & \wedge (e_{i,k} \wedge e_{j,k}) \to e_{i,j}\end{aligned}$$

6. Conjunct φ_{sk} and φ_{trans} , which then build the satisfiability-equivalent formula (lets call it again φ_{prop}):

$$\varphi_{prop} = \varphi_{sk} \quad \wedge \quad \varphi_{trans}$$

> ***End of story.***

Appendix

- **Simplifying Equality Logic Formula with E-Graph**

📌 **Definition (Contradictory cycle)**
Let $E_{\neq}$ be the set of disequality edges in an E-Graph $G^E(\varphi^{EQ})$ of some equality logic formula φ^{EQ}. A cycle with **exactly one** *disequality edge* from $E_{\neq}$ is a *contradictory cycle*.

📌 **Theorem**
Let S be the set of edges that are not part of any simple contradictory cycle.

Replacing

- all equations in φ^{EQ} that correspond to disequality edges in S with $false$,

and

- all equations in φ^{EQ} that correspond to equality edges in S with $true$, preserves satisfiability.

- **Constructing satisfiability-equivalent propositional formula for Equality Logic formula using E-Graphs**

📌 **Definition (Chordal graph)**

A graph is chordal iff every simple cycle over at least 4 different nodes has a chord.

I hereby indicate that all information in this document is without guarantee and I cannot be held liable for errors or incompleteness. For your convenience I refer to Lecture *08 Eager SMT solving: Equality logic and uninterpreted functions.*