Quantitative Congruence Closure

Extending an Equational Decision Procedure to Metric Reasoning

Paul-Gabriel Turcuman

8th of June 2026

Outline

From Equality to Distance

Classical equational reasoning asks a **yes**/**no** question:

$$s \overset{?}{\approx}_G t \qquad \text{(are } s \text{ and } t \text{ equal modulo the ground equations } G?)$$

From Equality to Distance

Classical equational reasoning asks a **yes/no** question:

$$s \overset{?}{\approx}_G t \qquad \text{(are } s \text{ and } t \text{ equal modulo the ground equations } G?)$$

But many modern settings demand **quantitative** answers:

- **Probabilistic programs:** small perturbations break equality but not *closeness*.
- **Privacy & security:** sensitivity of a computation is a *distance*.
- **DNA / edit distances:** molecules are measurably similar rather than identical.

From Equality to Distance

Classical equational reasoning asks a **yes/no** question:

$$s \overset{?}{\approx}_G t \qquad \text{(are } s \text{ and } t \text{ equal modulo the ground equations } G?)$$

But many modern settings demand **quantitative** answers:

- **Probabilistic programs:** small perturbations break equality but not *closeness*.
- **Privacy & security:** sensitivity of a computation is a *distance*.
- **DNA / edit distances:** molecules are measurably similar rather than identical.

The quantitative question

Given a set E of quantitative ground equations and query terms s, t: find the *smallest* ε such that s and t can be proved to be at most ε-apart from E using the rules of the quantitative equational theory (shown on slide 12).

Why Congruence Closure Matters

Congruence closure (CC) is a core engine for equality reasoning.

- Standard component in **SMT solvers** for the theory of equality with uninterpreted function symbols.
- **Incremental**: new equations can be integrated without restarting from scratch.
- Bridges syntactic equality and semantic congruence under function symbols.

Why this matters here

The quantitative version should preserve the same structural advantages: locality, incremental update, and compositional propagation of bounds.

What the literature gives us:

1. **Nelson-Oppen (1980):** CC via MERGE on a shared DAG; predecessor lists for detecting new congruences; union-find (Tarjan) for equivalence classes. Worst-case $O(m^2)$; average $O(m \log m)$ using a hash table on predecessor signatures (Downey-Sethi-Tarjan).
2. **Bachmair-Tiwari-Vigneron (2003):** *abstract* CC as ground completion; D-rules ($f(c_1, \ldots, c_k) \to c$) name equivalence classes, C-rules ($c \to d$) encode them. All known algorithms are *strategies* over the same transition rules.
3. **Nieuwenhuis-Oliveras (2007):** clean incremental CC; preprocessing by Curryfication and flattening (done once); Merge/Propagate via Use lists and Lookup table; $O(n \log n)$.

Challenge for the quantitative setting

Classical CC relies on equivalence partitions and union-find. In quantitative CC, terms carry *real-valued distances*, so a different data structure will be needed for efficient implementations.

Let G be a finite set of ground equalities over $\mathcal{T}(\mathcal{F}, \mathcal{V})$. The relation $\approx_G$ is the smallest congruence on $\mathcal{T}(\mathcal{F}, \mathcal{V}) \times \mathcal{T}(\mathcal{F}, \mathcal{V})$ closed under:

$$\frac{(s \approx t) \in G}{G \vdash s \approx t} \qquad \frac{}{G \vdash t \approx t}$$

$$\frac{G \vdash s \approx t}{G \vdash t \approx s} \qquad \frac{G \vdash s \approx t \quad G \vdash t \approx u}{G \vdash s \approx u}$$

Congruence (for each $f \in \mathcal{F}^{(n)}$):

$$\frac{G \vdash s_1 \approx t_1 \quad \cdots \quad G \vdash s_n \approx t_n}{G \vdash f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n)}$$

Word problem: given finite ground G and terms s, t, decide $s \approx_G t$.

Setting. G finite ground equalities; $S := \mathrm{Subterms}(G) \cup \mathrm{Subterms}(s) \cup \mathrm{Subterms}(t)$.

Closure operators on a set G:

$$R(G) := \{(t,t) \mid t \in \mathcal{T}(\mathcal{F},\mathcal{V})\}$$
$$S(G) := \{(s,t) \mid (t,s) \in G\}$$
$$T(G) := \{(s,w) \mid \exists r\colon (s,r),(r,w) \in G\}$$
$$C(G) := \big\{(f(s_1,\ldots,s_n), f(t_1,\ldots,t_n)) \mid (s_i,t_i) \in G \text{ for each } i\big\}$$

$$\mathrm{Cong}(G) := G \cup R(G) \cup S(G) \cup T(G) \cup C(G)$$

Decision procedure (set-based):

Iterate:

$$H_0 := G, \quad H_{i+1} := \mathrm{Cong}(H_i) \cap (S \times S)$$

The sequence stabilises; its fixed point is denoted $\mathrm{CC}_S(G)$.

Decision

$$s \approx_G t \iff (s,t) \in \mathrm{CC}_S(G)$$

Two classical implementations:

Congruence-class algorithm

- Represent subterms as a partition into equivalence classes; $[t]$ denotes the class of t.
- For each $l \approx r \in G$: Merge$([l], [r])$.
- Propagate: if $[s_i] = [t_i]$ for all i from 1 to n, then merge $[f(s_1, \ldots, s_n)]$ and $[f(t_1, \ldots, t_n)]$.
- Answer: $s \approx_G t \iff [s] = [t]$.

DAG algorithm (Nelson-Oppen)

- Build a shared DAG of subterms.
- Find/Union maintains classes.
- Union updates representatives (`repr`) and predecessor lists.
- Congruent(u, v): same label and congruent arguments.
- Predecessor lists detect new congruences.

Our question

How do these algorithms generalise when equalities carry *distances*?

Definition. A *unital quantale* is a tuple $(\mathbb{Q}, \leq, k, \otimes)$ such that:

- $(\mathbb{Q}, \leq)$ is a **complete lattice** (has all suprema and all infima);
- $(\mathbb{Q}, k, \otimes)$ is a monoid with unit k;
- $\otimes$ distributes over arbitrary joins from both sides:

$$x \otimes \bigvee_{i \in I} y_i = \bigvee_{i \in I} (x \otimes y_i), \qquad \bigvee_{i \in I} y_i \otimes x = \bigvee_{i \in I} (y_i \otimes x).$$

k is the unit and $\otimes$ is the tensor of the quantale. The bottom element $\bot = \bigvee_{\emptyset}$ is the join of the empty set.

The Lawvere Quantale

We work over the **Lawvere quantale**

$$\mathbb{L} = ([0, \infty], \geq, 0, +)$$

Key structure

- **Carrier:** extended non-negative reals $[0, \infty]$.
- **Order:** $\geq$ (numerically), so 0 is the *top* element (zero distance means identical terms). The element ∞ is the *bottom* (terms are not close at all).
- **Tensor:** addition $\varepsilon \otimes \delta = \varepsilon + \delta$ (encodes the triangle inequality).
- **Unit:** 0 $\quad (d(x, x) = 0)$.
- **Joins** (w.r.t. $\geq$): infima **Meets**: suprema.

Traditional equality $\longleftrightarrow$ zero distance: $\varepsilon = 0$

Quantitative Ground Equations

A **quantitative equation** $\varepsilon \vdash s \approx t$ reads: *"s and t are at most ε-apart."*

A finite set $E = \{\varepsilon_i \vdash s_i \approx t_i\}_{i=1}^{n}$ of **quantitative ground equations** is our input.

The quantitative word problem: given ground terms s, t, find the smallest $\varepsilon \in [0, \infty]$ such that $\varepsilon \vdash s \approx t$ is *derivable* from E (i.e., provable by the rules on slide 12).

Quantitative Ground Equations

A **quantitative equation** $\varepsilon \vdash s \approx t$ reads: *"s and t are at most ε-apart."*

A finite set $E = \{\varepsilon_i \vdash s_i \approx t_i\}_{i=1}^{n}$ of **quantitative ground equations** is our input.

The quantitative word problem: given ground terms s, t, find the smallest $\varepsilon \in [0, \infty]$ such that $\varepsilon \vdash s \approx t$ is *derivable* from E (i.e., provable by the rules on slide 12).

Example

$E = \{2 \vdash f(a) \approx a,\ 1 \vdash a \approx b\}$. Starting from these two equations:

- Congruence applied to $1 \vdash a \approx b$: $1 \vdash f(a) \approx f(b)$
- Symmetry of the first equation: $2 \vdash a \approx f(a)$
- Transitivity ($1 + 2 = 3$): $3 \vdash a \approx f(b)$

The best provable bound for the pair $(a, f(b))$ is $\varepsilon = 3$.

Unlike the Boolean case, the answer is a *real number* in $[0, \infty]$, not just yes/no.

Ground setting: all terms are ground (no variables).
The relation $\approx_E$ is the smallest relation closed under:

$$\frac{\varepsilon \vdash u \approx v \in E}{\varepsilon \vdash u \approx v}\ (\text{ax})$$

$$\frac{}{0 \vdash u \approx u}\ (\text{refl}) \qquad \frac{\varepsilon \vdash u \approx v}{\varepsilon \vdash v \approx u}\ (\text{sym})$$

$$\frac{\varepsilon \vdash u \approx v \quad \delta \vdash v \approx w}{\varepsilon + \delta \vdash u \approx w}\ (\text{trans})$$

$$\frac{\varepsilon_1 \vdash u_1 \approx v_1 \ \cdots \ \varepsilon_n \vdash u_n \approx v_n}{\varepsilon_1 + \cdots + \varepsilon_n \vdash f(u_1, \ldots, u_n) \approx f(v_1, \ldots, v_n)}\ (\text{cong})$$

$$\frac{\varepsilon \vdash u \approx v \quad \varepsilon \leq \delta}{\delta \vdash u \approx v}\ (\text{weak})$$

$$\frac{\delta \vdash u \approx v \text{ for all } \delta \ll \varepsilon}{\varepsilon \vdash u \approx v}\ (\text{Arch})$$

$$\frac{\varepsilon_i \vdash u \approx v \text{ for each } i \in I}{\inf_{i \in I} \varepsilon_i \vdash u \approx v}\ (\text{join})$$

Notes. No substitution rule is needed: we work entirely with ground terms. The join rule selects the *tightest* provable bound for any given pair; it has no Boolean analogue (in the Boolean case all bounds collapse to 0 or ∞).

The join rule from the theory has a direct algorithmic counterpart:

Definition

$$J(E) := \left\{ \inf \{\varepsilon \mid \varepsilon \vdash u \approx v \in E\} \vdash u \approx v \;\middle|\; \exists \varepsilon < \infty : \; \varepsilon \vdash u \approx v \in E \right\}$$

Effect:

- Exactly one equation per ordered pair (u, v), carrying the smallest available bound.
- Since E is finite, the infimum is over a *finite non-empty* set of reals in $[0, \infty]$, hence it is always attained. It equals ∞ when every known bound for (u, v) is ∞ (meaning: no finite relation has been derived for that pair yet).
- The infimum here is the classic infimum with $\leq$ order.

Quantitative Closure Operators

Lift each Boolean closure operator to carry *quantities*:

$$R(E) := \{0 \vdash u \approx u \mid u \in \mathcal{T}(\mathcal{F}, \mathcal{V})\}$$
$$S(E) := \{\varepsilon \vdash v \approx u \mid \varepsilon \vdash u \approx v \in E\}$$
$$T(E) := \left\{\varepsilon_1 + \varepsilon_2 \vdash u \approx w \mid \exists v : \ \varepsilon_1 \vdash u \approx v \in E,\ \varepsilon_2 \vdash v \approx w \in E\right\}$$
$$C(E) := \left\{\textstyle\sum_{i=1}^{n} \varepsilon_i \vdash f(u_1, \ldots, u_n) \approx f(v_1, \ldots, v_n) \;\middle|\; f \in \mathcal{F}^{(n)},\ \varepsilon_i \vdash u_i \approx v_i \in E \text{ for each } i = 1, \ldots, n\right\}$$

$$\mathrm{Cong}(E) \ := \ J\big(E \cup R(E) \cup S(E) \cup T(E) \cup C(E)\big)$$

Quantitative Closure Operators

Lift each Boolean closure operator to carry *quantities*:

$$R(E) := \{0 \vdash u \approx u \mid u \in \mathcal{T}(\mathcal{F}, \mathcal{V})\}$$
$$S(E) := \{\varepsilon \vdash v \approx u \mid \varepsilon \vdash u \approx v \in E\}$$
$$T(E) := \left\{\varepsilon_1 + \varepsilon_2 \vdash u \approx w \mid \exists v : \ \varepsilon_1 \vdash u \approx v \in E,\ \varepsilon_2 \vdash v \approx w \in E\right\}$$
$$C(E) := \left\{\textstyle\sum_{i=1}^{n} \varepsilon_i \vdash f(u_1, \ldots, u_n) \approx f(v_1, \ldots, v_n) \;\middle|\; f \in \mathcal{F}^{(n)},\ \varepsilon_i \vdash u_i \approx v_i \in E \text{ for each } i = 1, \ldots, n\right\}$$

$$\mathrm{Cong}(E) \ := \ J\big(E \cup R(E) \cup S(E) \cup T(E) \cup C(E)\big)$$

Key point

J is applied to the *whole* generated set before storing anything. This ensures the sharpest bound survives after all new distances are produced at each step. Bounds can only *decrease* (or remain the same) across iterations.

Algorithm

Input: finite set E of quantitative ground equations; query terms s, t.

1. Set $S := \mathrm{Subterms}(E) \cup \mathrm{Subterms}(s) \cup \mathrm{Subterms}(t)$.
2. Set $H_0 := J(E)$.
3. For $i \geq 0$: $H_{i+1} := \mathrm{Cong}(H_i) \cap \big([0, \infty) \times S \times S\big)$.
4. Repeat until $H_{i+1} = H_i$; call the fixed point $\mathrm{QCC}_S(E)$.
5. If $\varepsilon \vdash s \approx t \in \mathrm{QCC}_S(E)$, return ε; else return ∞.

Termination: **proved** Soundness & completeness: **work in progress**

Why Apply J at the Start and at Every Step

Example. $E = \{2 \vdash a \approx b,\ 5 \vdash a \approx b,\ 3 \vdash b \approx c\}$, query: best bound for (a, c).

Start from $H_0 = J(E)$:

- $H_0 = \{2 \vdash a \approx b,\ 3 \vdash b \approx c\}$ (redundant $5 \vdash a \approx b$ discarded)
- $T(H_0)$ gives $5 \vdash a \approx c$ $\quad (2 + 3)$
- **Optimal answer: $\varepsilon = 5$, one step**

Start from $H_0 = E$ (no prior join):

- $T(H_0)$ gives both $5 \vdash a \approx c$ $(2 + 3)$ and $8 \vdash a \approx c$ $(5 + 3)$
- J then discards 8, keeps 5
- Same final answer, but redundant work generated

Why Apply J at the Start and at Every Step

Example. $E = \{2 \vdash a \approx b,\ 5 \vdash a \approx b,\ 3 \vdash b \approx c\}$, query: best bound for (a, c).

Start from $H_0 = J(E)$:

- $H_0 = \{2 \vdash a \approx b,\ 3 \vdash b \approx c\}$ (redundant $5 \vdash a \approx b$ discarded)
- $T(H_0)$ gives $5 \vdash a \approx c \quad (2 + 3)$
- **Optimal answer: $\varepsilon = 5$, one step**

Start from $H_0 = E$ (no prior join):

- $T(H_0)$ gives both $5 \vdash a \approx c$ $(2 + 3)$ and $8 \vdash a \approx c$ $(5 + 3)$
- J then discards 8, keeps 5
- Same final answer, but redundant work generated

Why the difference matters in general:

Efficiency: weaker bounds produce more weaker distances (e.g. $5 + 3 = 8$ from $5 \vdash a \approx b, 3 \vdash b \approx c$, as opposed to $2 + 3 = 5$, from $2 \vdash a \approx b, 3 \vdash b \approx c$), which then propagate further and create unnecessary work.

Termination: without J, transitivity may generate an infinite sequence of larger (weaker) bounds $(5, 8, 11, \ldots)$ for the same pair. J keeps only the smallest bound, guaranteeing termination.

Worked Example

$E = \{2 \vdash f(a) \approx a,\ 1 \vdash a \approx b\}$, query: $? \vdash f(b) \approx a$
$S = \{a, b, f(a), f(b)\}$, $H_0 = J(E) = E$.

H_1: Apply Cong, restrict to $[0, \infty) \times S \times S$:

- R: $0 \vdash a \approx a,\ 0 \vdash b \approx b,\ 0 \vdash f(a) \approx f(a),\ 0 \vdash f(b) \approx f(b)$
- S: $1 \vdash b \approx a,\ 2 \vdash a \approx f(a)$
- T: $3 \vdash f(a) \approx b$ $(2 + 1)$
- C: $1 \vdash f(a) \approx f(b)$ (congruence from $1 \vdash a \approx b$)

H_2: New entries include $1 \vdash f(b) \approx f(a),\ 3 \vdash a \approx f(b),\ 3 \vdash b \approx f(a)$

H_3: From $1 \vdash f(b) \approx f(a)$ and $2 \vdash f(a) \approx a$ via transitivity $(1 + 2 = 3)$:

$$\boxed{3 \vdash f(b) \approx a}$$

Also: $4 \vdash b \approx f(b),\ 4 \vdash f(b) \approx b$ (both still in H_3). Then $H_4 = H_3 = \mathrm{QCC}_S(E)$.

Answer: $\varepsilon = 3$.

Theorem (Termination)

There exists $m \geq 0$ such that $H_{m+1} = H_m$.

- The algorithm restricts derivations strictly to pairs from the finite set of subterms ($S \times S$).
- The Join operator (J) guarantees that the distance bound for any given pair is non-increasing.
- Because all derived distances are finite sums of the initial positive quantities found in E, infinite strict improvements below any bound cannot happen.

Expected statement

$$\mathrm{QCC}_S(E) \;=\; J(\approx_E) \cap ([0,\infty) \times S \times S)$$

Why $J(\approx_E)$ and not $\approx_E$?

- $\approx_E$ contains every derivable bound: it is an infinite set.
- For each pair (u, v), exactly the single tightest provable bound is held.

Open tasks

Soundness: every equation in $\mathrm{QCC}_S(E)$ is contained in $J(\approx_E)$.

Completeness: for every derivable $\gamma \vdash u \approx v$ in $J(\approx_E)$, with $u, v \in S$, it follows that $\varepsilon \vdash u \approx v$ in QCC_S, with $\varepsilon \leq \gamma$.

Three Quantitative Settings

Three natural generalisations, in increasing complexity:

Non-expansive *(current work)*

Ground equations E; for every n-ary symbol f:
$d\big(f(u_1, \ldots, u_n),\, f(v_1, \ldots, v_n)\big) \leq \sum_{i=1}^{n} d(u_i, v_i)$. Application of functions does not amplify distances.

Graded

Ground equations E; each n-ary symbol f has coefficients $q_1, \ldots, q_n \geq 0$:
$d\big(f(u_1, \ldots, u_n),\, f(v_1, \ldots, v_n)\big) \leq \sum_{i=1}^{n} q_i\, d(u_i, v_i)$. Controls how $C(E)$ propagates distances.

Free variables *(possible extension)*

Allow a literal L with variables; find a substitution σ and distance ε such that, after substituting variables, the system E can prove the instantiated literal $L\sigma$ holds with distance at most ε.

The Road Ahead

Two Boolean procedures $\times$ three quantitative settings:

	Non-expansive	Graded	Free variables
Set-based	In progress	To do	-
Congr. classes	To do	To do	Open
DAG	To do	To do	Open

Research priority

1. Soundness & completeness for set-based (non-expansive)
2. Non-expansive: congruence-class + DAG algorithms
3. Graded: all variants
4. Free-variable extension: if time allows