Direct Combination of Completion and Congruence Closure

- Christelle Scharff, Pace University, NY, USA
- Leo Bachmair, SUNY Stony Brook, NY, USA

Outline

- Word problem
- Completion versus Congruence Closure for ground equalities
- SOUR graph Completion
- Abstract Congruence Closure
- "SER" graphs
- Abstract Congruence Closure by graph transformations
- Correctness results
- Convergent rewrite system over the original signature
- Conclusion and perspectives

Word problem

- Given a set of equalities E and a goal $s \approx t$, is $s \approx t$ true in all models of E?

  Word problem is undecidable.

- Interest here: **E is a ground set of equalities and** $s \approx t$ **is a ground equality.**

- Word problem is decidable when dealing with ground equalities.

- Solutions: 2 distinct approaches
  - Completion [Knuth Bendix 1970]
  - Congruence Closure [Kozen 1977, Downey Sethi Tarjan 1980,Nelson Oppen 1980, Shostak 1984]

Completion versus congruence closure

- Completion – Compilation of a set of equalities into a set of convergent directed rules.
- Congruence Closure – of a relation (on a graph).
- Completion is in general not as efficient as Congruence Closure.

  – O(n log n) ground Completion method [Snyder 1993].

Ground SOUR graphs Completion

Lynch Strogova 1995

- First graph-based implementation of Completion.
- Equalities representing by a graph with maximal structure sharing (DAG).
  - 1 Vertex = 1 Term
  -

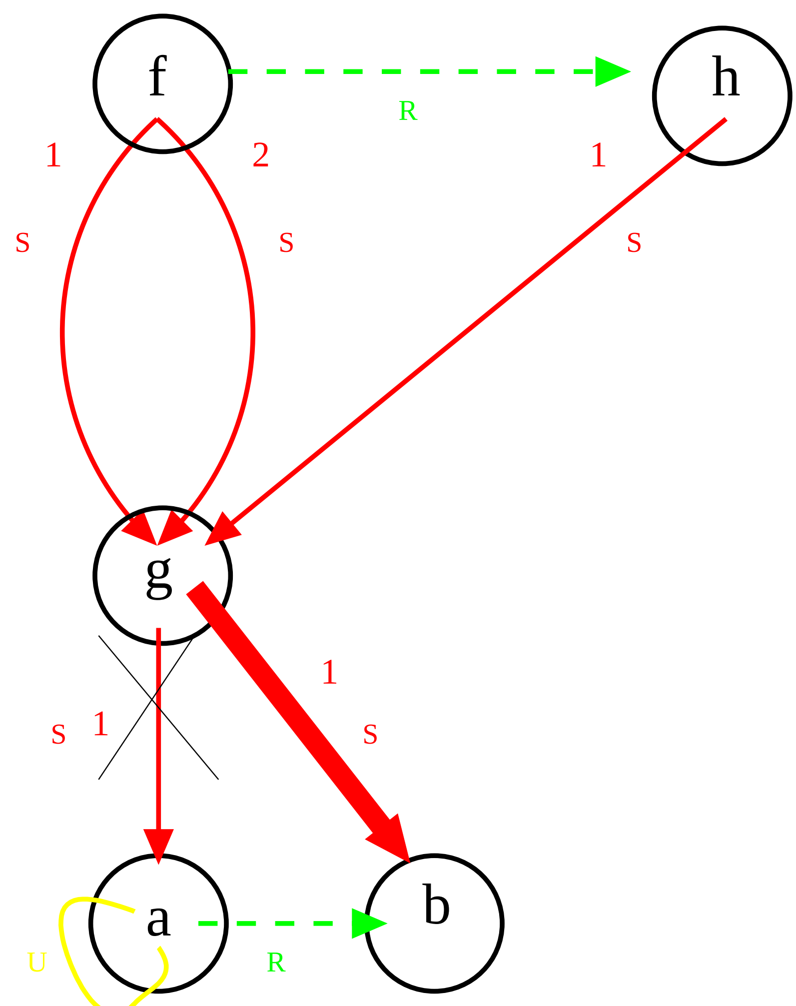
$$Edges = \begin{cases} \textit{Subterm} - \mathbf{S} \\ \textit{Orientation} - \mathbf{O} \\ \textit{Unification} - \mathbf{U} \\ \textit{Rewriting} - \mathbf{R} \end{cases}$$

- Completion = Saturation of the graph by graph transformations
  - Seek edge patterns

Example

$$E = \begin{cases} f(g(a), g(a)) \approx h(g(a)) \\ a \approx b \end{cases}$$

$$f \succ g \succ h \succ a \succ b$$



Abstract Congruence Closure

Bachmair Tiwari Vigneron 2000

- Construction of the congruence closure of a relation on a graph.
- Introduction of symbols and extension of the signature with constants to abstractly represent sharing.
- Convergent rewrite system over an **extended** signature.

“SER” graphs

- Simpler version of SOUR graphs.
  - No Unification edges.
  - No Orientation edges (efficiency).

- Vertex labeled by a symbol of the original signature (Σ) and a constant of K ($\Sigma \cap K = \emptyset$).

$$Edges = \begin{cases} \textit{Subterm} - \mathbf{S} \\ \textit{Equality} - \mathbf{E} \\ \textit{Rewriting} - \mathbf{R} \end{cases}$$

  - Advantage: No Orientation edges.
  - Disadvantage: Convergent rewrite system over $\Sigma \cup K$.

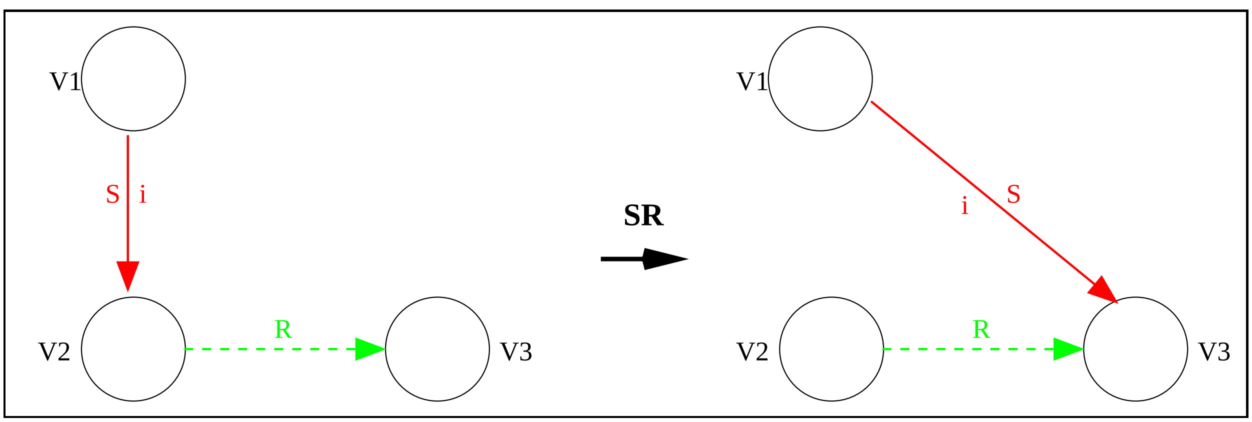
- Each vertex represents a term over the extended signature $\Sigma \cup K$ and over Σ (but **not directly**).

- It may make no sense to interpret rewrite rules over the original signature.

Abstract Congruence Closure

Implemented by graph transformations

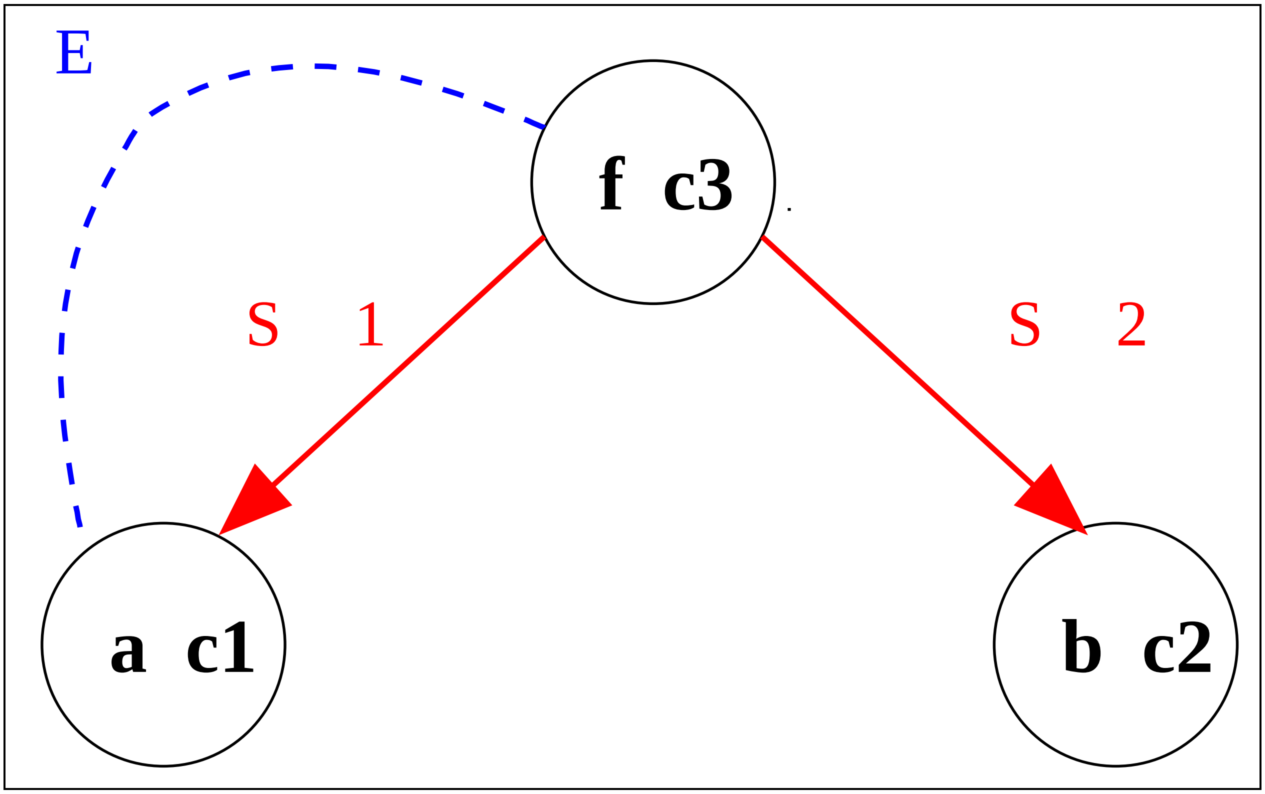
- Simpler version of the SOUR graph rules.
- Four rules:
  - **Orient** – an equality,
  - **SR, RR** – Critical Pairs/Simplification
  - **Merge** – to ensure closure under congruence.
- **SR rule:**



- Rules can be expressed by **Transition rules:**
  $(E_s,\ E_{eq},\ E_r,\ V,\ KC) \rightarrow (E_s',\ E_{eq}',\ E_r',\ V',\ KC')$

Example (1)

- $E = \{f(a,b) \approx a\}$, $\Sigma = \{f,\ a,\ b\}$



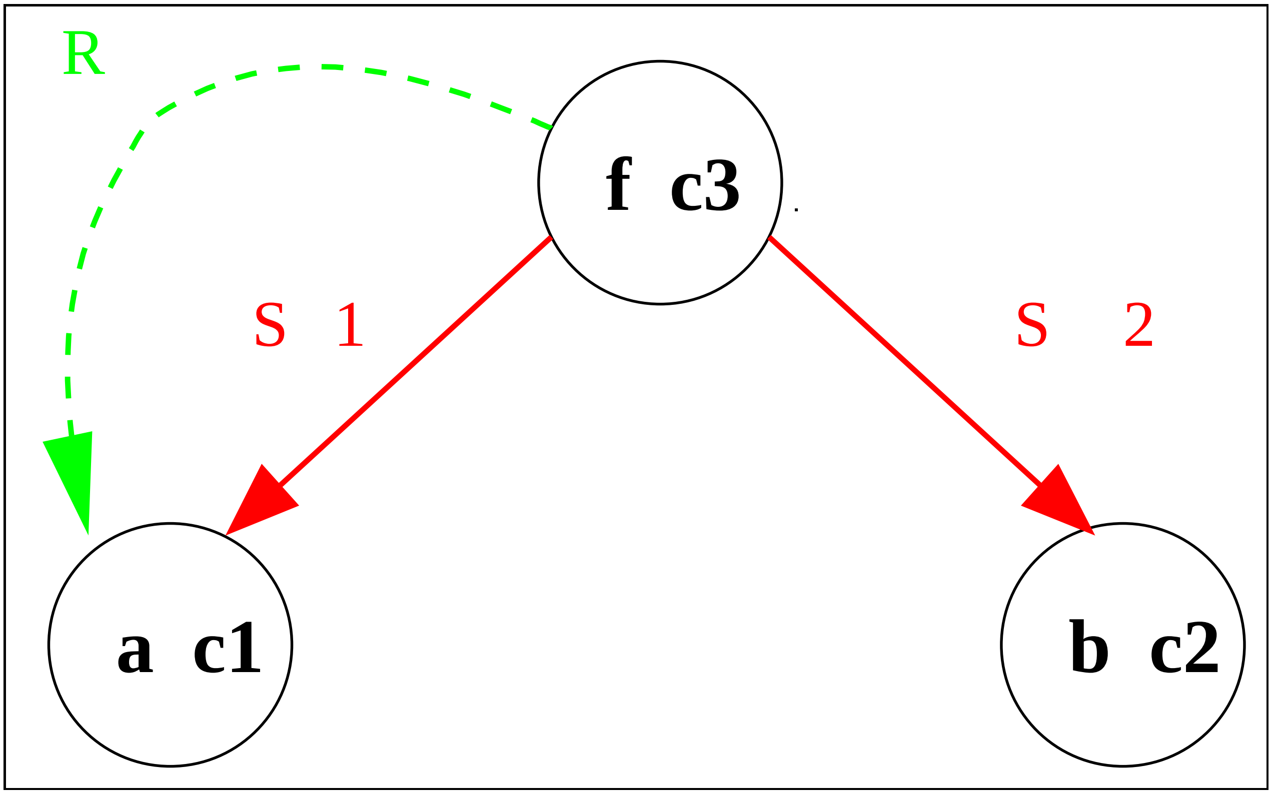
- Reading the graph:
  - Over the original signature: E
  - Over the extended signature, $\Sigma \cup K$, where $K = \{c1,\ c2,\ c3\}$

  D-rules: $a \rightarrow c1$, $b \rightarrow c2$, $f(c1, c2) \rightarrow c3$

  C-equalities: $c3 \approx c1$

Example (2)

Application of Orient - Case 1



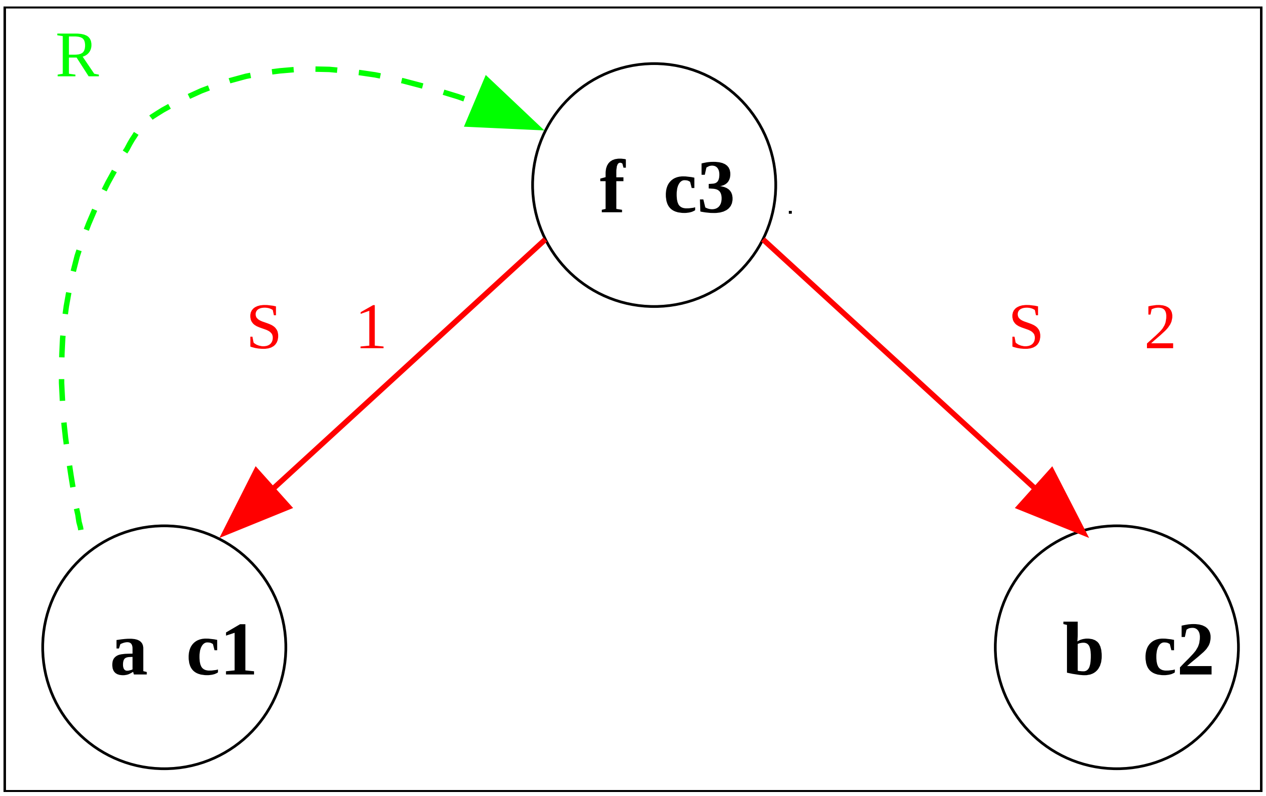
$c3 \succ c1$

Convergent rewrite system over the extended signature:

$a \rightarrow c1,\ b \rightarrow c2,\ f(c1, c2) \rightarrow c3,\ c3 \rightarrow c1$

Example (3)

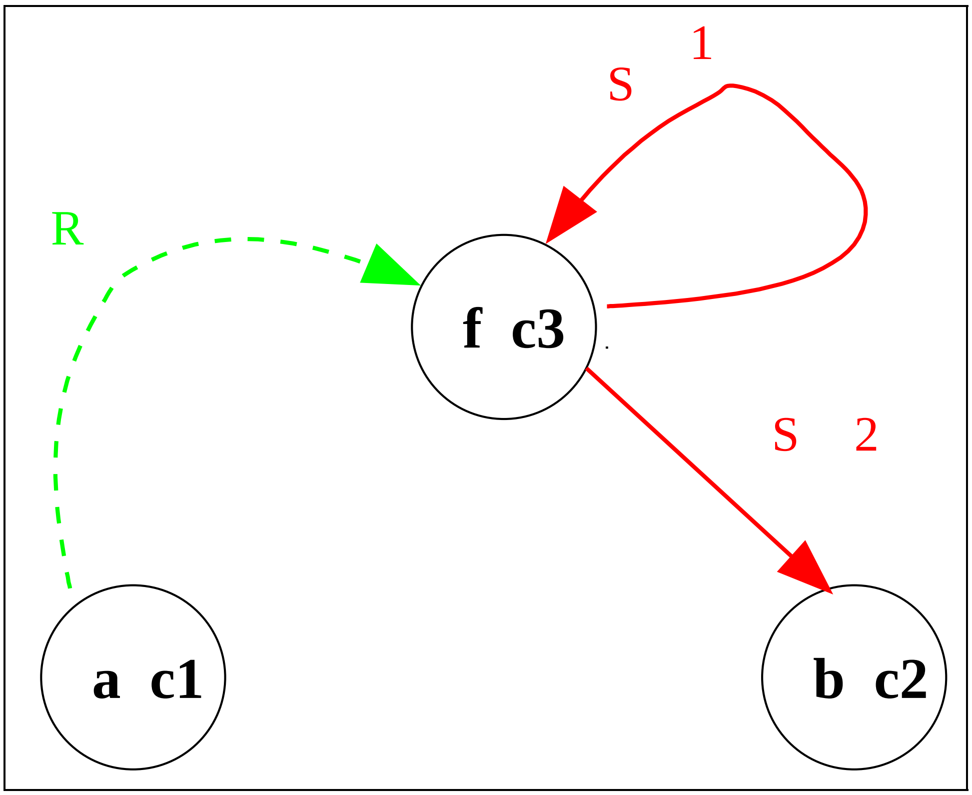
Application of Orient - Case 2



$c1 \succ c3$

Example (4)

Application of SR - Case 2



$$c1 \succ c3$$

Convergent rewrite system over the extended signature:

$a \to c1$, $b \to c2$, $f(c3, c2) \to c3$, $c1 \to c3$

– No rewrite system on the original signature.

Correctness results

- **Termination:** Exhaustive application of the transformation rules terminates.
- **Soundness:** Exhaustive application of the transformation rules is sound in that the equational theory represented over Σ-terms does not change.
- **Completeness:** Exhaustive application of the transformation rules is complete in the sense that the final rewrite system over the extended signature is convergent.
- **Abstract congruence closure**.

Convergent rewrite system over the original signature

- Abstract congruence closure produces a convergent rewrite system over the extended signature.

- To obtain a convergent rewrite system over the original signature the saturated graph must be further transformed.

- Transformation of the saturated graph using a graphical implementation of the **Compression** and **Selection rules**.
  - Eliminate constant from K
  - Redirect R edges.

Conclusion and perspectives

- Combination of different approaches to solve the word problem.
- Implementation.
- Complexity.
- Incremental abstract Congruence Closure in the described graph-based framework.
- Explore the connection of unification and Congruence Closure in the described graph-based framework.