Direct Combination of Completion and Congruence Closure

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

Outline

- Word problem
- Completion
- Abstract Congruence Closure
- Completion versus Congruence Closure for ground equalities
- SOUR graph Completion
- “SER” graphs
- Abstract Congruence Closure by graph transformations
- Correctness results
- Convergent rewrite system over the original signature
- Correctness results
- 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 confluent and terminating 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].

  – The standard completion approach is quadratic in the worst case [Plaisted Sattler-Klein 1996].

Completion

- **Critical Pair Inference Rule** (CP)

$$\frac{u[s'] \approx v \qquad s \approx t}{\sigma(u[t] \approx v)}$$

  - s' is not a variable, and
  - $\sigma = mgu(s =^? s')$ (unification)

Example

- An additive group G is defined by the set of equalities:

$$\begin{aligned} x + e &\approx x \\ x + (y + z) &\approx (x + y) + z \\ x + i(x) &\approx e \end{aligned}$$

- How to check that:

$$i(x + y) \approx i(y) + i(x)$$

Example (cont)

- **An equivalent deterministic term rewrite system G_∞:**

$$
\begin{array}{rcl}
x + e & \to & x \\
e + x & \to & x \\
x + (y + z) & \to & (x + y) + z \\
x + i(x) & \to & e \\
i(x) + x & \to & e \\
i(e) & \to & e \\
(y + i(x)) + x & \to & y \\
(y + x) + i(x) & \to & y \\
i(i(x)) & \to & x \\
i(x + y) & \to & i(y) + i(x)
\end{array}
$$

- The proof of $i(x + y) \approx i(y) + i(x)$ is then obvious.

Ground SOUR graphs Completion

Lynch Strogova 1995

- First graph-based implementation of Completion.
- Equalities representing by a graph
  - 1 Vertex = 1 Term
  -

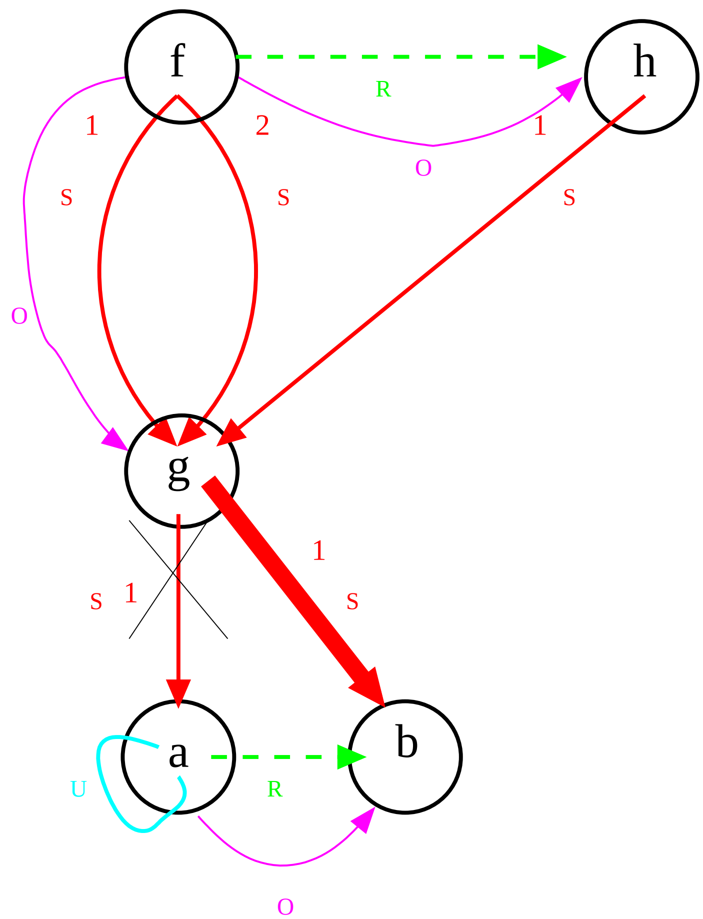
$$Edges = \begin{cases} \textit{Subterm} - \mathbf{S} \\ \textit{Orientation} - \mathbf{O} \\ \textit{Unification} - \mathbf{U} \\ \textit{Rewriting} - \mathbf{R} \end{cases}$$

- 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.
  - In general it uses a compact representation of the given terms by a directed acyclic graph [Kozen 1977, Downey Sethi Tarjan 1980, Nelson Oppen 1980, Shostak 1984].

- Introduction of symbols and extension of the signature ($\Sigma \cup \mathcal{K}$) with constants to abstractly represent sharing.

- Set of inference rules to construct the abstract congruence closure (Extension, Simplification, Orientation, Deletion, Deduction, Collapse, Composition)

- **A** convergent rewrite system over an **extended** signature $\Sigma \cup \mathcal{K}$.
  - Transitions: $(\mathcal{K}, E, R)$.

- Set of inference rules to construct the convergent rewrite system over the initial signature (Selection, Compression) that eliminate the constants of $\mathcal{K}$.

- **A** convergent rewrite system over the **initial** signature Σ
  - Transitions: $(\mathcal{K}, R)$.

Selection and Compression

- **Compression Rule**

$$\frac{(\mathcal{K} \cup \{c_0\},\ R \cup \{c_0 \to t\})}{(\mathcal{K},\ R < c_0 \mapsto t >)}$$

where c_0 is a redundant constant, $< c_0 \mapsto t >$ denotes the (homomorphic extension of the) mapping $c_0 \mapsto t$ and $R < c_0 \mapsto t >$ denotes the application of this homomorphism to each term in the set R.

- **Selection Rule**

$$\frac{(\mathcal{K} \cup \{c_0\},\ R \cup \{t \to c_0\})}{(\mathcal{K},\ R < c_0 \mapsto t >)}$$

where $t \in \mathcal{T}(\Sigma)$ and $c_0 \in \mathcal{K}$ is not redundant in R.

Example

- $E = \{f(a,b) \approx a\}$, $\Sigma = \{f,\ a,\ b\}$
- Transformation of E:
  - $R' = \{a \to c1,\ b \to c2,\ f(c1,c2) \to c3\}$
  - $E' = \{c3 \approx c1\}$
  - $\mathcal{K}' = \{c1, c2, c3\}$.
- Rewrite system over the extended signature:
  - **Case 1:** $c3 \succ c1$
    – D-rules: $D11 = \{a \to c1,\ b \to c2,\ f(c1,c2) \to c3\}$
    – C-rules: $C11 = \{c3 \to c1\}$
    – Congruence closure: $R = \{\text{a} \to c1,\ b \to c2,\ f(c1,c2) \to c3,\ c3 \to c1\}$
  - **Case 2:** $c1 \succ c3$
    – D-rules: $D21 = \{a \to c1,\ b \to c2,\ f(c3,c2) \to c3\}$
    – C-rules: $C21 = \{c1 \to c3\}$
    – Congruence closure: $R = \{\text{a} \to c1,\ b \to c2,\ f(c3,c2) \to c3,\ c3 \to c1\}$

- Rewrite system over the initial signature:
  - Apply Compression and Selection.
  - **Case 1:** $c3 \succ c1$
    $f(a, b) \to a$
  - **Case 2:** $c3 \succ c1$
    $f(a, b) \to a$

Combination of the 2 ideas

Ground SOUR graph Completion

$+$

Abstract Congruence Closure

$=$

Completely graph-based Abstract Congruence Closure

(Use of “SER” graphs)

“SER” graphs

– Simpler version of SOUR graphs.
  – No Unification edges (=> DAG).
  – No Orientation edges (efficiency).

– Vertex labeled by a symbol of the original signature (Σ) and a constant (c_i) of $\mathcal{K}$ ($\Sigma \cap \mathcal{K} = \emptyset$).

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

  – Advantage: No Orientation edges.
  – Disadvantage: Convergent rewrite system over $\Sigma \cup \mathcal{K}$.

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

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

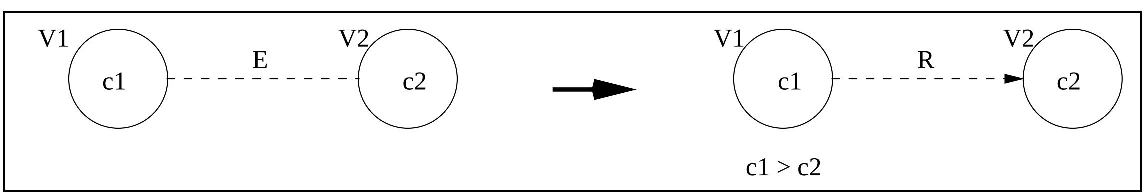
Abstract Congruence Closure

Implemented by graph transformations

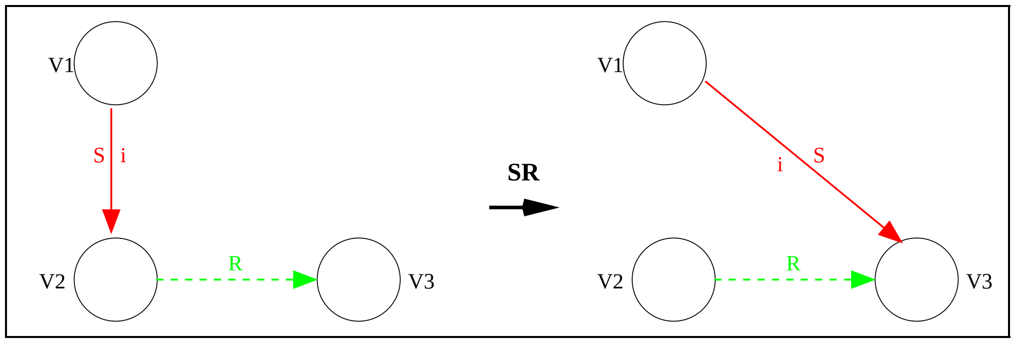
– Simpler version of the SOUR graph rules.

– Four rules:
  – **Orient** – an equality,
  – **SR, RR** – Critical Pairs/Simplification
  – **Merge** – to ensure closure under congruence.

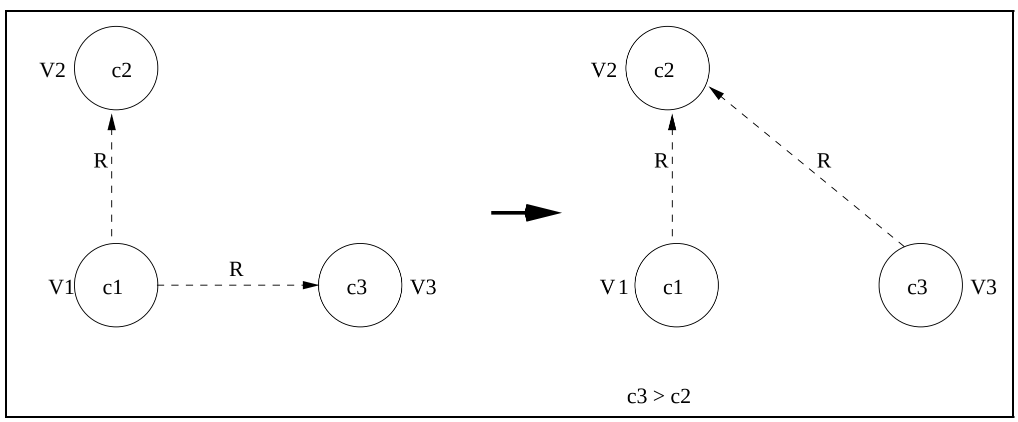
– **Orient rule:**



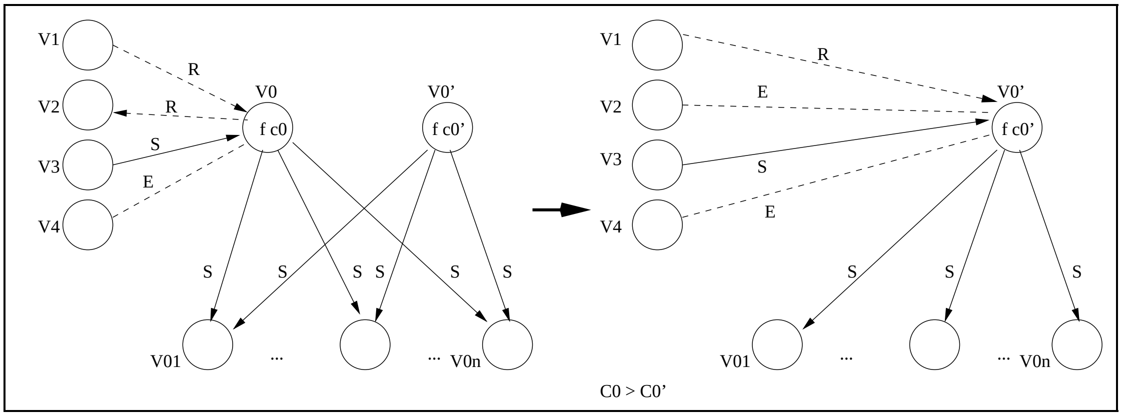
– SR rule:



– RR rule:



– **Merge rule:**



– Strategy of application of the rules:

(Merge* Orient * . (SR RR) *) *

– Rules can be expressed by **Transition rules:** $(E_s,\ E_{eq},\ E_r,\ V,\ KC) \to (E'_s,\ E'_{eq},\ E'_r,\ V',\ KC')$ where:

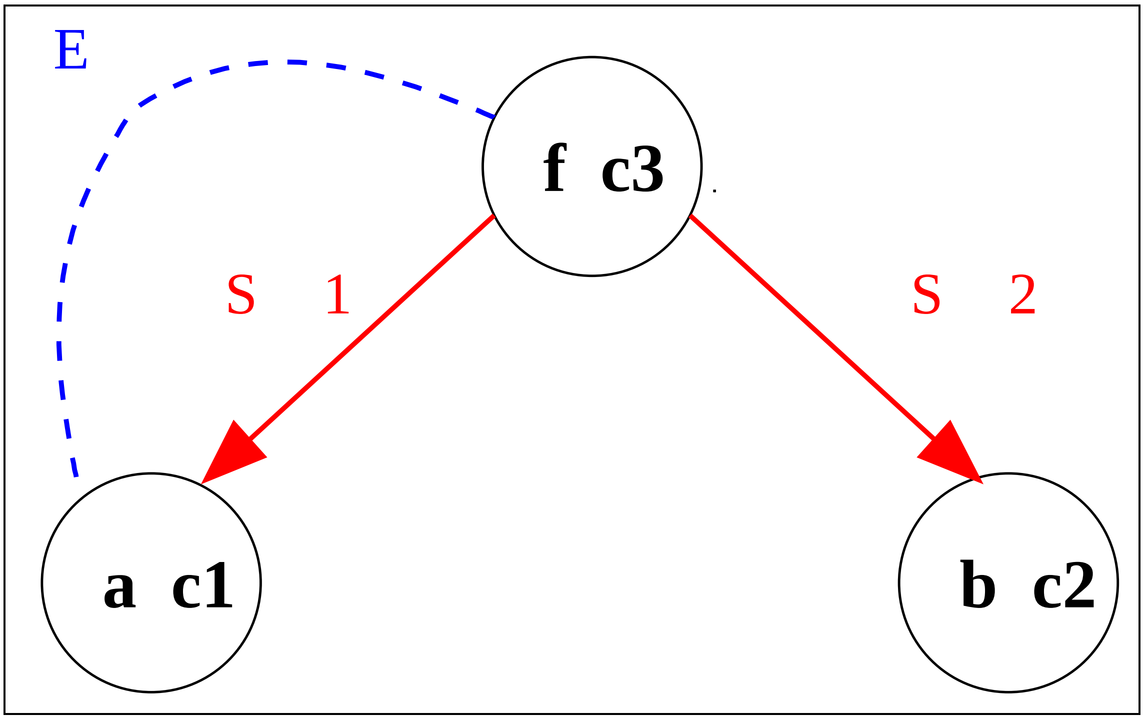
  * E_s, E'_s, E_{eq}, E'_{eq}, E_r and E'_r are set of S, EQ and R edges.

  * V and V' are the sets of vertices of the graphs.

  * KC is the set of initial ordering constraints on constants of $\mathcal{K}$. KC is a set $\{c_i \succ c_j \mid c_i,\ c_j \in \mathcal{K}\}$. The ordering on constants is irreflexive and transitive. KC' is the new set of constants constraints.

Example

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



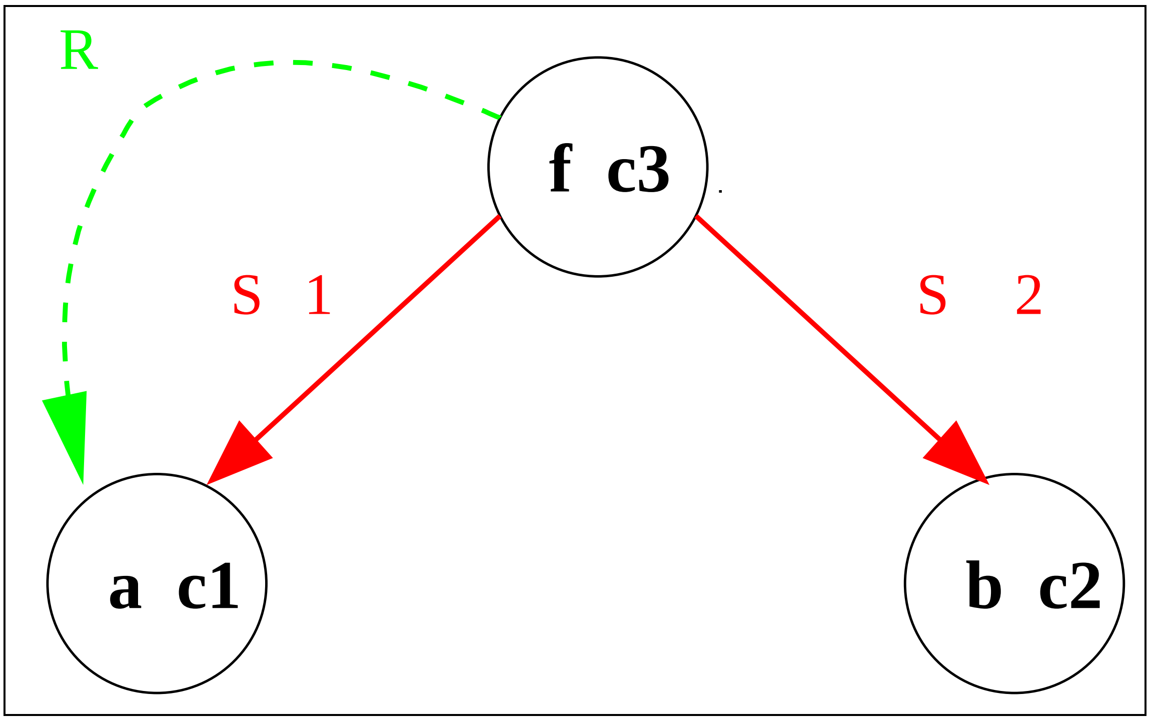
- Reading the initial graph:
  - Over the original signature: E
  - Over the extended signature, $\Sigma \cup \mathcal{K}$, where $\mathcal{K} = \{c1, \ c2, \ c3\}$

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

  C-equalities: $c3 \approx c1$

Example

Application of Orient - Case:
$c3 > c1$



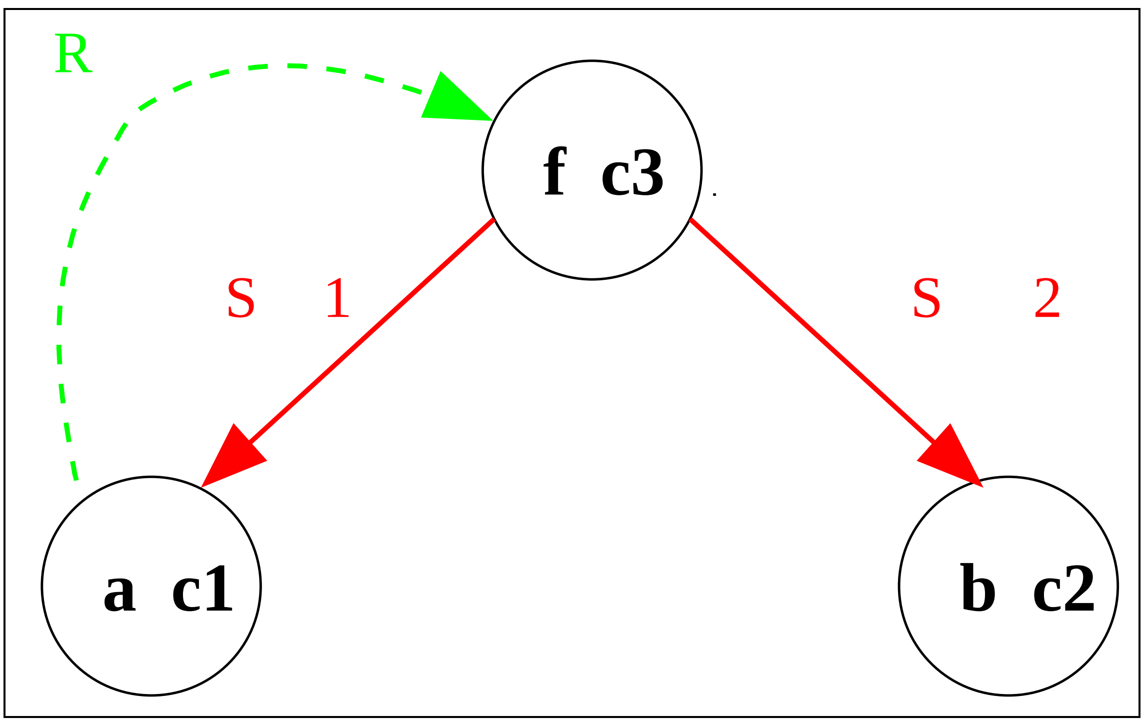
- Convergent rewrite system over the extended signature:

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

- We read teh following rewrite system over Σ: $f(a, b) \to a$

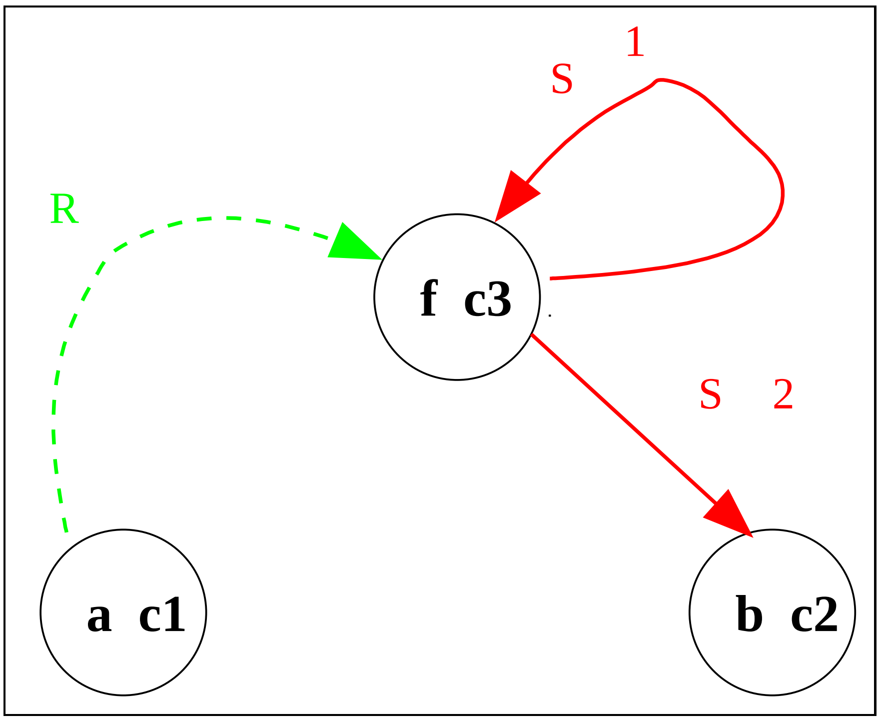
Example

Application of Orient - Case:
$c1 > c3$



Example (cont)

Application of SR - Case: $c1 > c3$



– Convergent rewrite system over the extended signature:

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

– We do not read a rewrite system over the original signature.

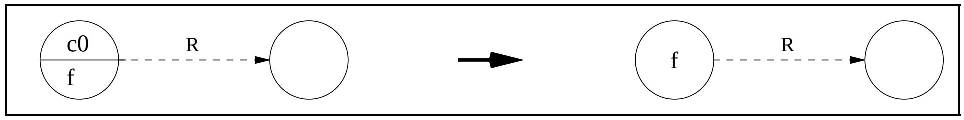
Correctness results

- **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**.

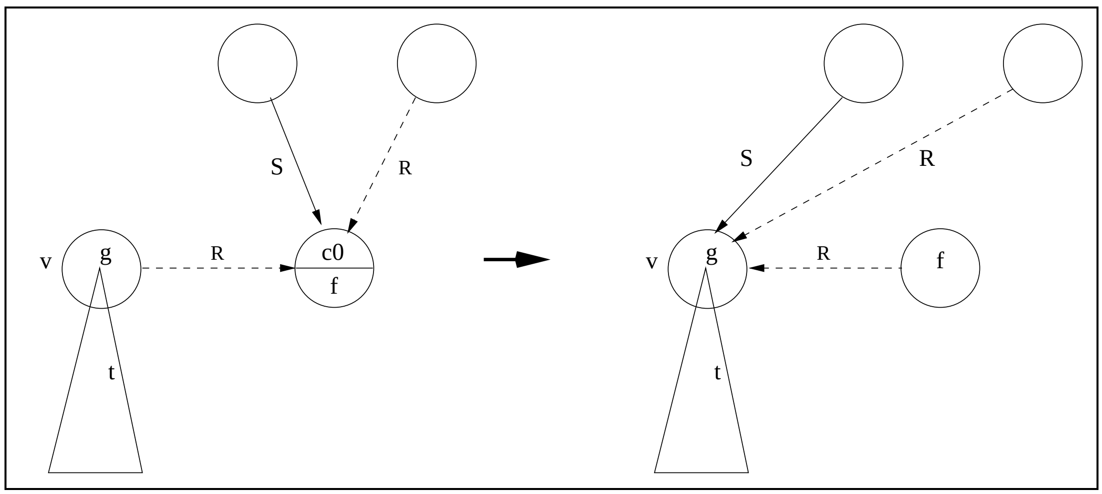
Convergent rewrite system over the original signature

- 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 $\mathcal{K}$
  - Redirect R edges.
  - 3 inference rules

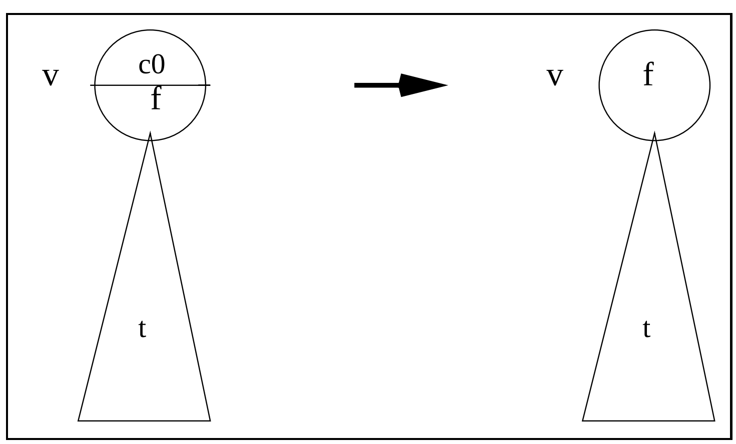
– Compression Transformation



– Selection Transformation 1

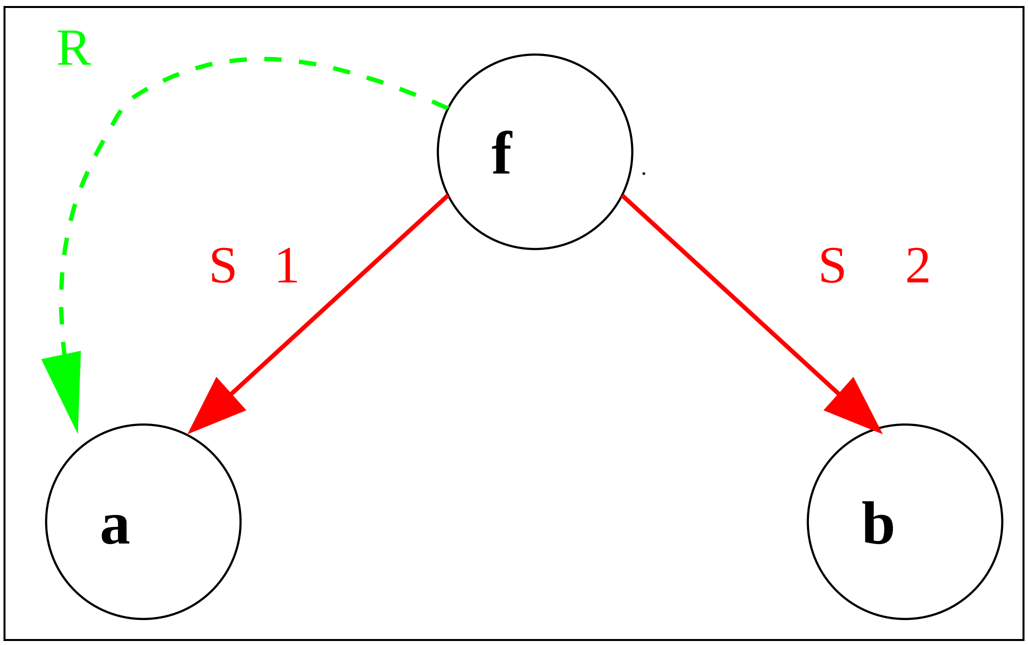


– Selection Transformation 2



Example

Case: $c1 > c3$



- Convergent rewrite system over Σ:
  $f(a, b) \rightarrow a$

Correctness results

- If there is a constant of $\mathcal{K}$ on the graph then we can always apply either the Constant, Compression or Selection graph transformation.
- **Termination:** The application of the Constant, Compression or Selection graph transformation terminates.
- **Soundness:** Exhaustive application of the transformation rules is sound in that the equational theory represented over Σ-terms does not change.
- **Completeness:** We obtain a convergent rewrite system on the initial signature Σ.

Conclusion and perspectives

– Combination of different approaches to solve the word problem.

– Implementation.
  http://www.unitedthinker.com/cc
  (Thanks to Eugene Kipnis)

– 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.