### Term Rewriting Systems (TRSs)

**Applications**
- Equational reasoning
- Program verification
- Deductive programming
- Other fields of computational logic and theoretical computer science

**Properties**
- Termination: There are no infinite reduction chains
- Confluence: Divergences are joinable
- Undecidable in general, but in some cases good criteria and methods exist.

**Methods**
- Proving confluence using Newman’s Lemma
- TRSs are terminating and critical pairs are joinable.

**Goal of work**
- Definition of framework for all transformations
- Comparison of transformations
- Improvement of transformations
- Defining and proving properties of transformations
- Mappings to back translate intermediate results
- Criteria for computational equivalence

**Problems of transformations**
- Infinitely critical pairs cause non-confluence in transformed TRS
- Transformed TRS may give rise to reductions that are not possible in CTRS (=simulation-soundness)
- No reliable simulation possible
- Only weak criteria (Mar96)[Mar97][NSt04]
- Counterexample of (Mar96) for simulation-soundness for all transformations
- Bad use of properties of syntactic properties (left-linearity of CTRS)
- Multiple instances of variables encoded in transformations cause unfaithful behavior
- Some transformations require a strategy

### Conditional Term Rewriting Systems (CTRSs)

**Rule application restricted by conditions**
- Intuitive extension of TRSs
- More compact definitions of functions
- Combination of logic and functional programming

**Consequences**
- More joinable critical pairs
- Infeasible critical pairs
- Many properties lose intuition
- Extra variables
- Termination = computability
- Orthogonality = confluence
- Infeasible critical pairs
- Variable overlaps may not be joinable
- Self overlaps at root positions may be critical

### Transformations of CTRSs to TRSs

**Motivations**
- Adapting methods of TRS to CTRSs
  - e.g. [BK86]: Counterexamples for properties of oriented CTRSs
- Proving properties of CTRS using TRSs
  - Better intuition e.g. syntactic properties of CTRSs
- Simulating CTRSs using TRSs
- Conditional narrowing without conditions

**Concrete Transformations**
1. Encode conditions and (necessary) variables in new auxiliary symbol (\(f\), [BK86][GM88][Mar96][On99]):
   \[ l \rightarrow r \xleftarrow{c} t_1 \ldots t_n \rightarrow t \]  
   \[ l = f(t_1, \ldots, t_n), \text{VAR}(r) \rightarrow r \]

2. Encode conditions in additional arguments (increase arity of existing function symbol) ([fri99][AB03][Res04][SR06]):
   \[ l \rightarrow r \xleftarrow{c} t_1 \ldots t_n \rightarrow t \]
   \[ l = f(t_1, \ldots, t_n), \text{VAR}(r) \rightarrow r \]

### Results, Contributions and Perspectives

**Results**
- Importance of left-linearity of transformed TRS for simulation-soundness
- Importance of encoding all variables of thes of conditional rules

**Contributions**
- Improvements of transformations of [Vir99] for DCTRSs
- Refusal and reformulation of some results of [Mar96][Vir99][SR06]

**Perspectives**
- New criteria for simulation-soundness
- Use of strategies, membership conditional rewriting or context-sensitive rewriting

---

**Literature**


---

**Concrete Transformations**

<table>
<thead>
<tr>
<th>CTRS</th>
<th>[Mar96]</th>
<th>[Vir99]</th>
<th>[SR06]</th>
<th>Our approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f(g(x), y) \rightarrow x = 0)</td>
<td>(f(g(x), y) \rightarrow x )</td>
<td>(f(g(x), y) \rightarrow x )</td>
<td>(f(g(x), y) \rightarrow x )</td>
<td>(f(g(x), y) \rightarrow x )</td>
</tr>
<tr>
<td>(g(y(x), y) \rightarrow y )</td>
<td>(g(y(x), y) \rightarrow y )</td>
<td>(g(y(x), y) \rightarrow y )</td>
<td>(g(y(x), y) \rightarrow y )</td>
<td>(g(y(x), y) \rightarrow y )</td>
</tr>
</tbody>
</table>

---

**Comparison of confluence of overlapping CTRSs**

**Consequences**
- Infinitely critical pairs cause non-confluence in transformed TRS
- Transformed TRS may give rise to reductions that are not possible in CTRS (=simulation-soundness)
- No reliable simulation possible
- Only weak criteria (Mar96)[Mar97][NSt04]
- Counterexample of (Mar96) for simulation-soundness for all transformations
- Bad use of properties of syntactic properties (left-linearity of CTRS)
- Multiple instances of variables encoded in transformations cause unfaithful behavior
- Some transformations require a strategy