Generalized Rewrite Theories

Roberto Bruni

bruni@di.unipi.it

University of Pisa

University of Illinois at Urbana-Champaign

(joint work with **José Meseguer**)

Plan of the talk

- Rewriting Logic (RL) in the 90's
- Feedback from practice
- Generalized Rewrite Theories
- RL today

Plan of the talk

- Rewriting Logic (RL) in the 90's
  - the many faces of RL (semantic and logical framework, programming paradigm)
  - theoretical foundations (inference rules, models, completeness results)
- Feedback from practice
- Generalized Rewrite Theories
- RL today

Plan of the talk

- Rewriting Logic (RL) in the 90's
  - the many faces of RL (semantic and logical framework, programming paradigm)
  - theoretical foundations (inference rules, models, completeness results)
- Feedback from practice
  - which equational logic?
  - which conditions for rules?
  - should rewrites be applicable anywhere?
- Generalized Rewrite Theories
- RL today

Plan of the talk

- Rewriting Logic (RL) in the 90's
  - the many faces of RL (semantic and logical framework, programming paradigm)
  - theoretical foundations (inference rules, models, completeness results)
- Feedback from practice
  - which equational logic? membership equational logic
  - which conditions for rules?
  - should rewrites be applicable anywhere?
- Generalized Rewrite Theories
- RL today

Plan of the talk

- Rewriting Logic (RL) in the 90's
  - the many faces of RL (semantic and logical framework, programming paradigm)
  - theoretical foundations (inference rules, models, completeness results)
- Feedback from practice
  - which equational logic? membership equational logic
  - which conditions for rules? equations, memberships, rewrites
  - should rewrites be applicable anywhere?
- Generalized Rewrite Theories
- RL today

Plan of the talk

- Rewriting Logic (RL) in the 90's
  - the many faces of RL (semantic and logical framework, programming paradigm)
  - theoretical foundations (inference rules, models, completeness results)
- Feedback from practice
  - which equational logic? membership equational logic
  - which conditions for rules? equations, memberships, rewrites
  - should rewrites be applicable anywhere? frozen arguments
- Generalized Rewrite Theories
- RL today

Plan of the talk

- Rewriting Logic (RL) in the 90's
  - the many faces of RL (semantic and logical framework, programming paradigm)
  - theoretical foundations (inference rules, models, completeness results)
- Feedback from practice
  - which equational logic? membership equational logic
  - which conditions for rules? equations, memberships, rewrites
  - should rewrites be applicable anywhere? frozen arguments
- Generalized Rewrite Theories
  - revisiting the foundations
- RL today

Plan of the talk

- Rewriting Logic (RL) in the 90's
  - the many faces of RL (semantic and logical framework, programming paradigm)
  - theoretical foundations (inference rules, models, completeness results)
- Feedback from practice
  - which equational logic? membership equational logic
  - which conditions for rules? equations, memberships, rewrites
  - should rewrites be applicable anywhere? frozen arguments
- Generalized Rewrite Theories
  - revisiting the foundations
- RL today
  - related work
  - future directions

Birth

Origins

- Term rewriting as a technique for equational deduction
  - simplification to canonical forms [Knuth Bendix 1970]
- Rewriting as a unified model of concurrency
  - a logic about change and evolution [Meseguer CONCUR'90]
  - conceptual unification and semantic integration of a wide variety of (concurrent) programming paradigms
    - Turing machines
    - phrase structure grammars
    - labeled transition systems
    - Petri nets
    - process algebras
    - linear logic
    - chemical abstract machine
    - actors
    - ...

Main ingredients and features

- Rewrite rules modulo structural axioms
- Simple syntax and clean semantics
- Distribution and concurrency from underlying algebraic structure
- Sound and complete logic with initial and loose semantics
- Basis of declarative programming languages (with functional sublanguages, system modules, and object-oriented modules)
- Reflection

The many faces of RL

- Semantic framework
  - models of computation, programming languages, hardware, Sw architectures, ρ-calculus,...
- Logical framework
  - equational logic, Horn logic, linear logic, sequent systems constraint logic, linear temporal logic, tile logic,...
- Programming paradigm
  - ELAN, CafeOBJ, **Maude 2.0**
- Applications and extensions
  - formal methods, communication protocols, bioinformatics, constraint solving, mobility, real-time, probabilistic, behavioral
- Stimulating scientific community
  - WRLA 1996, 1998, 2000, 2002
  - special issue of TCS (vol. 285)

The many faces of RL

- Semantic framework
  - models of computation, programming languages, hardware, Sw architectures, ρ-calculus,...
- Logical framework
  - equational logic, Horn logic, linear logic, sequent systems constraint logic, linear temporal logic, tile logic,...
- Programming paradigm
  - ELAN, CafeOBJ, **Maude 2.0**
- Applications and extensions
  - formal methods, communication protocols, bioinformatics, constraint solving, mobility, real-time, probabilistic, behavioral
- Stimulating scientific community
  - WRLA 1996, 1998, 2000, 2002 , **2004 Upcoming event!**
  - special issue of TCS (vol. 285)

Rewrite theories in the 90's

Definition A *(labeled) conditional rewrite theory* is a triple $\mathcal{R} = (\Sigma, E, R)$, where

- (Σ, E) is an unsorted equational theory
- R is a set of (labeled) conditional rewrite rules having the form below, with $t, t', t_i, t_i' \in \mathbb{T}_\Sigma(X)$

$$(\forall \mathbf{X})\ \mathbf{r}\colon \mathbf{t} \to \mathbf{t}' \quad \textit{if} \quad \mathbf{t_1} \to \mathbf{t_1'} \wedge \ldots \wedge \mathbf{t}_\ell \to \mathbf{t}_\ell'$$

Rules with empty conditions $(\forall X)\ r\colon t \to t'$ are called *unconditional*.

Example: Petri nets

Let $\mathbf{N} = (\mathbf{S}, \mathbf{T}, pre, post)$ be a Petri net. The associated rewrite theory $\mathcal{R}_{\mathbf{N}}$ is defined as:

- Signature Σ:

- Equations E:

- Rewrite rules R:

Example: Petri nets

Let $\mathbf{N} = (\mathbf{S}, \mathbf{T}, pre, post)$ be a Petri net. The associated rewrite theory $\mathcal{R}_{\mathbf{N}}$ is defined as:

- Signature Σ:
  - places $a_1, ..., a_n \in S$ become constants
  - a binary operator $\otimes$ models multiset union
  - a special constant $\varnothing$ denote the empty marking
- Equations E:
- Rewrite rules R:

Example: Petri nets

Let $\mathbf{N} = (\mathbf{S}, \mathbf{T}, pre, post)$ be a Petri net. The associated rewrite theory $\mathcal{R}_{\mathbf{N}}$ is defined as:

- Signature Σ:
  - places $a_1, ..., a_n \in S$ become constants
  - a binary operator $\otimes$ models multiset union
  - a special constant $\varnothing$ denote the empty marking
- Equations E:
  - $\otimes$ is associative, commutative and has identity $\varnothing$
- Rewrite rules R:

Example: Petri nets

Let $\mathbf{N} = (\mathbf{S}, \mathbf{T}, pre, post)$ be a Petri net. The associated rewrite theory $\mathcal{R}_{\mathbf{N}}$ is defined as:

- Signature Σ:
  - places $a_1, ..., a_n \in S$ become constants
  - a binary operator $\otimes$ models multiset union
  - a special constant $\varnothing$ denote the empty marking
- Equations E:
  - $\otimes$ is associative, commutative and has identity $\varnothing$
- Rewrite rules R:
  - each transition $t \in T$ becomes an unconditional rewrite rule $(\forall \varnothing)\ t : pre(t) \rightarrow post(t)$

RL as a sequent calculus

$$\frac{t \in \mathbb{T}_\Sigma(X)}{(\forall X)\ t \to t}\ \textbf{Reflexivity} \qquad \frac{(\forall X)\ t_1 \to t_2 \quad (\forall X)\ t_2 \to t_3}{(\forall X)\ t_1 \to t_3}\ \textbf{Transitivity}$$

$$\frac{E \vdash (\forall X)\ t = u \quad (\forall X)\ u \to u' \quad E \vdash (\forall X)\ u' = t'}{(\forall X)\ t \to t'}\ \textbf{Equality}$$

$$\frac{f \in \Sigma_n \quad (\forall X)\ t_i \to t_i' \text{ for } i \in [1,n]}{(\forall X)\ f(t_1,\ldots,t_n) \to f(t_1',\ldots,t_n')}\ \textbf{Congruence}$$

$$\frac{\begin{array}{c}(\forall X)\ r\colon t \to t' \text{ if } \bigwedge_{i\in[1,\ell]} t_i \to t_i' \in R \quad \theta,\theta'\colon X \to \mathbb{T}_\Sigma(Y) \\ (\forall Y)\ \theta(t_i) \to \theta(t_i') \text{ for } i \in [1,\ell] \quad (\forall Y)\ \theta(x) \to \theta'(x) \text{ for } x \in X\end{array}}{(\forall Y)\ \theta(t) \to \theta'(t')}\ \textbf{Nested Replacement}$$

Entailment and models

- Entailment
  - we write $\mathcal{R} \vdash (\forall \mathbf{X})\ \mathbf{t} \rightarrow \mathbf{t}'$ if the sequent $(\forall X)\ t \rightarrow t'$ can be derived from the inference rules

Entailment and models

- Entailment
  - we write $\mathcal{R} \vdash (\forall \mathbf{X})\ \mathbf{t} \to \mathbf{t}'$ if the sequent $(\forall X)\ t \to t'$ can be derived from the inference rules
- Model-theoretic view
  - sequents can be decorated with **proof terms**
  - proof terms can be suitably axiomatized (category, functoriality, decomposition and exchange laws)
  - the resulting proof algebra is initial in the class of **$\mathcal{R}$-systems** (i.e. the models of $\mathcal{R}$)

Entailment and models

- Entailment
  - we write $\mathcal{R} \vdash (\forall \mathbf{X})\ \mathbf{t} \to \mathbf{t}'$ if the sequent $(\forall X)\ t \to t'$ can be derived from the inference rules
- Model-theoretic view
  - sequents can be decorated with **proof terms**
  - proof terms can be suitably axiomatized (category, functoriality, decomposition and exchange laws)
  - the resulting proof algebra is initial in the class of **$\mathcal{R}$-systems** (i.e. the models of $\mathcal{R}$)
- **Completeness theorem**
  - a sequent is provable from $\mathcal{R}$
  - **iff** it is satisfied in all models of $\mathcal{R}$
  - **iff** it is satisfied in the (initial/free) proof term algebra of $\mathcal{R}$

Curry-Howard-like perspective

- **Computational reading of $\mathbf{t \to t'}$**
  - the state t can evolve to the state t'

Curry-Howard-like perspective

- **Computational reading of $\mathbf{t \to t'}$**
  - the state t can evolve to the state t'
- **Logical meaning of $\mathbf{t \to t'}$**
  - the formula t' can be deduced from the formula t

Curry-Howard-like perspective

- **Computational reading of $\mathbf{t} \to \mathbf{t}'$**
  - the state t can evolve to the state t'
- **Logical meaning of $\mathbf{t} \to \mathbf{t}'$**
  - the formula t' can be deduced from the formula t
- Curry-Howard-like view
  - (Σ, E)-terms — **states** — **logical propositions**
  - rewrites — **computations** — **proofs**
  - algebraic structure — **distribution** — **propositional structure**

Curry-Howard-like perspective

- **Computational reading of $\mathbf{t \to t'}$**
  - the state t can evolve to the state t'
- **Logical meaning of $\mathbf{t \to t'}$**
  - the formula t' can be deduced from the formula t
- Curry-Howard-like view
  - (Σ, E)-terms — **states** — **logical propositions**
  - rewrites — **computations** — **proofs**
  - algebraic structure — **distribution** — **propositional structure**
- Example
  - Petri nets and linear logic [Martí-Oliet Meseguer 1991]

Attractions for a *teenager*

Learning from practice

- Along the years, RL has been claimed to ***parametric w.r.t. the underlying equational theory***
  - unsorted, many sorted, order-sorted, membership equational logic
  - implemented and very useful in practice
  - is the claim really true?

Learning from practice

- Along the years, RL has been claimed to ***parametric w.r.t. the underlying equational theory***
  - unsorted, many sorted, order-sorted, membership equational logic
  - implemented and very useful in practice
  - is the claim really true?
- Rewrites ***should not happen everywhere***
  - evaluation strategies and context-sensitive rewriting
  - can improve performance and avoid non-termination
  - frozen kinds / operators / arguments
  - how can this fit in the theoretical foundations of RL?

Learning from practice

- Along the years, RL has been claimed to ***parametric w.r.t. the underlying equational theory***
  - unsorted, many sorted, order-sorted, membership equational logic
  - implemented and very useful in practice
  - is the claim really true?
- Rewrites ***should not happen everywhere***
  - evaluation strategies and context-sensitive rewriting
  - can improve performance and avoid non-termination
  - frozen kinds / operators / arguments
  - how can this fit in the theoretical foundations of RL?
- Rewrites can be conditional to ***equality and membership***
  - implemented even before than rewrites as conditions
  - what is the impact on RL semantics?

Learning from practice

- ***All this has been already implemented in Maude!***
- We must expand and consolidate the semantic foundations of rewriting logic along these three directions:
  - a castle with too many dangling rooms can easily fall

Examples: Memberships

- **Making the Petri net example more precise**
  - unsorted equational theories: as seen

Examples: Memberships

- **Making the Petri net example more precise**
  - unsorted equational theories: as seen
  - many sorted equational theories:

```
sorts Place Marking.
ops a1 a2 ... an : -> Place.
op _⊗_ : Marking Marking -> Marking [assoc comm id: ∅].
op ∅ : -> Marking.
op {_} : Place -> Marking.
```

Examples: Memberships

- **Making the Petri net example more precise**
  - unsorted equational theories: as seen
  - many sorted equational theories:
  - **order-sorted equational theories:**

```
sorts Place Marking. subsort Place < Marking.
ops a1 a2 ... an : -> Place.
op _⊗_ : Marking Marking -> Marking [assoc comm id: ∅].
op ∅ : -> Marking.
```

Examples: Memberships

- **Making the Petri net example more precise**
  - unsorted equational theories: as seen
  - many sorted equational theories:
  - **order-sorted equational theories:**
  - **membership equational theories:**

```
sorts Place Marking. subsort Place < Marking.
ops a1 a2 ... an : -> Place.
op _⊗_ : Marking Marking -> Marking [assoc comm id: ∅].
op ∅ : -> Safe.
sort Safe.
subsorts Place < Safe < Marking.
cmb P ⊗ S : Safe
    if P:Place ∧ S:Safe ∧ (P in S = false).
```

Examples: Frozen kinds

- Does it make sense to rewrite under any operator?

Examples: Frozen kinds

- Does it make sense to rewrite under any operator?
  - the Context lemma would say so
  - but let us consider the cardinality example:

```
op |_| : Marking -> Nat.
eq |∅| = 0.
ceq |P⊗M| = 1+|M|    if P:Place ∧ M:Marking.
```

Examples: Frozen kinds

- Does it make sense to rewrite under any operator?
  - the Context lemma would say so
  - but let us consider the cardinality example:
  - The multiset whose cardinality we wish to determine becomes a moving target! (transitions would induce rewrites at the `Nat` level)

```
op |_| : Marking -> Nat.
eq |∅| = 0.
ceq |P⊗M| = 1+|M|    if P:Place ∧ M:Marking.

rl [t1] : a⊗a -> b.
rl [t2] : b⊗c -> d.
```

Examples: Frozen kinds

- Does it make sense to rewrite under any operator?
  - the Context lemma would say so
  - but let us consider the cardinality example:
  - The multiset whose cardinality we wish to determine becomes a moving target! (transitions would induce rewrites at the `Nat` level)

```
op |_| : Marking -> Nat.
eq |∅| = 0.
ceq |P⊗M| = 1+|M|    if P:Place ∧ M:Marking.

rl [t1] : a⊗a -> b.
rl [t2] : b⊗c -> d.

|a⊗a⊗c| -> |b⊗c| -> |d|
3 -> 2 -> 1
```

Examples: Frozen kinds

- Does it make sense to rewrite under any operator?
  - the Context lemma would say so
  - but let us consider the cardinality example:
  - The multiset whose cardinality we wish to determine becomes a moving target! (transitions would induce rewrites at the `Nat` level)
  - Data are *frozen*, states evolve [Meseguer, Ölveczky, Stehr 2001]

```
op |_| : Marking -> Nat.
eq |∅| = 0.
ceq |P⊗M| = 1+|M|    if P:Place ∧ M:Marking.

rl [t1] : a⊗a -> b.
rl [t2] : b⊗c -> d.

|a⊗a⊗c| -> |b⊗c| -> |d|
3 -> 2 -> 1
```

Examples: Frozen positions

- Frozen kinds could be too coarse

Examples: Frozen positions

- Frozen kinds could be too coarse
  - frozen operators [Meseguer Stefani]

With $_ + _$ frozen:

$$\frac{P \to R}{P + Q \to R}\ \textbf{left choice} \qquad \frac{Q \to R}{P + Q \to R}\ \textbf{right choice}$$

Examples: Frozen positions

- Frozen kinds could be too coarse
  - frozen operators [Meseguer Stefani]
  - frozen arguments

With $_ + _$ frozen:

$$\frac{P \to R}{P + Q \to R}\ \textbf{left choice} \qquad \frac{Q \to R}{P + Q \to R}\ \textbf{right choice}$$

With the second argument of $_;_$ frozen:

$$\frac{P \to R}{0; P \to R}\ \textbf{comp}$$

Examples: Frozen positions

- Frozen kinds could be too coarse
  - frozen operators [Meseguer Stefani]
  - frozen arguments
  - in essence, this brings RL closer to SOS

With $_ + _$ frozen:

$$\frac{P \to R}{P + Q \to R}\ \textbf{left choice} \qquad \frac{Q \to R}{P + Q \to R}\ \textbf{right choice}$$

With the second argument of $_;_$ frozen:

$$\frac{P \to R}{0; P \to R}\ \textbf{comp}$$

Examples: Conditional rules

- Should any substitution be applicable to rewrites?

Examples: Conditional rules

- Should any substitution be applicable to rewrites?
  - the Substitution lemma would say so

Examples: Conditional rules

- Should any substitution be applicable to rewrites?
  - the Substitution lemma would say so
  - but side-conditions ar often useful

$$\frac{P \xrightarrow{a(n)} R}{P|Q \xrightarrow{a(n)} R|Q} \; n \notin freeNames(Q)$$

Growing Up

Generalized rewrite theories

Definition A *generalized rewrite theory* is a tuple $\mathcal{R} = (\Sigma, E, \phi, R)$, where

- (Σ, E) is a *membership* equational theory
- $\phi\colon \Sigma \to \wp_{\mathrm{f}}(\mathbb{N})$ is a functions assigning to each operator $f\colon k_1 \cdots k_n \to k$ the set of its *frozen argument positions* $\phi(f) \subseteq \{1, \ldots, n\}$ and let $\nu(f) = \{1, \ldots, n\} \smallsetminus \phi(f)$
- R is a set of conditional rewrite rules having the *general form* below:

$$(\forall \mathbf{X})\ \mathbf{r}\colon \mathbf{t} \to \mathbf{t}' \text{ if } \bigwedge_{\mathbf{i} \in \mathbf{I}} \mathbf{p_i} = \mathbf{q_i} \wedge \bigwedge_{\mathbf{j} \in \mathbf{J}} \mathbf{w_j} : \mathbf{s_j} \wedge \bigwedge_{\mathbf{l} \in \mathbf{L}} \mathbf{t_l} \to \mathbf{t'_l}$$

GRT: Inference rules

$$\frac{t \in \mathbb{T}_\Sigma(X)_k}{(\forall X)\ t \to t}$$ **Reflexivity**

$$\frac{(\forall X)\ t_1 \to t_2 \quad (\forall X)\ t_2 \to t_3}{(\forall X)\ t_1 \to t_3}$$ **Transitivity**

$$\frac{E \vdash (\forall X)\ t = u \quad (\forall X)\ u \to u' \quad E \vdash (\forall X)\ u' = t'}{(\forall X)\ t \to t'}$$ **Equality**

$$\frac{f \in \Sigma_n \quad (\forall X)\ t_i \to t_i' \text{ for } i \in \nu(f) \quad t_i = t_i' \text{ for } i \in \phi(f)}{(\forall X)\ f(t_1,\ldots,t_n) \to f(t_1',\ldots,t_n')}$$ **Congruence**

$$\frac{\begin{array}{c}(\forall X)\ r\colon t \to t' \text{ if } \bigwedge_{i\in I} p_i = q_i \wedge \bigwedge_{j\in J} w_j : s_j \wedge \bigwedge_{l\in L} t_l \to t_l' \in R \\ \theta,\theta'\colon X \to \mathbb{T}_\Sigma(Y) \quad \theta(x) = \theta'(x) \text{ for } x \in \phi(t,t') \\ E \vdash (\forall Y)\ \theta(p_i) = \theta(q_i) \text{ for } i \in I \quad E \vdash (\forall Y)\ \theta(w_j) : s_j \text{ for } j \in J \\ (\forall Y)\ \theta(t_l) \to \theta(t_l') \text{ for } l \in L \quad (\forall Y)\ \theta(x) \to \theta'(x) \text{ for } x \in \nu(t,t')\end{array}}{(\forall Y)\ \theta(t) \to \theta'(t')}$$ **Nested Replacement**

Main results: Deduction

- Any unsorted rewrite theory $\mathcal{R}$ can be regarded as a GRT $\widehat{\mathcal{R}}$ where:
  - Σ has a unique kind and no sorts
  - all the operators are total and unfrozen
  - conditions in rewrite rules contain neither equalities nor membership predicates

Theorem Deduction via rules for conditional rewrite theories $\mathcal{R}$ coincides with deduction via rules for GRTs $\widehat{\mathcal{R}}$:

- $\mathcal{R} \vdash (\forall \mathbf{X})\ \mathbf{t} \to \mathbf{t}' \quad \Leftrightarrow \quad \widehat{\mathcal{R}} \vdash (\forall \mathbf{X})\ \mathbf{t} \to \mathbf{t}'$

GRT: Rechability models

- Reachability models focus on *what* states can be reached from t, ignoring *how* the rewrites are performed.
  - $[\mathbf{t}] \rightarrow_{\mathcal{R}} [\mathbf{t'}] \quad \Leftrightarrow \quad \mathcal{R} \vdash (\forall X)\, t \longrightarrow t'$

GRT: Rechability models

- Reachability models focus on *what* states can be reached from t, ignoring *how* the rewrites are performed.
  - $[\mathbf{t}] \rightarrow_{\mathcal{R}} [\mathbf{t'}] \quad \Leftrightarrow \quad \mathcal{R} \vdash (\forall X)\, t \longrightarrow t'$
- Model-theoretic presentation of the reachability relation
  - free model of a suitable MEL theory $Reach(\mathcal{R})$
  - sorts:
    - $Pair_{\mathbf{k}}$ (all possible source/target pairs)
    - $Arrow_{\mathbf{k}}$ (reachability relation)
    - $Arrow^{\mathbf{1}}_{\mathbf{k}}$ (one-step reachability relation)
    - $Arrow^{\mathbf{0}}_{\mathbf{k}}$ (idle rewrites)
  - subsorts: $Arrow^0_k\ Arrow^1_k < Arrow_k < Pair_k$
  - operators:
    - $(_ \rightarrow _) : k\ k \longrightarrow Pair_k$ (pair constructor)
    - $\mathtt{s}, \mathtt{t} : Pair_k \longrightarrow k$ (source/target projections)
    - $(_\,;_) : [Pair_k]\ [Pair_k] \longrightarrow [Pair_k]$ (concatenation)

Main results: Reachability

- The theory $Reach(\mathcal{R})$ provides an algebraic model for the reachability relation
  - for ground terms, such a model is given by the interpretation of the sorts $Arrow_k$ in the initial model $\mathbb{T}_{Reach(\mathcal{R})}$
  - for terms with variables in X, we consider the free algebra $\mathbb{T}_{Reach(\mathcal{R})}(\mathbf{X})$

Theorem We have the equivalences:

$$\begin{aligned}
\mathcal{R} \vdash (\forall X)\, t \to t' \quad & \Leftrightarrow \quad Reach(\mathcal{R}) \vdash (\forall \mathbf{X})\ (\mathbf{t} \to \mathbf{t'}) : Arrow_{\mathbf{k}} \\
& \Leftrightarrow \quad Reach(\mathcal{R}) \models (\forall \mathbf{X})\ (\mathbf{t} \to \mathbf{t'}) : Arrow_{\mathbf{k}} \\
& \Leftrightarrow \quad [(\mathbf{t} \to \mathbf{t'})] \in \mathbb{T}_{Reach(\mathcal{R})}(\mathbf{X})_{Arrow_{\mathbf{k}}}
\end{aligned}$$

GRT: Concurrent models

- In general, many proofs concluding $\mathcal{R} \vdash (\forall X)\, t \longrightarrow t'$ are possible and not all of them are computationally equivalent
  - they may differ e.g. in applied rules, causal dependencies

GRT: Concurrent models

- In general, many proofs concluding $\mathcal{R} \vdash (\forall X)\, t \longrightarrow t'$ are possible and not all of them are computationally equivalent
  - they may differ e.g. in applied rules, causal dependencies
- Model-theoretic presentation of the true concurrent models for $\mathcal{R}$ based on decorated sequents
  - free model of a suitable MEL theory $Proof(\mathcal{R})$
  - sorts:
    - $Rew_{\mathbf{k}}$ (all admissible rewrites)
    - $Rew^{\mathbf{1}}_{\mathbf{k}}$ (one-step rewrites)
    - $\mathbf{k}$ (idle rewrites)
  - subsorts: $k\ Rew^1_k < Rew_k$
  - operators: Σ (lifted when needed), source and target, sequential composition, labels of rewrite rules

GRT: Concurrent models

- In general, many proofs concluding $\mathcal{R} \vdash (\forall X)\, t \longrightarrow t'$ are possible and not all of them are computationally equivalent
  - they may differ e.g. in applied rules, causal dependencies
- Model-theoretic presentation of the true concurrent models for $\mathcal{R}$ based on decorated sequents
  - free model of a suitable MEL theory $Proof(\mathcal{R})$
  - sorts:
    - $Rew_{\mathbf{k}}$ (all admissible rewrites)
    - $Rew^{\mathbf{1}}_{\mathbf{k}}$ (one-step rewrites)
    - $\mathbf{k}$ (idle rewrites)
  - subsorts: $k\ Rew^1_k < Rew_k$
  - operators: Σ (lifted when needed), source and target, sequential composition, labels of rewrite rules
- Petri net example: $|\mathtt{t1} \otimes \mathtt{c}|$ has no sort

Main results: Decorated proofs

- The theory $Proof(\mathcal{R})$ is a conservative extension of the underlying theory (Σ, E)
- The theory $Proof(\mathcal{R})$ is complete w.r.t. inference in GRT

Theorem We have the equivalences:

$$\begin{aligned}
\mathcal{R} \vdash (\forall X)\, t \to t' \quad &\Leftrightarrow \quad \exists \alpha.\, Proof(\mathcal{R}) \vdash (\forall \mathbf{X})\, \alpha : Rew_{\mathbf{k}} \wedge \mathsf{s}(\alpha) = \mathbf{t} \wedge \mathsf{t}(\alpha) = \mathbf{t}' \\
&\Leftrightarrow \quad \exists \alpha.\, Proof(\mathcal{R}) \models (\forall \mathbf{X})\, \alpha : Rew_{\mathbf{k}} \wedge \mathsf{s}(\alpha) = \mathbf{t} \wedge \mathsf{t}(\alpha) = \mathbf{t}' \\
&\Leftrightarrow \quad \exists \alpha.\, \mathbb{T}_{Proof(\mathcal{R})}(\mathbf{X}) \models (\forall \mathbf{X})\, \alpha : Rew_{\mathbf{k}} \wedge \mathsf{s}(\alpha) = \mathbf{t} \wedge \mathsf{t}(\alpha) = \mathbf{t}'
\end{aligned}$$

Shaking up the tree

Conclusion

- Main results and advantageous features
  - deduction rules for GRTs, GRT models as MEL algebras, existence of initial and free (reachability/concurrent) models, completeness w.r.t. model theory
  - full generalization of the original results for unsorted rewrite theories
  - GRTs substantially extend the expressiveness of RL in many applications
  - theory meets what is demanded by practice
  - appropriate mathematical semantics
  - reuse of results and techniques of MEqtl
- Is MEqtl expressive enough to encompass GRT?
  - yes, but separation of concerns (equational vs operational) makes life easier in most applications

Related work, future directions

- Flourishing literature towards the full integration of type theory with RL
  - (typed) rho-calculus [Cirstea Kirchner Liquori]
  - pure type systems [Meseguer Stehr]
  - higher-order vs reflection [Meseguer et al.]
- Future work
  - make explicit the 2-categorical nature of our model theory
  - semantics of parametric GRTs and views

Acknowledgments

Research supported by

- MIUR Project COFIN 2001013518 CoMeta
- FET-GC Project IST-2001-32747 Agile
- ONR Grant N00014-02-1-0715
- Italian CNR fellowship for research on Information Sciences and Technologies