Non-ω-overlappings TRSs are UN

Stefan Kahrs and Connor Smith

University of Kent

University of Kent

This is about:

- When is the equational theory of a TRS consistent (CON), when does it have unique normal forms (UN),…
- How can we prove it?
- How are these issues related to ω-substitutions, substitutions with infinitary terms in their range …

Ultimately, we got:

- The result mentioned in the title.
- A new proof technique for proving consistency.

In this talk…

- …I will gloss over the straightforward parts of the proof (and skip quickly over slides with that content)
- …and focus mostly on the trickier areas

The issue:

- We are generally interested in the consistency of equational theories, i.e. that not everything is equal to everything (on open terms)
- This is normally proved via confluence (CR): A confluent system is consistent, because:
  - Variables are normal forms
  - Distinct variables are distinct normal forms
- Other kinds of consistency proofs?
  - One can create a non-trivial equational model, but…
  - …that is hard to “bootstrap” in this case (no CPO structure).

Standard TRS confluence criteria

- For terminating systems:
  - weak confluence, all critical pairs between rewrite rules have common reducts
- For non-terminating systems:
  - There are no overlaps (rules that give rise to critical pairs) in the first place, and…
  - The system is left-linear.

Not just for Confluence, UN & CON too:

$F(x,x) \to A$
$F(C(x),x) \to B$
$E \to C(E)$

F(E,E) has distinct normal forms A and B. So this is not UN, despite having no overlaps. Moreover, add the rules:

$G(A,x,y) \to x$
$G(B,x,y) \to y$

Now, $x=G(A,x,y)=G(F(E,E),x,y)=G(F(C(E),E),x,y)=G(B,x,y)=y$. The modified system still has no overlaps but is not CON.

What is going on?

- The system actually did have overlaps, but the standard definition of overlap does not acknowledge them:
  - If we allow for substitutions to replace variables with infinitary terms then the first two rules overlap, i.e. can be applied to the same term
  - And we had a finite term that was "semantical equal" to such an infinite term
- So if we move from LL & non-overlapping to non-ω-overlapping then this counter-example goes away, but is that it?
  - We will not regain confluence, but UN and/or CON?
  - Open problem #79 since 1989.

Reducing the problem (i)

It suffices to look at CON:

- (On open terms) UN implies CON
- Suppose we had a non-ω-overlapping TRS that was not UN. Then
  - There are distinct but equivalent normal forms t and u.
  - We can get non-ω-unifiable but equivalent ground normal forms t' and u' from t and u, possibly via signature extension.
  - We add rules $G(t',x,y) \to x$, $G(u',x,y) \to y$, with new ternary symbol G.
  - The resulting system remains non-ω-overlapping but it fails CON too.

Reducing the problem (ii)

- We can reduce the CON problem of a TRS to the CON problem of its constructor translation.
- The constructor translation of a TRS T is:
  - A constructor TRS (first-order functional program) T', with…
  - Back-and-forth translations between the terms of T and T', which preserves variables and…
  - preserves equations either way. (So this preserves and reflects CON.)
  - Aside: our translation also preserves and reflects SN (and WN).

Constructor Translation of a TRS

- Step 1: duplicate the signature; the new constructor TRS has a destructor F_d and a constructor F_c for every symbol F of the original signature
- Step 2: for every old rule $F(p_1,\ldots,p_n)\rightarrow r$ we get a new rule:
  $F_d(\lfloor p_1 \rfloor, \ldots, \lfloor p_n \rfloor) \rightarrow \lceil r \rceil$
- Step 3: for every non-variable pattern $G(q_1,\ldots,q_k)$ (strict subterm of a left-hand side of an old rule) we get a new rule:
  $G_d(\lfloor q_1 \rfloor, \ldots, \lfloor q_k \rfloor) \rightarrow G_c(\lfloor q_1 \rfloor, \ldots, \lfloor q_k \rfloor)$

Example

- Take Combinatory Logic: $K\ x\ y \to x, S\ x\ y\ z \to x\ z\ (y\ z)$.
- As a TRS this really is: $A(A(K, x), y) \to x,\ A(A(A(S, x), y), z) \to A\big(A(x, z), A(y, z)\big)$.
- Constructor translation (slightly abbreviated) for this:

$$A_d(A_c(K, x), y) \to x$$
$$A_d(K, x) \to A_c(K, x)$$
$$A_d(A_c(A_c(S, x), y), z) \to A_d\big(A_d(x, z), A_d(y, z)\big)$$
$$A_d(A_c(S, x), y) \to A_c(A_c(S, x), y)$$
$$A_d(S, x) \to A_c(S, x)$$

What does the translation do to overlaps?

- The translation does not introduce overlaps, except between pattern rules.
- If a TRS is non-ω-overlapping then its constructor-translation is "strongly almost non-ω-overlapping". This means: whenever two rules overlap then they are substitution instances of a common generalisation rule.
- For rules derived for patterns with root G we always have the generalisation $G_d(x_1, \ldots, x_n) \rightarrow G_c(x_1, \ldots, x_n)$
- This implies that all ω-overlaps between rules are trivial ("almost non-ω-overlapping"), but is even stronger than that.

Intermission: a tool for reasoning about terms

- Given a relation R between terms, we write $\tilde{R}$ write for the relation on terms defined as:

$$t\ \tilde{R}\ u \equiv\ \exists F \in \Sigma, t_1, \dots, t_n, u_1, \dots, u_n.\, t = F(t_1, \dots, t_n) \wedge u = F(u_1, \dots, u_n) \wedge \forall i.\, t_i\ R\ u_i$$

- Similarly, $t\hat{R}u$ and $t\overline{R}u$ express the corresponding relations when the shared symbol F is requested to be constructor (in Σ_c) or destructor (in Σ_d), respectively.
- A relation R is called Σ-closed iff $\tilde{R} \subseteq R$.

Observation: confluence vs. consistency

Why does confluence give us consistency?

- A system is confluent iff the joinability relation $\downarrow$ is transitive.
- The joinability relation can be defined like this:

$$\downarrow \equiv \mu x.\, id \cup x \cup x^{-1} \cup \tilde{x} \cup \rightarrow_R \cdot\, x$$

- Thus: joinability is by construction reflexive, symmetric and Σ-closed, and contains rewrite steps. It is just short of transitivity from being a congruence.
- It is also by construction consistent.

Note: there are other relations that share these properties with $\downarrow$, so they could take its part in consistency proofs.

Computational invariants

- We typically prove confluence by showing that $\downarrow$ is some kind of computational invariant. For this it needs to "survive" pattern matching. Relation-algebraically, it is this property:
- A relation R between terms is called constructor-compatible iff we have $\widehat{id} \cdot R \cdot \widehat{id} \subseteq \hat{R}$
- In long form: if two constructor-topped terms are related by R then they are topped by the same constructor and their direct subterms are pairwise related by R
- $\downarrow$ is always constructor-compatible

Pattern Matching; Rule Application

- Let p be a constructor term.
- Let $t = \sigma(p)$ and $u = \theta(p)$ be two substitution instances of p.
- If $t\ R\ u$ and R is constructor-compatible then σ and θ are pointwise related by R. (on variables occurring in p)
- If in addition R is Σ-closed then it must survive parallel rule application with the same rule.

We need more though…

- How can we make sure though that parallel rule applications are with the same rule?
- We have this result: whenever two redexes t and u are related by $\overline{=_c}$, where $=_c$ is a constructor-compatible equivalence, then t and u are instances of two ω-unifiable left-hand sides.
- Why? Informal reason: when we do ω-unification of we perform some equational transformations. If the terms we unify are constructor terms then each transformational step is sound for any constructor-compatible equivalence.

Another invariant

- The semi-joinability relation can be defined like this:

$$\Downarrow \; \equiv \mu x. id \cup x \cup x^{-1} \cup \tilde{x} \cup \overline{x} \cdot x \cup \overset{\varepsilon}{\rightarrow} \cdot x$$

- So, this relation $\Downarrow$ is reflexive, symmetric, Σ-closed, closed under prefixing with root-rewrite-steps, and it is closed under prefixing with itself on subterms of destructor-topped terms.
- Regardless of TRS, this relation is also constructor-compatible (and therefore consistent – when we view variables as constructors).
- So, this gives us a more relaxed invariant for consistency proofs than joinability. So, if $\Downarrow$ is transitive then we are home and dry.

One key difference to joinability

- Joinability is closed under prefixing with $\overline{\twoheadrightarrow_R}$, semi-joinability is closed under prefixing with $\overline{\Downarrow}$ - which is a symmetric relation.
- This gives extra flexibility when trying to construct a common “semi-reduct”.

Term-coalgebras

- Σ-coalgebras are sets whose elements (nodes) are term-like objects.
  - We may have additional structure, e.g. node labels.
  - The terms associated with nodes could be infinitary, and we may have the same term associated with more than one node.
- Term-coalgebras (for Σ) is the special case of sets of finite terms, closed under subterms.
- The $\tilde{R}$ notations carry over naturally to term-coalgebras (and indeed arbitrary Σ-coalgebras).

Transporting definitions

- We can view relations such as $\Downarrow$ as being defined (in the same way), for a particular coalgebra A.
- However, $\Downarrow_A$ is not just the restriction of $\Downarrow$ to AxA, because A is not required to include all terms – a redex may lose redex-status.
- In any case, $\Downarrow_A$ (on a term-coalgebra A) is a subrelation of $\Downarrow$ - because of monotonicity of the construction.
- Generally, if $t \Downarrow u$ holds then it is also the case that $t \Downarrow_A u$ for some finite term-coalgebra A.

Constructing an equivalence

- To prove that $\Downarrow_A$ is an equivalence for a finite term coalgebra A we simply build an equivalence relation which:
  - is constructor-compatible,
  - is a subrelation of $\Downarrow_A$,
  - is Σ-closed and contains $\overline{\Downarrow_A}^*$ as a subrelation, and which
  - includes "sufficiently many" redex contractions

How do we build it?

- As a union/find structure (with proof annotations).
- The node set of the structure is all of A.
- An edge from a to b requires that either

  $a \mathrel{\overline{\Downarrow}} b$ or $a \xrightarrow{\epsilon} b$ or $a \mathrel{\widehat{=_e}} b$ where $=_e$ is the equivalence defined by the structure.
- We merge equivalence classes by adding an edge to a root of the structure that points to another class.
- We prioritise $\overline{\Downarrow}$ edges over redex edges.

Proof graphs…

- The co-algebra is a set of finite terms (closed under subterms).
- These terms are the nodes of our union/find-structure.
- Our invariant relation $\Downarrow_A$ is reflexive (and symmetric), i.e. every term is related to itself
- The edge relation $\rightarrow_e$ of the union/find structure preserves the invariant: $\rightarrow_e \cdot \Downarrow_A \subseteq \Downarrow_A$
- Therefore all elements in a connected component are $\Downarrow_A$-related to each other.
- Overall: any proof graph defines a constructor-compatible equivalence, which is a subrelation of the invariant

Prioritisation

- In a proof graph we can connect at most one edge to a node
- For terms with a destructor-root this could be a redex-contraction or an "inner"-step.
- We prioritise inner steps, so that all equivalence classes of the relation $\overline{\Downarrow_A}^*$ are eventually connected in the graph.
- Consequence: the equivalence $=_e$ defined by the proof graph is necessarily Σ-closed, because it is a subrelation of $\Downarrow_A$; overall it is a constructor-compatible congruence relation.
- But is it the same as $=_R$?

Missing rewrite steps?

- Let $a \xrightarrow{\epsilon} b$ be any rewrite step of the co-algebra that is not an edge in the completed proof graph. Then either:
  - a is the (local) root of its equivalence class of $\overline{\Downarrow_A}^*$ and b is also in that class (and therefore $a =_e b$), or…
  - The local root of a is some redex c, with $c \xrightarrow{\epsilon} d$, and $c =_e d$. Then a and c are connected with inner steps in the proof graph (and we can ensure that these steps lie within $\overline{=_e}$)
  - Hence $a \overline{=_e} c$ and therefore b and d are related by $=_e$ (because it is a constructor-compatible congruence parallel steps stay within the invariant), and so $a =_e b$.

Consequences

- $\Downarrow_A$ is transitive (for finite A) for "well-behaved" TRSs
- $\Downarrow$ is transitive for well-behaved TRSs too:
  - If $t \Downarrow u$ and $u \Downarrow v$ then for some finite term-coalgebras A and B we have $t \Downarrow_A u$ and $u \Downarrow_B v$.
  - But $\mathrm{C} = A \cup B$ is then a term-coalgebra too, we get $t \Downarrow_C u$ and $u \Downarrow_C v$ by monotonicity, $t \Downarrow_C v$ by transitivity of $\Downarrow_C$, and $t \Downarrow v$ by monotonicity.
- Thus, "well-behaved" TRSs are consistent.
- Strongly almost non-ω-overlapping Constructor TRSs are "well-behaved".
- Therefore: non-ω-overlapping TRSs have unique normal forms.

Future Work

Almost non-ω-overlapping Constructor TRSs

Parallel steps could be with different rules, but we should still have our consistency invariant property.

Relaxing the condition on constructor-compatible equivalences:

We currently require that for all constructor-compatible sub-equivalences S of $\Downarrow$ that CT(S) holds for the contracta whenever $\bar{S}$ holds between redexes.

But: one does not need “all”, one only needs “all that are sufficiently large”.