universität innsbruck

CeTA 3.0 – Improved Support for ARI, CoCo and Infeasibility

René Thiemann
joint work with Akihisa Yamada, Dohan Kim and Teppei Saito
supported by the Austrian Science Fund (FWF) project I 5943

ARI Project Meeting, February 22, 2024

ARI Objective B: Four Tasks

- B.1: formalize techniques for commutation
  - commutation: ✓ CPP 2024 on parallel critical pairs (Dohan, Kiraku, Nao, René)
  - non-commutation: this talk, ongoing
- B.2: formalize techniques for infeasibility
  - previous talks, this talk, ongoing (Akihisa, Dohan, René, Teppei)
- B.3: add certificates for B.1 and B.2 to tools and CeTA
  - B.1: ✓, part of CPP 2024, but also this talk
  - B.2: this talk, ongoing
- B.4: formalize techniques for rewriting induction and ground confluence
  - sorted rewriting and pattern completeness (✓?) FSCD 2024 (Akihisa, René)
  - soundness of rewriting induction (Akihisa, Dohan, René): ongoing work
  - generation of certificates (Naoki, Takahito): format not fixed

CeTA 2.x – Problems prior to ARI-Format

- consider commutation and GCR
  - common property: signature matters
- consider CeTA 2.x and TPDB format
  - common property: signature is implicit
- resolving conflict in CeTA 2.x: maximally accepting
  - in commutation proofs, signature consists of symbols in input
  - in non-commutation proofs, signature is not restricted

$\Longrightarrow$ in CeTA 2.x it might have been possible to accept commutation proof and non-commutation proof of same two TRSs due to different implicit signature

CeTA 3.0 – Improved Support for ARI

- ARI format explicitly contains signature
- improved support:
  - CeTA 3.0 reads signature from ARI format
  - all proofs in CeTA using ARI format are now taking care of proper handling of signature
    - COM
    - INF
    - CR: signature ignored, but CR has signature extension property
    - GCR: ongoing work, will include signature
  - complexity proofs always allowed signature (to define basic terms)
  - termination proofs now consistently use “signature is not restricted” semantics

CeTA 3.0 – Improved Support for CoCo (Management)

- initial design of CPF used in CeTA 2.x
  - one self-contained CPF, containing input and proof, e.g.
    - TRS and non-termination proof
    - CTRS and confluence proof
  - similar to Isabelle theories, which also contain mixture of specs and proofs
  - no synchronization problem, as in "CR proof of COPS 120"
- problem in conducting competition
  - how to ensure that certificate on task

    is COPS 120 confluent?

    is not

    YES, {} is orthogonal (accepted by CeTA)
  - competition management somehow needs to extract input from certificate and then find conflict "COPS 120 $\neq$ {}"
- Akihisa's idea: instead of extracting input, let CeTA take input separately from competition management software

CeTA 3.0 – Improved Support for CoCo (Management)

- CeTA 3.0 extends Akihisa's idea, and splits a CPF into four parts
  - input: a TRS, two TRSs, CTRS + infeasibility query, ...
  - property: termination, confluence, ground-confluence, commutation, ...
  - answer: yes, no, upperbound $\mathcal{O}(n^2)$, ...
  - proof: proof-tree with applied methods and parameters as before
- tools still have to provide a self-contained CPF 3.0 file
- CeTA allows to overwrite input, property, answer in given CPF, e.g.

```
trs-conversion -f ARI -t CPF3 -o db34.cpf_input db34.ari # Fabian
ceta --inputf db34.cpf_input --property CR --answer YES fullCPF.xml
```

- advantage: all mismatches will be detected by CeTA itself, e.g.
  - tool says YES, but CPF contains a disproof
  - property is CR, but CPF contains a termination proof
  - input was some TRS, but CPF contains a proof for different TRS
    (if the same proof can be used for both TRSs, then this is accepted)

CeTA 3.0 – Improved Support for CoCo (Tool Authors)

- CPF 2 had several inconsistencies or non-uniform treatments
  - sometimes removed rules had to be specified, sometimes remaining rules
  - four different formats to specify joinable critical pairs, e.g. in <ruleLabeling>, <parallelClosed>, <pcpClosed>, <decreasingDiagrams>
- in CPF 3 and CeTA 3.0 the format has been simplified and unified
  - always specify removed rules
    (decreases certificate size from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$)
  - uniform way to specify joining sequences, choose between
    - $\mathit{left}, t_1, t_2, \ldots, t_n, \mathit{right}$ – intermediate terms suffice
    - specify upper bounds on steps – bfs; fails on conversions that are not joins
    - for WCR only: "rewrite to normal form"

CeTA 3.0 – Improved Support for CoCo (Tool Authors)

- CPF 2 is very verbose
  - terms and rules are always fully spelled out
  - certificates often contain several occurrences of the same rule
- CPF 3 is more concise
  - optional specification of a term index and a rule index
    - example: in compositional confluence criteria, one can specify

      consider sub-TRS {1,2,5}

      where 1,2,5 are rule indices that are specified once globally
  - several CPF-elements have been cleansed, e.g., no <arg>, <polynomial>, ...
- converter of CPF 2 to CPF 3 introduces perfect sharing of rules (and terms)
  - CPFs of termCOMP 2023: 8600 MB $\rightarrow$ 7200 MB
  - CPFs of CoCo 2023: 171 MB $\rightarrow$ 101 MB
- CeTA 3.0 directly expands parsed indices (future work)

CPF 2.0 vs 3.0

- demo
  - Example 39 of FSCD 2022 paper (Kiraku, Nao)
  - some rule-labeling proof (by Julian Nagele)

CeTA 3.0 – Improved Support for Infeasibility (and Non-CR and Non-COM)

- common theme for all these properties: show non-reachability property
  - non-CR and non-COM: given peak $s \xleftarrow[\mathcal{R}]{*} u \xrightarrow[\mathcal{S}]{*} t$, show that

    $$s \xrightarrow[\mathcal{S}]{*} v \xleftarrow[\mathcal{R}]{*} t$$

    is impossible
  - given oriented infeasibility query $s_1 \approx t_1, \ldots, s_n \approx t_n$, define $s := c(s_1, \ldots, s_n)$ and $t := c(t_1, \ldots, t_n)$ for fresh symbol c and show that

    $$s\sigma \xrightarrow[\mathcal{R}]{*} t\sigma$$

    is impossible
- available semantic solutions (Akihisa, Takahito): find discrimination pair (non-CR, non-COM) or co-rewrite pair (infeasibility) and solve some constraints involving $\mathcal{R}$, $\mathcal{S}$, s, t, e.g., $\mathcal{R} \subseteq \succsim$ and $t \succ s$ for infeasibility

Discrimination Pairs and Co-Rewrite Pairs

- co-rewrite pair $(\succ, \succsim)$
  - $\succsim \cap \prec = \emptyset$
  - $\succsim$ is transitive and reflexive, closed under contexts and substitutions
  - $\succ$ is irreflexive and closed under substitutions
- discrimination pair $(\succ, \succsim)$
  - $\succsim \circ \succ \subseteq \succ$
  - $\succsim$ is closed under contexts and substitutions
  - $\succ$ is irreflexive
- CeTA 2.x
  - historical interface for reduction orders: always demand that $\succ$ is SN
  - $\Longrightarrow$ cannot exploit power of relations that are irreflexive, but not SN
- CeTA 3.0
  - complete redesign of interface for relations on terms
  - basic properties can individually be demanded
  - wrapper functions for common cases

New Interface

- considers three relations (S = strict, NS = non-strict, NST = non-strict top)
- simplified(!) properties (dropped argument filters, Ce-compatibility,...)

```
"rel_impl_prop impl ≡ ∃ S NS NST.
    — ‹implementation approximates real relations›
    (∀ st. (isOK(rel_impl.s impl st) ⟶ st ∈ S) ∧
           (isOK(rel_impl.ns impl st) ⟶ st ∈ NS) ∧
           (isOK(rel_impl.nst impl st) ⟶ st ∈ NST))
    — ‹unconditional properties›
    ∧ irrefl S
    ∧ ctxt.closed NS
    ∧ subst.closed NS
    ∧ trans NS
    ∧ refl NS
    — ‹properties that can be tested via flags›
    ∧ (isOK(rel_impl.standard impl) ⟶
        trans S ∧ S ⊆ NS ∧ S O NS ⊆ S ∧ NS O S ⊆ S ∧ subst.closed NST
      ∧ trans NST ∧ NST O S ⊆ S ∧ S O NST ⊆ S)
    ∧ (isOK(rel_impl.mono impl []) ⟶ ctxt.closed S)
    ∧ (isOK(rel_impl.top_mono impl) ⟶ top_mono NS NST)
    ∧ (isOK(rel_impl.top_refl impl) ⟶ refl NST)
    ∧ (isOK(rel_impl.SN impl) ⟶ SN S)
    ∧ (isOK(rel_impl.subst_s impl) ⟶ subst.closed S)
    ∧ (isOK(rel_impl.co_rewr impl) ⟶ NS ∩ S⁻¹ = {})
    ∧ (∀ cm cc. isOK(rel_impl.cpx impl cm cc) ⟶ deriv_bound_measure_class S cm cc)"
```

Co-Rewrite Pairs in New Interface

- specification via interface is simple

```
definition rel_impl_co_rewrite_pair where
 "rel_impl_co_rewrite_pair impl = do {
      rel_impl.co_rewr impl;
      rel_impl.subst_s impl
    } <+? (λ_. showsl_lit (STR ''error message⏎''))"
```

- property is easy to use

```
lemma rel_impl_co_rewrite_pair: assumes "rel_impl_prop impl"
  and "isOK(rel_impl_co_rewrite_pair impl)"
  and "isOK(rel_impl_s impl s_constraints)" "isOK(rel_impl_ns impl ns_constraints)"
shows "∃ S NS. co_rewrite_pair S NS ∧ set s_constraints ⊆ S ∧ set ns_constraints ⊆ NS"
```

- discrimination pairs are similar

CeTA 3.0 has New Relations via New Interface

- WPO has been generalized in formalization (René)
  - example: SN of underlying order is propagated, but not demanded
- co-WPO (Dohan, René)
  - formalization insight: lexicographic comparison cannot be changed to multiset comparison as for WPO
- linear polynomial interpretations over $\mathbb{Z}$ (Dohan)
- MSPO and GWPO (Teppei)
- tuple interpretations (Akihisa, René, ongoing)

you are invited to add these relations to your certificate generating tool; increase the power of certified non-CR, non-COM, infeasibility proving

Summary

- big restructuring efforts have gone into CeTA 3.0 and CPF 3.0
- achieved
  - better support for certification in competitions
  - more consistency in CPF 3 $\Longrightarrow$ ask for support of CPF 3 in tools
  - reduced size of certificates
  - new term orders became available $\Longrightarrow$ ask for support of these in tools
- unclear
  - plans to publish restructuring efforts
    - is there a plan to write ARI-infrastructure paper?
    - $\Longrightarrow$ interest: add section on new certification approach
  - plans to publish formalization of new orders
    - let's discuss among Akihisa, Dohan, René, Teppei

Questions?