Proving Ground Confluence of Equational Specifications Modulo Axioms

Francisco Durán[1], José Meseguer[2], and Camilo Rocha[3]

[1] Universidad de Málaga, Málaga, Spain
[2] University of Illinois, Urbana-Champaign, IL, USA
[3] Pontificia Universidad Javeriana, Cali, Colombia

Abstract. Terminating functional programs should be *deterministic*, i.e., should evaluate to a *unique* result, regardless of the evaluation order. For equational functional programs such determinism is exactly captured by the *ground confluence* property. For terminating equations this is equivalent to *ground local confluence*, which follows from *local confluence*. Checking local confluence by computing critical pairs is the *standard* way to check ground confluence. The problem is that some perfectly reasonable equational programs are *not* locally confluent and it can be very hard or even impossible to make them so by adding more equations. We propose a three-step strategy to prove that an equational program *as is* is ground confluent: *First*: apply the strategy proposed in [8] to use non-joinable critical pairs as *completion hints* to either achieve local confluence or reduce the number of critical pairs. *Second*: use the *inductive inference system* proposed in this paper to prove the remaining critical pairs ground joinable. *Third*: to show ground confluence of the original specification, prove also ground joinable the equations added. These methods apply to order-sorted and possibly conditional equational programs modulo axioms such as, e.g., Maude functional modules.

1 Introduction

Functional programs should be *deterministic*; that is, if they terminate for a given input, they should return a *unique* value, regardless of the evaluation order. *Ground confluence* is the precise characterization of such determinism for functional equational programs associated to equational theories of the form $\mathcal{E} = (\Sigma, E \uplus B)$, were B are structural axioms and E are, possibly conditional, equations that are executed as rewrite rules $\overrightarrow{E}$ *modulo* B. Therefore, for execution purposes, all the relevant information is contained in the rewrite theory $\mathcal{R}_{\mathcal{E}} = (\Sigma, B, \overrightarrow{E})$. Since ground confluence is essential both for correct execution and for almost any form of formal verification about properties of $\mathcal{E}$ and $\mathcal{R}_{\mathcal{E}}$, methods to prove ground confluence are very important.

The *standard method* to do so for a terminating equational program $\mathcal{R}_{\mathcal{E}} = (\Sigma, B, \overrightarrow{E})$ is to: (i) prove that it is indeed *operationally terminating* (and if Σ is order-sorted, also *sort decreasing*); and then (ii) since operational termination plus local confluence imply confluence, prove the *stronger* property that $\mathcal{R}_{\mathcal{E}}$ is locally confluent (modulo B). This tends to work well in many cases, but not always. The thorny issue addressed in this paper is what to do when this standard method does not work.

In [8], the wild goose chase for a convergent specification by attempting a Knuth-Bendix completion of $\mathcal{E}$ was explicitly discouraged, since it can often lead to an infinite loop and, even if it were to succeed, can result in a highly bloated and hard to understand specification. Instead, the following *incremental strategy* in the spirit of Knuth-Bendix was suggested: since failure of a proof of local confluence will generate a set of unjoinable *critical pairs* characterizing the most general cases in which rules cannot be shown confluent, such critical pairs can be used as very useful *hints* for a user to try to either: (i) orient a critical pair as a rule and add it to the specification; or (ii) if the critical pair has the form $C[u] = C[v]$ with C a common context, orient instead $u = v$ and add it to the specification; or (iii) *generalize* $u = v$ in cases (i) and (ii) into a more general $u' = v'$ that has $u = v$ as a substitution instance and add an oriented version of $u' = v'$ to the specification. In this way, we obtain a new specification $\mathcal{R}_{\mathcal{E}'} = (\Sigma, B, \overrightarrow{E} \uplus \overrightarrow{G})$, where $\overrightarrow{G}$ are the new oriented equations added by methods (i)–(iii). If $\mathcal{R}_{\mathcal{E}'}$ is locally confluent, operationally terminating, and sort-decreasing, we are done; otherwise, we can *iterate* the process with the critical pairs obtained for $\mathcal{R}_{\mathcal{E}'}$.

In practice, this incremental strategy works reasonably well, but not always. Furthermore, it raises the following unsolved questions:

1. Have we *changed* the initial algebra semantics? That is, do the original $\mathcal{R}_{\mathcal{E}}$ and its extension $\mathcal{R}_{\mathcal{E}'}$ have the same initial algebra when viewed as equational theories? If only additions of type (i) are made, this is always true; but additions of type (ii)–(iii) are often needed in practice.
2. Was the *original* specification $\mathcal{R}_{\mathcal{E}}$ already ground confluent? That is, can we use $\mathcal{R}_{\mathcal{E}'}$ as the proverbial "Wittgenstein ladder" that we can kick away *after* we have proved its local confluence?
3. What do we do when we run into a *wall*? Specifically, the "wall" of having an equation $u = v$ obtained by methods (i)–(iii) above that *cannot* be oriented because it would lead to non-termination.

Our Contributions. This paper provides new methods that answer these three questions and can greatly help in proving an original specification ground confluent. In a nutshell, a more general and powerful strategy is proposed with three steps: *First*: use the above-described strategy from [8] as far as it can go. *Second*: if you hit the wall of non-orientability for some critical pairs (Question 3), prove the *ground joinability* of such remaining pairs by the *inductive methods* presented in this work. *Third*: to ensure preservation of the initial algebra semantics (Question 1) and the ground confluence of the original specification (Question 2), use the same inductive methods to prove ground joinability of all the equations added along the first step. Of course, one could skip the first step altogether and merge the second and third steps into one; but this may require a considerably bigger effort, since the whole point of taking the first step is to *greatly reduce* the number of pairs to be proved ground joinable. Furthermore, the user may have made an *actual mistake* in the original specification $\mathcal{R}_{\mathcal{E}}$, so that the second and third steps become meaningless. In such a case, the first step can be quite helpful in identifying such mistakes and help the user *restart* the process with a new specification.

Paper organization. Preliminaries are gathered in Section 2. The strategy's First Step is illustrated in Section 3 by a hereditarily finite sets specification that does indeed run

into a non-orientability wall. The inductive inference system for ground confluence is presented and proved sound in Section 4, and is illustrated by proving the inductive joinability of the non-orientable critical pair from Section 3, thus illustrating the Second Step. The Third Step is then illustrated in detail for the running example in Section 5. Some related work and conclusions are discussed in Section 6. Results for the mechanized proofs and proofs for auxiliary results can be found in the Appendices A-D.

2 Preliminaries

Notation on terms, term algebras, and equational theories is used as, e.g., in [10]. An *order-sorted signature* Σ is a tuple $\Sigma = (S, \leq, F)$ with a finite poset of sorts $(S, \leq)$ and set of function symbols F typed with sorts in S. The binary relation $\equiv_{\leq}$ denotes the equivalence relation $(\leq \cup \geq)^+$ generated by $\leq$ on S and its point-wise extension to strings in S^*. The function symbols in F can be subsort-overloaded. For any sort $s \in S$, the expression $[s]$ denotes the connected component of s, that is, $[s] = [s]_{\equiv_{\leq}}$. A *top sort* in Σ is a sort $s \in S$ such that for all $s' \in [s]$, $s' \leq s$. Let $X = \{X_s\}_{s \in S}$ be an S-indexed family of disjoint variable sets with each X_s countably infinite. For any $s \in S$, let $X_{\leq s} = \bigcup_{s' \in S \wedge s' \leq s} X_{s'}$. The *set of terms of sort s* and the *set of ground terms of sort s* are denoted, respectively, by $T_\Sigma(X)_s$ and $T_{\Sigma,s}$; similarly, $T_\Sigma(X)$ and T_Σ denote, respectively, the set of terms and the set of ground terms. $\mathcal{T}_\Sigma(X)$ and $\mathcal{T}_\Sigma$ denote the corresponding order-sorted Σ-term algebras. All order-sorted signatures are assumed *preregular* [10], i.e., each Σ-term t has a unique *least sort* $ls(t) \in S$ s.t. $t \in T_\Sigma(X)_{ls(t)}$. It is also assumed that Σ has *nonempty sorts*, i.e., $T_{\Sigma,s} \neq \emptyset$ for each $s \in S$. The *set of variables* of t is written $vars(t)$ and for a list of terms $t_1, \ldots, t_n$, $vars(t_1, \ldots, t_n) = vars(t_1) \cup \cdots \cup vars(t_n)$.

A *substitution* is an S-indexed mapping $\theta \in [X \longrightarrow T_\Sigma(X)]$ that is different from the identity only for a finite subset of X and such that $\theta(x) \in T_\Sigma(X)_s$ if $x \in X_s$, for any $x \in X$ and $s \in S$. The expression $\theta|_Y$ denotes the restriction of θ to a family of variables $Y \subseteq X$. The *domain* of θ, denoted $dom(\theta)$, is the subfamily of X such that $x \in dom(\theta)$ iff $\theta(x) \neq x$, for each $x \in X$. If $dom(\theta) = \{x_1, \ldots, x_n\}$ we write $\theta = \{x_1 \mapsto \theta(x_1), \ldots, x_n \mapsto \theta(x_n)\}$. The *range* of θ is the set $ran(\theta) = \bigcup\{vars(\theta(x)) \mid x \in dom(\theta)\}$. Substitutions extend homomorphically to terms in the natural way. A substitution θ is called *ground* iff $ran(\theta) = \emptyset$. The application of a substitution θ to a term t is denoted by $t\theta$ and the composition (in diagrammatic order) of two substitutions θ_1 and θ_2 is denoted by $\theta_1\theta_2$, so that $t\theta_1\theta_2$ denotes $(t\theta_1)\theta_2$. A *context* C is a λ-term of the form $C = \lambda x_1, \ldots, x_n.c$ with $c \in T_\Sigma(X)$ and $\{x_1, \ldots, x_n\} \subseteq vars(c)$; it can be viewed as an n-ary function $(t_1, \ldots, t_n) \mapsto C(t_1, \ldots, t_n) = c\theta$, where $\theta(x_i) = t_i$ for $1 \leq i \leq n$ and $\theta(x) = x$ for $x \notin \{x_1, \ldots, x_n\}$.

An *equational theory* is a tuple (Σ, E), with Σ an order-sorted signature and E a finite collection of (possibly conditional) Σ-equations. An equational theory $\mathcal{E} = (\Sigma, E)$ induces the congruence relation $=_\mathcal{E}$ on $T_\Sigma(X)$ defined for $t, u \in T_\Sigma(X)$ by $t =_\mathcal{E} u$ iff $\mathcal{E} \vdash t = u$, where $\mathcal{E} \vdash t = u$ denotes $\mathcal{E}$-provability by the deduction rules for order-sorted equational logic in [12]. For the purpose of this paper, such inference rules, which are analogous to those of many-sorted equational logic, are even simpler thanks to the assumption that Σ has nonempty sorts, which makes unnecessary the explicit treatment of universal quantifiers. The expressions $\mathcal{T}_\mathcal{E}(X)$ and $\mathcal{T}_\mathcal{E}$ (also written $\mathcal{T}_{\Sigma/E}(X)$

and $\mathcal{T}_{\Sigma/E}$) denote the quotient algebras induced by $=_{\mathcal{E}}$ on the term algebras $\mathcal{T}_{\Sigma}(X)$ and $\mathcal{T}_{\Sigma}$, respectively. $\mathcal{T}_{\Sigma/E}$ is called the *initial algebra* of (Σ, E).

We assume acquaintance with the usual notions of position p in a term t, subterm $t|_p$ at position p, and term replacement $t[u]_p$ at position p (see, e.g., [5]). A *rewrite theory* is a tuple $\mathcal{R} = (\Sigma, E, R)$ with (Σ, E) an order-sorted equational theory and R a finite set of possibly conditional Σ-rules, with conditions being a conjunction of Σ-equalities. A rewrite theory $\mathcal{R}$ induces a rewrite relation $\rightarrow_{\mathcal{R}}$ on $T_{\Sigma}(X)$ defined for every $t, u \in T_{\Sigma}(X)$ by $t \rightarrow_{\mathcal{R}} u$ iff there is a rule $(l \rightarrow r \text{ if } \phi) \in R$, a term t', a position p in t', and a substitution $\theta : X \longrightarrow T_{\Sigma}(X)$ satisfying $t =_E t' = t'[l\theta]_p$, $u =_E t'[r\theta]_p$, and $(\Sigma, E) \vdash \phi\theta$. The tuple $\mathcal{T}_{\mathcal{R}} = (\mathcal{T}_{\Sigma/E}, \rightarrow_{\mathcal{R}}^*)$ is called the *initial reachability model of* $\mathcal{R}$ [3].

In this paper we will mostly focus on rewrite theories of the form $\mathcal{R}_{\mathcal{E}} = (\Sigma, B, \overrightarrow{E})$ associated to an equational theory $\mathcal{E} = (\Sigma, E \uplus B)$, were: (i) B are decidable structural axioms whose equations $u = v \in B$ are linear (no repeated variables in either u or v) and regular (same variables in u and v), for which a matching algorithm exists, and (ii) the possibly conditional rewrite rules $\overrightarrow{E}$ are *strictly B-coherent* [13]. Under such assumptions, the rewrite relation $t \rightarrow_{\mathcal{R}_{\mathcal{E}}} u$ holds iff there exists u' such that $u' =_B u$, and $t \rightarrow_{\overrightarrow{E},B} u'$, where, by definition, $t \rightarrow_{\overrightarrow{E},B} u'$ iff there exists a rule $(l \rightarrow r \text{ if } \phi) \in \overrightarrow{E}$, a position p in t and a substitution θ such that $t|_p =_B l\theta$, $u' = t[r\theta]_p$, and $\mathcal{R}_{\mathcal{E}} \vdash \phi\theta$. We will assume throughout that the rules $\overrightarrow{E}$ are *always strictly B*-coherent. We finally assume that the axioms B are: (i) *sort-preserving*, i.e., for each $(u = v) \in B$ and substitution σ we have $ls(u\sigma) = ls(v\sigma)$; and (ii) *term-size preserving*[4], i.e., if $t =_B t'$, then $|t| = |t'|$.

Appropriate requirements are needed to make an equational theory $\mathcal{E} = (\Sigma, E \uplus B)$ *admissible* as an equational program, i.e., for making $\mathcal{R}_{\mathcal{E}} = (\Sigma, B, \overrightarrow{E})$ *executable* in languages such as Maude [4]. In this paper, besides the above assumptions about B and $\overrightarrow{E}$, we assume that the rules in $\overrightarrow{E}$ are sort-decreasing, operationally terminating, and ground confluent modulo B. The rewrite rules $\overrightarrow{E}$ are *sort decreasing* modulo B iff for each $(t \rightarrow u \text{ if } \gamma) \in \overrightarrow{E}$ and substitution θ, $ls(t\theta) \geq ls(u\theta)$ if $\mathcal{R}_{\mathcal{E}} \vdash \gamma\theta$. $\mathcal{R}_{\mathcal{E}} = (\Sigma, B, \overrightarrow{E})$ is *operationally terminating* modulo B [6] iff there is no infinite well-formed proof tree in $(\Sigma, B, \overrightarrow{E})$. Call $t, t' \in T_{\Sigma}(X)$ *joinable* in $\mathcal{R}_{\mathcal{E}}$, denoted $t \downarrow_{\mathcal{R}_{\mathcal{E}}} t'$ iff there exist u, v such that $t \rightarrow_{\overrightarrow{E},B}^* u$, $t' \rightarrow_{\overrightarrow{E},B}^* v$, and $u =_B v$. Call $\mathcal{R}_{\mathcal{E}} = (\Sigma, B, \overrightarrow{E})$ *confluent* (resp., *ground confluent*) modulo B iff for all $t, t_1, t_2 \in T_{\Sigma}(X)$ (resp., for all $t, t_1, t_2 \in T_{\Sigma}$), if $t \rightarrow_{\overrightarrow{E},B}^* t_1$ and $t \rightarrow_{\overrightarrow{E},B}^* t_2$, then $t_1 \downarrow_{\mathcal{R}_{\mathcal{E}}} t_2$. For $\mathcal{R}_{\mathcal{E}} = (\Sigma, B, \overrightarrow{E})$ to have good executability properties as a terminating equational program, the following requirements are needed : (a) sort decreasingness, (b) operational termination, and (c) ground confluence. If conditions (a)–(c) are met, we call $\mathcal{R}_{\mathcal{E}}$ *ground convergent*. $\mathcal{R}_{\mathcal{E}}$ is called *convergent* if it satisfies the stronger requirements of sort-decreasingness, operational termination, and confluence.

[4] For combinations of associativity, commutativity, and identity axioms, this last condition only rules out identity axioms. However, both for termination and confluence analysis purposes, identity axioms can always be turned into convergent rewrite rules modulo associativity and/or commutativity axioms, as explained in [7].

3 An Equational Specification for Hereditarily Finite Sets

When checking the confluence of an equational specification, the CRC tool [8, 9] provides as result a set of critical pairs that cannot be joined automatically by its built-in heuristics. They are proof obligations that can either be proved joinable or used as guidance for modifying the input specification. The methodology proposed in [8] for using the CRC tool suggests that critical pairs can help in identifying theorems of the original specification which, when added to it, may lead to a confluent or ground confluent specification. However, as the example of HF-SETS presented in this section shows, the analysis of critical pairs to modify a specification, though a useful first strategy, may be insufficient to make the specification ground confluent. Other techniques, such as the ones presented in Section 4, may be needed.

Consider the specification of hereditarily finite sets below, namely, of finite sets whose elements are all hereditarily finite sets (see, e.g., [11]). The recursive definition of well-founded hereditary sets has the empty set as the base case and if $s_1, \ldots, s_k$ are hereditarily finite, then so is $\{s_1, \ldots, s_k\}$. These sets play a key role in axiomatic set theory because they are a model of all the axioms of set theory except for the axiom of infinity. Furthermore, as the methods developed in this work will show, the initial model of the HF-SETS specification below is a *consistent* model of set theory without the axiom of infinity.

```
fmod HF-SETS is protecting BOOL-OPS .
  sorts Magma Set .
  subsort Set < Magma .
  op _,_ : Magma Magma -> Magma [ctor assoc comm] .
  op {_} : Magma -> Set [ctor] .
  op {} : -> Set [ctor] .
  vars M M' : Magma .  vars S S' : Set .
    eq [01]: M, M = M .
  op _in_ : Magma Set -> Bool .            *** set membership for several elements
    eq [11]: S in {S} = true .
    eq [12]: S in {} = false .
    eq [13]: {} in {{M}} = false .
    eq [14]: {M} in {{}} = false .
    eq [15]: {M} in {{M'}} = M in {M'} and M' in {M} .
    eq [16]: S in {S', M} = S in {S'} or S in {M} .
    eq [17]: (S, M) in S' = (S in S') and (M in S') .
  op _~_ : Set Set -> Bool .               *** set equality
    eq [21]: S ~ S' = (S <= S') and (S' <= S) .
  op _<=_ : Set Set -> Bool .              *** set containment
    eq [31]: {} <= S = true .
    eq [32]: {M} <= S = M in S .
  op _U_ : Set Set -> Set [assoc comm] .   *** union
    eq [41]: S U {} = S .
    eq [42]: {M} U {M'} = {M, M'} .
    eq [43]: S U {M} U {M'} = S U {M, M'} .
  op P : Set -> Set .                      *** powerset
    eq [51]: P({}) = {{}} .
    eq [52]: P({S}) = {{},{S}} .
    eq [53]: P({S, M}) = P({M}) U augment(P({M}), S) .
  op augment : Set Set -> Set .            *** augmentation
    eq [61]: augment({}, S) = {} .
    eq [62]: augment({S}, S') = {{S'} U S} .
    eq [63]: augment({M, M'}, S) = augment({M}, S) U augment({M'}, S) .
  op _&_ : Set Set -> Set .                *** intersection
    eq [71]: {} & S = {} .
   ceq [72]: {S} & S' = {S} if S in S' = true .
   ceq [73]: {S} & S' = {} if S in S' = false .
   ceq [74]: {S, M} & S' = {S} U ({M} & S') if S in S' = true .
```

```
    ceq [75]: {S, M} & S' = {M} & S' if S in S' = false .
endfm
```

The Church-Rosser check of the HF-SETS module using the CRC tool says that the specification is sort-decreasing, but it cannot show that it is locally confluent, returning eight critical pairs as proof obligations. At this point, there are two alternatives: either (i) we try to prove the ground joinability of these critical pairs to conclude that the specification is locally ground confluent, or (ii) we follow the iterative strategy proposed in [8] to get a locally confluent specification or at least reduce the number of critical pairs for which a proof of joinability is necessary. In the rest of this section, we explore the second alternative. The first alternative will be revisited after the second one is exhausted (both are useful) in Section 5.

The following one is one of the critical pairs returned by the check:

```
cp HF-SETS1123 for 11 and 15 true = M':Magma in {M':Magma} .
```

It comes from the overlap of equations 11 and 15. Although there are equations for all possible instances of the term `M in {M}`, Maude cannot reduce it as magmas. We can attempt adding equations to reduce it as follows:

```
fmod HF-SETS-0 is protecting HF-SETS .
  vars M M' : Magma .
  eq [18]: M in {M} = true .
  eq [19]: M in {M', M} = true .
endfm
```

A check of the Church-Rosser property for HF-SETS-0 returns seven critical pairs. Let us consider one of these critical pairs:

```
cp HF-SETS-095 for 01 and 63
  augment({M':Magma}, S:Set) = augment({M':Magma}, S:Set) U augment({M':Magma}, S:Set) .
```

This critical pair comes from the overlap of equations 01 and 63. Indeed, this critical pair cannot be further reduced because there is no idempotency equation for the union operator on sets. We can see the same problem in other four of the critical pairs reported by the tool. Although S U S = S could be proven in HF-SETS-0, there is the alternative option of extending the specification with an idempotency equation for set union.

```
fmod HF-SETS-1 is protecting HF-SETS-0 .
  var S : Set .
  eq [44]: S U S = S .
endfm
```

The Church-Rosser checker tool produces the following output for HF-SETS-1:

```
The following critical pairs must be proved joinable:
  cp HF-SETS-118 for 53 and 53
     P({#6:Magma}) U augment(P({#6:Magma}), S:Set) U augment(P({#6:Magma}) U
     augment(P({#6:Magma}), S:Set), #1:Set)
   = P({#6:Magma}) U augment(P({#6:Magma}), #1:Set) U augment(P({#6:Magma}) U
     augment(P({#6:Magma}), #1:Set), S:Set).
  cp HF-SETS-1355 for 01 and 53
     P({#3:Magma}) U augment(P({#3:Magma}), S:Set)
   = P({#3:Magma}) U augment(P({#3:Magma}), S:Set) U augment(P({#3:Magma}) U
     augment(P({#3:Magma}), S:Set), S:Set).
  The module is sort-decreasing.
```

A careful study of these critical pairs suggests the need for an equation to apply augment over the union operator.

```
fmod HF-SETS-2 is protecting HF-SETS-1 .
  vars S S' T : Set .
  eq [64]: augment(S U S', T) = augment(S, T) U augment(S', T) .
endfm
```

The critical pairs get further reduced in HF-SETS-2, but two remain:

```
The following critical pairs must be proved joinable:
  cp HF-SETS-218 for 53 and 53
     P({#6:Magma}) U augment(P({#6:Magma}), S:Set) U augment(P({#6:Magma}), #1:Set) U
     augment(augment(P({#6:Magma}), S:Set), #1:Set)
   = P({#6:Magma}) U augment(P({#6:Magma}), S:Set) U augment(P({#6:Magma}), #1:Set) U
     augment(augment(P({#6:Magma}), #1:Set), S:Set).
  cp HF-SETS-2411 for 01 and 53
     P({#3:Magma}) U augment(P({#3:Magma}), S:Set)
   = P({#3:Magma}) U augment(P({#3:Magma}), S:Set)
                   U augment(augment(P({#3:Magma}), S:Set), S:Set).
  The module is sort-decreasing.
```

The second critical pair suggests the need for an equation handling the repeated application of the augment operator.

```
fmod HF-SETS-3 is protecting HF-SETS-2 .
  vars S T : Set .
  eq [65]: augment(augment(S, T), T) = augment(S, T) .
endfm
```

However, one critical pair remains in HF-SETS-3:

```
Church-Rosser check for HF-SETS-3
The following critical pairs must be proved joinable:
  cp HF-SETS-318 for 53 and 53
     P({#6:Magma})U augment(P({#6:Magma}),S:Set)U augment(P({#6:Magma}),#1:Set)U
     augment(augment(P({#6:Magma}),S:Set),#1:Set)
   = P({#6:Magma})U augment(P({#6:Magma}),S:Set)U augment(P({#6:Magma}),#1:Set)U
     augment(augment(P({#6:Magma}),#1:Set),S:Set).
  The module is sort-decreasing.
```

It is not obvious at all how to eliminate this critical pair, since adding the equation

```
 eq augment(augment(S, S'), T) = augment(augment(S, T), S') .
```

would make the specification *non-terminating*. This suggests that the second approach, i.e., the strategy of trying to complete the specification by analyzing the unjoinable critical pairs has now been exhausted. However, the original problem has now been reduced to a *single* critical pair. At this point, the best approach is to prove the *inductive joinability* of the critical pair HF-SETS-318 obtained in the check of HF-SETS-3, and thus conclude that the specification is ground locally confluent. Section 4 presents techniques for carrying out such inductive proofs. Indeed, it will also present results showing that the *original specification was already ground confluent!*, without the need for the extra equations added in the process. The specification is terminating. Indeed, the MTT tool is able to find termination proofs for all the versions of the HF-SETS module, and specifically for HF-SETS-3 (see Appendix B). A proof of the sufficient completeness of the specification can be found in Appendix C.

Finally, note that if an added equation comes from orienting a critical pair, it is a logical consequence of the specification and therefore the new specification has the *same* initial model of the old one. Although the additional equations added during the process may not be those obtained from critical pairs as such, proving that they are

ground joinable is enough to show that they are actually *inductive lemmas*, and therefore –as explained in more detail in Theorem 6 in Section 4– that they both preserve the initial algebra semantics *and* can be *removed* from the original specification.

4 Proving Ground Joinability

This section presents inductive techniques for proving ground joinability for rewrite theories associated to equational specifications. These techniques are presented as meta-theorems about the ground reachability relation induced by a rewrite theory and are used to justify the inference system also presented in this section.

Definition 1. *Let $\mathcal{R}$ be a rewrite theory with signature $\Sigma = (S, \leq, F)$ and $t, u \in T_\Sigma(X)_s$ for some $s \in S$. The terms t and u are called:*

1. $\mathcal{R}$-joinable*, written $\mathcal{R} \vdash (\forall X)\, t \downarrow u$, iff there is $v \in T_\Sigma(X)_s$ such that $\mathcal{R} \vdash (\forall X)\, t \rightarrow^* v$ and $\mathcal{R} \vdash (\forall X)\, u \rightarrow^* v$.*
2. ground $\mathcal{R}$-joinable*, written $\mathcal{R} \Vdash (\forall X)\, t \downarrow u$, iff $\mathcal{R} \vdash t\theta \downarrow u\theta$ for all ground substitutions $\theta \in [X \longrightarrow T_\Sigma]$.*

The authors of [16] investigate constructor-based inductive techniques for proving ground joinability. They distinguish two notions of constructors for a rewrite theory $\mathcal{R}$, namely, one for the equations and another one for the rules in $\mathcal{R}$.

Definition 2 (Defs. 5 and 6 [16]). *Let $\mathcal{R} = (\Sigma, E, R)$ be a rewrite theory with underlying equational theory $\mathcal{E} = (\Sigma, E)$. A* constructor signature pair *for $\mathcal{R}$ is a pair (Υ, Ω) of order-sorted subsignatures $\Upsilon = (S, \leq, F_\Upsilon) \subseteq \Omega = (S, \leq, F_\Omega)$. The sets of terms $T_\Upsilon = \{T_{\Upsilon,s}\}_{s\in S}$ and $T_\Omega = \{T_{\Omega,s}\}_{s\in S}$ are called, respectively, E-*constructor terms *and $\mathcal{R}$-*constructor terms*. The rewrite theory $\mathcal{R}$ is called:*

1. E-sufficiently complete relative to Ω *iff* $(\forall s \in S)(\forall t \in T_{\Sigma,s})(\exists u \in T_{\Omega,s})\ \mathcal{E} \vdash t = u$.
2. $\mathcal{R}$-sufficiently complete relative to Υ *iff* $(\forall s \in S)(\forall t \in T_{\Sigma,s})(\exists v \in T_{\Upsilon,s})\ \mathcal{R} \vdash t \rightarrow^* v$.
3. sufficiently complete relative to (Υ, Ω) *iff (1) and (2) hold.*

The notion of sufficient completeness for a rewrite theory $\mathcal{R}$ relative to a constructor signature pair (Υ, Ω) is that $\Omega \subseteq \Sigma$ are the constructors for the equations and $\Upsilon \subseteq \Omega$ the constructors for the rules, thus including the standard concept of constructor for equational specifications as a special case. The intuition behind equational constructor terms is that any ground Σ-term should be *provably equal* to a term in T_Ω and for rewrite constructors that any Σ-term should be *rewritable* to a term in T_Υ.

It is sufficient to consider all $\mathcal{R}$-constructor terms in $T_{\Upsilon,s}$ when inducting on a variable x of sort s, for a proof on inductive joinability in $\mathcal{R}$ to be sound.

Theorem 1 (Thm. 6 [16]). *Let $\mathcal{R}$ be a rewrite theory with signature $\Sigma = (S, \leq, F)$ and $t, u \in T_\Sigma(X)_s$ for some $s \in S$. If $\mathcal{R}$ is sufficiently complete relative to the constructor signature pair (Υ, Ω), then $\mathcal{R} \Vdash (\forall X)\, t \downarrow u$ iff $(\forall \eta \in [X \longrightarrow T_\Upsilon])\ \mathcal{R} \vdash t\eta \downarrow u\eta$.*

$$\frac{\mathcal{R} \vdash (\forall X)\, t \downarrow u}{\mathcal{R} \Vdash (\forall X)\, t \downarrow u}\ \textsc{Join} \qquad \frac{\mathcal{R} \Vdash (\forall X)\, t \downarrow u}{\mathcal{R} \Vdash (\forall X)\, C[t] \downarrow C[u]}\ \textsc{Ctx} \qquad \frac{\mathcal{R} \Vdash (\forall X)\, t \downarrow u}{\mathcal{R} \Vdash (\forall X)\, t\theta \downarrow u\theta}\ \textsc{Gral}$$

Fig. 1: Inference rules for proving joinability for a rewrite theory $\mathcal{R}$ by rewrite-based reasoning, and inductive reasoning for contexts and substitution instances.

Figure 1 presents the Join, Ctx and Gral inference rules for proving joinability for a rewrite theory $\mathcal{R}$, respectively, by rewrite-based reasoning, inductive reasoning under contexts, and generalization. The soundness of the Join rule is straightforward to obtain, while Theorem 2 justifies the soundness of the Ctx and Gral rules. This result can be used to simplify the complexity of terms to be joinable if they share a common context.

Theorem 2. *Let $\mathcal{R}$ be a rewrite theory with signature $\Sigma = (S, \leq, F)$ and $C[t], C[u] \in T_\Sigma(X)_s$ for some $s \in S$. If $\mathcal{R} \Vdash (\forall X)\, t \downarrow u$, then:*

1. *$\mathcal{R} \Vdash (\forall X)\, C[t] \downarrow C[u]$;*
2. *$\mathcal{R} \Vdash (\forall X)\, t\theta \downarrow u\theta$, for any substitution $\theta \in [X \longrightarrow T_\Sigma(X)]$.*

Proof. The two properties follow from the fact that the rewrite relation $\rightarrow_{\mathcal{R}}$ is closed under contexts and substitutions. □

Since the goal is to prove ground joinability of a rewrite theory of the form $\mathcal{R}_\mathcal{E} = (\Sigma, B, \overrightarrow{E})$ associated to an equational theory $\mathcal{E} = (\Sigma, E \uplus B)$, such as that for hereditarily finite sets presented in Section 3, the most appropriate notion of constructor is that of $\mathcal{R}_\mathcal{E}$-constructors. More precisely, a constructor signature pair for $\mathcal{R}_\mathcal{E}$ has always the form (Υ, Σ) because the only equations in $\mathcal{R}_\mathcal{E}$ are the axioms B not associated to any rewriting. Hence, $\mathcal{R}_\mathcal{E}$ sufficient completeness is *always* relative only to Υ.

Reasoning about ground joinability requires inductive inference support, e.g., in the form of a constructor-based scheme using finite generating sets.

Definition 3. *Let $\mathcal{E} = (\Sigma, E \uplus B)$ be an equational theory, with $\Sigma = (S, \leq, F)$, such that the rewrite theory $\mathcal{R}_\mathcal{E}$ is weakly terminating, ground sort-decreasing, and has subsignature Υ of $\mathcal{R}_\mathcal{E}$-constructors. Further, let $s \in S$. A set $G_s \subseteq T_{\Upsilon,s}(X)$ is a* (finite) generating set for s modulo B *iff G_s is finite, $G_s \cap X = \emptyset$, and*

$$T_{\Upsilon/B,s} = \bigcup_{w \in G_s} \{[w\sigma]_B \mid \sigma \in [vars(w) \longrightarrow T_\Upsilon]\}\,.$$

The following induction scheme is sound for inferring ground joinability in $\mathcal{R}_\mathcal{E}$.

Theorem 3. *Let $\mathcal{R}_\mathcal{E}$ be a weakly terminating and ground sort-decreasing rewrite theory, with signature $\Sigma = (S, \leq, F)$ and subsignature Υ of $\mathcal{R}_\mathcal{E}$-constructors. Moreover, let $t, u \in T_\Sigma(X)$, $x \in vars(t, u) \cap X_s$ for some $s \in S$, and G_s a generating set for s modulo B, such that (without loss of generality) $vars(G_s) \cap vars(t, u) = \emptyset$. Then:*

$$\textit{If } \mathcal{R}_\mathcal{E} \Vdash (\forall X) \bigwedge_{w \in G_s} \left[\bigwedge_{y \in vars(w) \cap X_{<s}} (t \downarrow u)\{x \mapsto y\} \right] \Rightarrow (t \downarrow u)\{x \mapsto w\},$$

then $\mathcal{R}_\mathcal{E} \Vdash (\forall X)\, t \downarrow u$.

Proof. By contradiction. Suppose the antecedent holds, but there is a ground substitution $\sigma \in [vars(t,u) \longrightarrow T_\Sigma]$ such that $\mathcal{R}_\mathcal{E} \nvdash (t \downarrow u)\sigma$. Note, however, that by $\overrightarrow{E}$ being strict B-coherent and G_s being a generating set for s modulo B, σ is always of the form $\sigma =_B \{x \mapsto w\}\tau$, for some $w \in G_s$ and substitution τ, and then we have

$$\mathcal{R}_\mathcal{E} \nvdash (t \downarrow u)\sigma \quad \text{iff} \quad \mathcal{R}_\mathcal{E} \nvdash (t \downarrow u)\{x \mapsto w\}\tau.$$

Consider now the non-empty set of ground terms

$$\{w\tau \mid w \in G_s \ \wedge \ \tau \in [Y_w \longrightarrow T_\Sigma] \ \wedge \ \mathcal{R}_\mathcal{E} \nvdash (t \downarrow u)\{x \mapsto w\}\tau\}$$

where $Y_w = (vars(t,u) \setminus \{x\}) \cup vars(w)$. Pick $w\tau_0$ of smallest term size possible in the above set. By the strict B-coherence of $\overrightarrow{E}$ and the assumption that the axioms B are size-preserving, this means that for any ground substitution $\sigma \in [vars(t,u) \longrightarrow T_\Sigma]$, such that $\mathcal{R}_\mathcal{E} \nvdash (t \downarrow u)\sigma$, we must have $|\sigma(x)| \geq |w\tau_0|$. In particular, since $w \cap X = \emptyset$, this means that for each $y \in vars(w) \cap X_{\leq s}$ we must have $|\tau_0(y)| < |w\tau_0|$ and therefore $\mathcal{R}_\mathcal{E} \vdash (t \downarrow u)\{x \mapsto y\}\tau_0$. But, by hypothesis this implies $\mathcal{R}_\mathcal{E} \vdash (t \downarrow u)\{x \mapsto w\}\tau_0$, a contradiction. □

It is also sound to reason about ground joinability in $\mathcal{R}_\mathcal{E}$ using case analysis based on the $\mathcal{R}_\mathcal{E}$-constructor signature Υ.

Theorem 4. *Let $\mathcal{R}_\mathcal{E}$ be a weakly terminating and ground sort-decreasing rewrite theory, with signature $\Sigma = (S, \leq, F)$ and subsignature Υ of $\mathcal{R}_\mathcal{E}$-constructors. Moreover, let $t, u \in T_\Sigma(X)$, $x \in vars(t,u) \cap X_s$ for some $s \in S$, and G_s a generating set for s modulo B, such that (without loss of generality) $vars(G_s) \cap vars(t,u) = \emptyset$. Then:*

$$\mathcal{R}_\mathcal{E} \Vdash (\forall X)\, t \downarrow u \quad \textit{iff} \quad \mathcal{R}_\mathcal{E} \Vdash (\forall X) \bigwedge_{w \in G_s} (t \downarrow u)\{x \mapsto w\}.$$

Proof. If $\mathcal{R}_\mathcal{E} \Vdash (\forall X)\, t \downarrow u$, then clearly $\mathcal{R}_\mathcal{E} \Vdash (\forall X) \bigwedge_{w \in G_s} (t \downarrow u)\{x \mapsto w\}$. For the proof in the opposite direction, let $\sigma \in [X \longrightarrow T_\Sigma]$ be such that $\mathcal{R}_\mathcal{E} \nvdash (t \downarrow u)\sigma$: the goal is to show that $\mathcal{R}_\mathcal{E} \nVdash (\forall X) \bigwedge_{w \in G_s} (t \downarrow u)\{x \mapsto w\}$, for some $w \in G_s$. Since G_s is a generating set for the sort s and $x \in X_s$, then there is $w \in G_s$ and $\rho \in [X \longrightarrow T_\Sigma]$ such that $\sigma(x) =_B w\rho$. Let $\sigma' = \sigma_{|vars(t,u)\setminus\{x\}} \uplus \rho$ and observe that σ' is well-defined because of the assumption $vars(G_s) \cap vars(t,u) = \emptyset$. Furthermore, observe:

$$\begin{aligned}
(t \downarrow u)\sigma &= (t \downarrow u)\{x \mapsto \sigma(x)\}\sigma_{|vars(t,u)\setminus\{x\}} \\
&=_B (t \downarrow u)\{x \mapsto w\rho\}\sigma_{|vars(t,u)\setminus\{x\}} \\
&= (t \downarrow u)\{x \mapsto w\}(\sigma_{|vars(t,u)\setminus\{x\}} \uplus \rho) \\
&= (t \downarrow u)\{x \mapsto w\}\sigma'.
\end{aligned}$$

Hence, by the strict B-coherence of $\overrightarrow{E}$, we must have $\mathcal{R}_\mathcal{E} \nVdash (\forall X) \bigwedge_{w \in G_s} (t \downarrow u)\{x \mapsto w\}$. □

This concludes the inference system for proving ground joinability. However, an important practical issue remains: how should the checking of $\mathcal{R} \vdash (\forall X)\, t \downarrow u$ used

$$\frac{\mathcal{R}_{\mathcal{E}} \Vdash (\forall X) \bigwedge_{w \in G_s} \left[\bigwedge_{y \in vars(w) \cap X_{\leq s}} (t \downarrow u)\{x \mapsto y\} \right] \Rightarrow (t \downarrow u)\{x \mapsto w\}}{\mathcal{R}_{\mathcal{E}} \Vdash (\forall X)\, t \downarrow u} \text{ GSInd}$$

$$\frac{\mathcal{R}_{\mathcal{E}} \Vdash (\forall X) \bigwedge_{w \in G_s} (t \downarrow u)\{x \mapsto w\}}{\mathcal{R}_{\mathcal{E}} \Vdash (\forall X)\, t \downarrow u} \text{ CtorCases}$$

Fig. 2: Inference rules for proving ground joinability for a rewrite theory $\mathcal{R}_{\mathcal{E}}$ with $\mathcal{R}_{\mathcal{E}}$-constructors Υ by induction relative to the generating set G_s and by constructor-based case analysis on a variable $x \in vars(t,u) \cap X_s$.

in inference rule Join be best *mechanized*? After all, $t \downarrow u$ is a somewhat *complex* relation, involving existential quantification. This issue can be satisfactorily addressed by means of a program transformation $\mathcal{R}_{\mathcal{E}} \mapsto \mathcal{R}_{\mathcal{E}}^{\approx}$ that extends the possibly conditional and operationally terminating rewrite theory $\mathcal{R}_{\mathcal{E}}$, associated to an equational theory $\mathcal{E} = (\Sigma, E \uplus B)$, to a theory $\mathcal{R}_{\mathcal{E}}^{\approx}$ with: (i) a new sort *Prop* with constant `tt` and (ii) a new operator _ ≈ _ with the rule $x \approx x \to tt$, such that

$$\mathcal{R}_{\mathcal{E}} \vdash (\forall X)\, t \downarrow u \quad \text{iff} \quad \mathcal{R}_{\mathcal{E}}^{\approx} \vdash (\forall X)\, t \approx u \to^* tt.$$

Since the right side of the equivalence is a *reachability property* and the transformation $\mathcal{R}_{\mathcal{E}} \mapsto \mathcal{R}_{\mathcal{E}}^{\approx}$ preserves operational termination, the theory $\mathcal{R}_{\mathcal{E}}^{\approx}$ and Maude's *search* command can be used to check that $\mathcal{R} \vdash (\forall X)\, t \downarrow u$. This is used in the Example 1 below, where the binary function symbol `join` implements the operator _ ≈ _. The precise description of the $\mathcal{R}_{\mathcal{E}} \mapsto \mathcal{R}_{\mathcal{E}}^{\approx}$ transformation is given in Appendix D.

Example 1. Recall from Section 3 the only critical pair output by the CRC tool for the HF-SETS-3 specification; the goal is to prove:

$$\text{HF-SETS-3} \Vdash (\forall M : Magma; S, T : Set)\, t(M,S,T) \downarrow u(M,S,T)$$

where

$$\begin{aligned} t(M,S,T) = {} & P(\{M\}) \cup augment(P(\{M\}), S) \cup augment(P(\{M\}), T) \\ & \cup augment(augment(P(\{M\}), S), T) \\ u(M,S,T) = {} & P(\{M\}) \cup augment(P(\{M\}), S) \cup augment(P(\{M\}), T) \\ & \cup augment(augment(P(\{M\}), T), S) \end{aligned}$$

By the Ctx rule it suffices to prove:

$$\begin{aligned} \text{HF-SETS-3} \Vdash {} & (\forall M : Magma; S, T : Set) \\ & augment(augment(P(\{M\}), S), T) \downarrow augment(augment(P(\{M\}), T), S) \end{aligned}$$

Moreover, since $P(\{M\})$ has sort Set, this statement can be proved by considering a stronger property, namely, by using the Gral rule and proving:

$$\text{HF-SETS-3} \Vdash (\forall S, S', T : Set)\, augment(augment(S', S), T) \downarrow augment(augment(S', T), S)$$

This proof obligation can be dealt with by using the CtorCases rule on $S' \in X_{\mathsf{Set}}$ with generating set $G_{\mathsf{Set}} = \{\{\}, \{M\}\}$ and $M \in X_{\mathsf{Magma}}$. This rule application results in the following two proof obligations:

$$\mathsf{HF\text{-}SETS\text{-}3} \Vdash (\forall S, T : Set)\ augment(augment(\{\}, S), T) \downarrow augment(augment(\{\}, T), S)$$

$$\mathsf{HF\text{-}SETS\text{-}3} \Vdash (\forall S, T : Set; M : Magma)$$
$$augment(augment(\{M\}, S), T) \downarrow augment(augment(\{M\}, T), S)$$

The first proof obligation can be discharged by a search command in $\mathcal{R}^{\approx}_{\mathsf{HF\text{-}SETS\text{-}3}}$:

```
search in HF-SETS-3-REACH :
  join(augment(augment({{}}, S), T), augment(augment({{}}, T), S)) =>! tt .
Solution 1 (state 1)
```

The second proof obligation can be handled using the GSInd rule on $M \in X_{\mathsf{Magma}}$ with generating set $G_{\mathsf{Magma}} = \{S', (S', M')\}$, $S' \in X_{\mathsf{Set}}$, and $M' \in X_{\mathsf{Magma}}$:

$$\mathsf{HF\text{-}SETS\text{-}3} \vdash (\forall S, S', T : Set)$$
$$augment(augment(\{S'\}, S), T) \downarrow augment(augment(\{S'\}, T), S)$$

$$\mathsf{HF\text{-}SETS\text{-}3} \vdash (\forall S, S', T : Set; M' : Magma)$$
$$\psi \Rightarrow augment(augment(\{S', M'\}, S), T) \downarrow augment(augment(\{S', M'\}, T), S)$$

where ψ is the formula:

$$augment(augment(\{S'\}, S), T) \downarrow augment(augment(\{S'\}, T), S) \wedge$$
$$augment(augment(\{M'\}, S), T) \downarrow augment(augment(\{M'\}, T), S).$$

For the first one of these two proof obligations, a proof can be found as follows:

```
search in HF-SETS-3-REACH :
  join(augment(augment({S'}, S), T), augment(augment({S'}, T), S)) =>! tt .
Solution 1 (state 14)
```

For the second proof obligation, it suffices to rewrite both terms in the consequent of the implication and use the second conjunct in ψ, together with the Join and Ctx, to join the resulting terms:

```
search in HF-SETS-3-REACH : augment(augment({M',S'}, S), T) =>! X:Set .
Solution 1 (state 6)
X:Set --> {S' U {S,T}} U augment(augment({M'}, S), T)


search in HF-SETS-3-REACH : augment(augment({M',S'}, T), S) =>! X:Set .
Solution 1 (state 6)
X:Set --> {S' U {S,T}} U augment(augment({M'}, T), S)
```

Therefore, all critical pairs of HF-SETS-3 are ground joinable; hence, HF-SETS-3 is ground convergent, as desired.

But is the original specification HF-SETS itself ground convergent? That is, can the extra equations in HF-SETS-3 just be used as *scaffolding* and then be removed as unnecessary? The following result shows that, if the successive addition of oriented equalities leads us to a ground convergent theory and such equalities are ground joinable, then the added equations are indeed unnecessary. The main idea is that, starting from an equational specification $\mathcal{E}_0$, if a sequence of equational theories $\mathcal{E}_0 \subseteq \mathcal{E}_1 \subseteq \cdots \subseteq \mathcal{E}_n$ can be

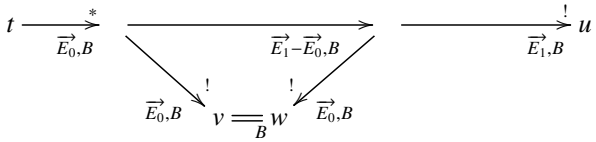
built by incrementally adding new equations (e.g., resulting from critical pairs between the equations), and if the new equations added at each step can be shown ground joinable, then the ground confluence of $\mathcal{E}_n$ implies the ground confluence of each $\mathcal{E}_i$, and in particular of $\mathcal{E}_0$.

Theorem 5. *Let $(\Sigma, E_0 \uplus B) \subseteq (\Sigma, E_1 \uplus B)$ where $\overrightarrow{E_0}, B$ is sufficiently complete with respect to a subsignature Ω, $(\Sigma, E_1 \uplus B)$ is ground convergent, $\rightarrow_{\overrightarrow{E_0},B} |_\Omega = \rightarrow_{\overrightarrow{E_1},B} |_\Omega$, and all equations in $E_1 - E_0$ are ground E_0, B-joinable. Then,*

$$\left(\rightarrow^{!}_{\overrightarrow{E_0},B}; =_B\right)|_{T_\Sigma} = \left(\rightarrow^{!}_{\overrightarrow{E_1},B}; =_B\right)|_{T_\Sigma}.$$

That is, the normal forms of the rewriting relation modulo B restricted to the initial term algebra T_Σ coincide.

Proof. First of all note that, since $\overrightarrow{E_0} \subseteq \overrightarrow{E_1}$, $(\Sigma, B, \overrightarrow{E_0})$ is operationally terminating. Consider some $t \in T_\Sigma$ and rewrite $t \rightarrow^{!}_{\overrightarrow{E_1},B} u$. Since $\overrightarrow{E_0}, B$ is sufficiently complete and $\rightarrow_{\overrightarrow{E_0},B} |_\Omega = \rightarrow_{\overrightarrow{E_1},B} |_\Omega$, $u \in T_\Omega$. If all rules applied in the chain are in $\overrightarrow{E_0}$, then the chains obviously coincide. Otherwise, let us consider the first rewrite step using a rule in $\overrightarrow{E_1} - \overrightarrow{E_0}$:

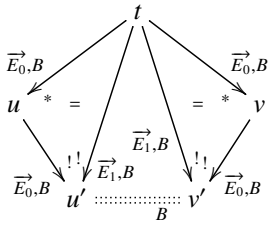


First, we have $v =_B w$ by ground joinability of equations in $E_1 - E_0$. Then, by the assumption that $\rightarrow_{\overrightarrow{E_0},B} |_\Omega = \rightarrow_{\overrightarrow{E_1},B} |_\Omega$, u and w are in E_1, B-canonical form, and by the ground confluence of $\overrightarrow{E_1}, B$ we must have $u =_B w$. Therefore, we can conclude that $\rightarrow^{!}_{\overrightarrow{E_0},B}; =_B |_{T_\Sigma} = \rightarrow^{!}_{\overrightarrow{E_1},B}; =_B |_{T_\Sigma}$, as desired. □

Theorem 6. *Suppose $(\Sigma, E_0 \uplus B) \subseteq \ldots \subseteq (\Sigma, E_n \uplus B)$, with $n \geq 0$, such that $\overrightarrow{E_0} \uplus B$ is sufficiently complete with respect to a subsignature Ω, $\overrightarrow{E_n} \uplus B$ is ground convergent, $(\rightarrow_{\overrightarrow{E_0},B})|_\Omega = (\rightarrow_{\overrightarrow{E_n},B})|_\Omega$, and all $E_{i+1} - E_i$ are ground E_i-joinable modulo B. Then, each $(\Sigma, E_i \uplus B)$ is ground convergent, for $0 \leq i \leq n$. Furthermore, all theories in the chain have the same initial algebra.*

Proof. By induction on n. It is trivial for $n = 0$. Suppose it true for n, and let us prove it true for $n + 1$. Given a chain $(\Sigma, E_0 \uplus B) \subseteq (\Sigma, E_1 \uplus B) \subseteq \ldots \subseteq (\Sigma, E_n \uplus B)$, by the induction hypothesis —plus the fact that $(\Sigma, E_0 \uplus B)$ sufficiently complete makes $(\Sigma, E_1 \uplus B)$ so as well— we get that $(\Sigma, E_1 \uplus B)$ is ground convergent. The proof that $(\Sigma, E_0 \uplus B)$ is ground convergent is as follows. Since $(\Sigma, E_1 \uplus B)$ is ground convergent, $(\Sigma, E_0 \uplus B)$ is *a fortiori* sort-decreasing and operationally terminating, so all we need to

prove is its ground confluence. But since, by Theorem 5, $\rightarrow^{!}_{\overrightarrow{E_0},B}; =_B |_{T_\Sigma} = \rightarrow^{!}_{\overrightarrow{E_1},B}; =_B |_{T_\Sigma}$, the following diagram proves ground confluence of $\rightarrow_{\overrightarrow{E_0},B}$:



Note that $u' =_B v'$ by ground confluence of $\rightarrow_{\overrightarrow{E_1},B}$.

Finally, we already know by the Induction Hypothesis that all the theories

$$(\Sigma, E_1 \uplus B) \subseteq \cdots \subseteq (\Sigma, E_n \uplus B)$$

have the same initial algebra, and, by ground-joinability of $E_1 - E_0$, that

$$\mathcal{T}_{\Sigma/E_0 \uplus B} \models E_1 - E_0.$$

Therefore, we also get $\mathcal{T}_{\Sigma/E_0 \uplus B} = \mathcal{T}_{\Sigma/E_1 \uplus B}$, as desired. □

Theorem 6 justifies the view of the new equations suggested by critical pairs obtained, say, from the CRC tool, as hints for extending our original specification as "scaffolding" that can be abandoned *after* we have reached a ground convergent extension $(\Sigma, E_n \uplus B)$. Going back to the example in Section 3, once the HF-SETS-3 module has been proven ground convergent, we can conclude that the original HF-SETS specification is also ground convergent, *provided* we can show that the equations added at stage $i + 1$ were ground joinable relative to stage i. This is shown to be the case in Section 5 by providing proofs of ground joinability for the five equations added in HF-SETS-0, HF-SETS-1, HF-SETS-2, and HF-SETS-3 in Section 3.

5 Ground Convergence of HF-SETS

The goal of this section is to conclude that the equational specification HF-SETS presented in Section 3 is ground convergent, and therefore that its initial model is a model of set theory without the axiom of infinity. The key tools for achieving this goal are the inference system for inductive joinability and Theorem 6, both presented in Section 4. By knowing that $\mathcal{R}_{\text{HF-SETS-3}}$ is terminating (Appendix B), sort decreasing (Section 3), and that HF-SETS is sufficiently complete (Appendix C), the conditions in Theorem 6 apply and we just need to show the ground joinability of the added equations.

That is, since HF-SETS-3 is ground convergent and the theory inclusions

HF-SETS ⊆ HF-SETS-0 ⊆ HF-SETS-1 ⊆ HF-SETS-2 ⊆ HF-SETS-3

satisfy the requirements of Theorem 6, it suffices to prove

$$\begin{aligned}
&\mathsf{HF\text{-}SETS} \Vdash (\forall M : Magma)\ M \in \{M\} \downarrow true\\
&\mathsf{HF\text{-}SETS} \Vdash (\forall M, M' : Magma)\ M \in \{M, M'\} \downarrow true\\
&\mathsf{HF\text{-}SETS\text{-}0} \Vdash (\forall S : Set)\ S \cup S \downarrow S\\
&\mathsf{HF\text{-}SETS\text{-}1} \Vdash (\forall S, S', T : Set)\ augment(S \cup S', T) \downarrow augment(S, T) \cup augment(S', T)\\
&\mathsf{HF\text{-}SETS\text{-}2} \Vdash (\forall S, T : Set)\ augment(augment(S, T), T) \downarrow augment(S, T)
\end{aligned}$$

in order to conclude that HF-SETS is ground convergent. In what follows, detailed proofs are provided for the last three proof obligations. The first two properties can be proved by following a similar approach.

The third proof obligation is dealt with by using the CtorCases rule on $S \in X_{\mathsf{Set}}$ with generating set $G_{\mathsf{Set}} = \{\{\}, \{M\}\}$ and $M \in X_{\mathsf{Magma}}$:

$$\begin{aligned}
&\mathsf{HF\text{-}SETS\text{-}0} \Vdash \{\} \cup \{\} \downarrow \{\}\\
&\mathsf{HF\text{-}SETS\text{-}0} \Vdash (\forall M : Magma)\ \{M\} \cup \{M\} \downarrow \{M\}
\end{aligned}$$

These two proof obligations can be automatically discharged by Maude in $\mathcal{R}^{\approx}_{\mathsf{HF\text{-}SETS\text{-}0}}$:

```
search in HF-SETS-0-REACH : join({} U {}, {}) =>! tt .
Solution 1 (state 2)

search in HF-SETS-0-REACH : join({M} U {M}, {M}) =>! tt .
Solution 1 (state 3)
```

Next, for the fourth proof obligation, a sequence of inference steps are needed. First, the CtorCases rule is used on S_{Set} with generating set $G_{\mathsf{Set}} = \{\{\}, \{M\}\}$ and $M \in X_{\mathsf{Magma}}$, resulting in the following proof obligations:

$$\begin{aligned}
&\mathsf{HF\text{-}SETS\text{-}1} \Vdash (\forall S', T : Set)\ augment(\{\} \cup S', T) \downarrow augment(\{\}, T) \cup augment(S', T)\\
&\mathsf{HF\text{-}SETS\text{-}1} \Vdash (\forall S', T : Set; M : Magma)\\
&\qquad\qquad augment(\{M\} \cup S', T) \downarrow augment(\{M\}, T) \cup augment(S', T)
\end{aligned}$$

For the second one of these two proof obligations, the CtorCases rule on $S' \in X_{\mathsf{Set}}$ with generating set $H'_{\mathsf{Set}} = \{\{\}, \{M'\}\}$ and $M' \in X_{\mathsf{Magma}}$ is used; this transforms the second proof obligation in the following two proof obligations:

$$\begin{aligned}
&\mathsf{HF\text{-}SETS\text{-}1} \Vdash (\forall T : Set; M : Magma)\\
&\qquad\qquad augment(\{M\} \cup \{\}, T) \downarrow augment(\{M\}, T) \cup augment(\{\}, T)\\
&\mathsf{HF\text{-}SETS\text{-}1} \Vdash (\forall T : Set; M, M' : Magma)\\
&\qquad\qquad augment(\{M\} \cup \{M'\}, T) \downarrow augment(\{M\}, T) \cup augment(\{M'\}, T)
\end{aligned}$$

The remaining three proof obligations can be automatically discharged by Maude in $\mathcal{R}^{\approx}_{\mathsf{HF\text{-}SETS\text{-}1}}$ as follows:

```
search in HF-SETS-1-REACH : join(augment({} U S', T), augment({}, T) U augment(S', T)) =>! tt .
Solution 1 (state 6)
```

```
search in HF-SETS-1-REACH :
  join(augment({} U {M}, T), augment({}, T) U augment({M}, T)) =>! tt .
Solution 1 (state 6)


search in HF-SETS-1-REACH :
  join(augment({M} U {M'}, T), augment({M}, T) U augment({M'}, T)) =>! tt .
Solution 1 (state 3)
```

The fifth, and last proof obligation, is dealt with by using the CtorCases rule on $S \in X_{\mathsf{Set}}$ with generating set $G_{\mathsf{Set}} = \{\{\}, \{M\}\}$ and $M \in X_{\mathsf{Magma}}$. This rule application results in the following two proof obligations:

$$\mathsf{HF\text{-}SETS\text{-}2} \Vdash (\forall T : Set)\ augment(augment(\{\}, T), T) \downarrow augment(\{\}, T)$$

$$\mathsf{HF\text{-}SETS\text{-}2} \Vdash (\forall T : Set; M : Magma)\ augment(augment(\{M\}, T), T) \downarrow augment(\{M\}, T)$$

The first proof obligation can be discharged automatically:

```
search in HF-SETS-2-REACH : join(augment(augment({}, T), T), augment({}, T)) =>! tt .
Solution 1 (state 4)
```

The remaining proof obligation can be handled with the help of the GSInd rule with generating set $G_{\mathsf{Magma}} = \{S', (S', M')\}$, $S' \in X_{\mathsf{Set}}$ and $M' \in X_{\mathsf{Magma}}$:

$$\mathsf{HF\text{-}SETS\text{-}2} \Vdash (\forall S', T : Set)\ augment(augment(\{S'\}, T), T) \downarrow augment(\{S'\}, T)$$

$$\mathsf{HF\text{-}SETS\text{-}2} \Vdash (\forall\ S', T : Set; M' : Magma)\ \psi \Rightarrow augment(augment(\{S', M'\}, T), T) \downarrow augment(\{S', M'\}, T)$$

where ψ is the formula:

$$augment(augment(\{S'\}, T), T) \downarrow augment(\{S'\}, T)$$
$$\wedge\ augment(augment(\{M'\}, T), T) \downarrow augment(\{M'\}, T).$$

These two proof obligations can be solved with the help of Maude:

```
search in HF-SETS-2-REACH : join(augment(augment({S'}, T), T), augment({S'}, T)) =>! tt .
Solution 1 (state 10)


search in HF-SETS-2-REACH : augment(augment({M',S'}, T), T) =>! X:Set .
Solution 1 (state 7)
X:Set --> {S' U {T}} U augment(augment({M'}, T), T)


search in HF-SETS-2-REACH : augment({M',S'}, T) =>! X:Set .
Solution 1 (state 2)
X:Set --> {S' U {T}} U augment({M'}, T)
```

Note that the terms obtained by the last two search commands can be joined by assuming ψ.

The initial goal has now been reached. Namely, since all the equations added in the process of building the tower of theory inclusions

$$\mathsf{HF\text{-}SETS} \subseteq \mathsf{HF\text{-}SETS\text{-}0} \subseteq \mathsf{HF\text{-}SETS\text{-}1} \subseteq \mathsf{HF\text{-}SETS\text{-}2} \subseteq \mathsf{HF\text{-}SETS\text{-}3}$$

have been shown ground joinable, Theorem 6 guarantees that the equational specification HF-SETS for hereditarily finite sets is ground convergent.

6 Related Work and Conclusion

In [2], A. Bouhoula proposes an inference system for simultaneously checking the sufficient completeness and ground confluence of constructor-based equational specifications. His approach computes a pattern tree for every defined symbol and identifies a set of proof obligations whose inductive validity has to be checked: it they all are inductive theorems, then the specification is both sufficiently complete and ground confluent; otherwise, it outputs a counterexample. The main difference between the two approaches is that the one presented in this paper can handle both conditional specifications and reasoning modulo axioms, while [2] does not support reasoning modulo axioms. More recently, Nakamura et al. [15] propose incremental techniques for proving termination, confluence, and sufficient completeness of OBJ specifications. Their inference system is also based on the notion of constructor subsignatures, handles conditional equations, and provides sufficient conditions for ensuring such an incremental extension in a modular way. However, for ground confluence, their method has been developed for extensions that preserve the set of critical pairs relative to the extended specification.

Different tools and techniques have been proposed for proving and disproving confluence. Tools such as CSI [14] or ACP [1] are automatic confluence provers for first-order rewrite systems. These tools implement different criteria for proving both confluence and non-confluence.

This work has addressed a thorny and important problem in reasoning about equational programs and algebraic specifications with an initial algebra semantics: the fact that in practice a substantial number of such programs and specifications are *perfectly reasonable* and there is nothing wrong with them, yet they are not locally confluent and therefore fall outside the scope of the standard methods to prove them ground convergent. As the `HF-SETS` example has shown, it is quite mistaken to assume that, since our program is perfectly reasonable, we should be able to complete it in some Knuth-Bendix-like fashion, because we can easily hit a non-orientability "wall." We have proposed a general methodology to help verify the ground convergence of a given equational program in such a way that: (i) the *heuristic* value of using unjoinable critical pairs as *hints* is preserved; (ii) we can break through the wall of non-orientable equations by means of *inductive joinability* proof methods; and (iii) we can prove that our *original* specification is ground convergent and that its initial algebra semantics has been preserved by its subsequent extensions using the same inductive joinability proof techniques.

Future work suggested by this work includes: (i) full mechanization of the inductive joinability inference system and its integration within the Maude Formal Environment; (ii) further experimentation with these methods on a rich collection of examples; and (iii) development of new proof techniques complementing those presented here.

References

1. T. Aoto, J. Yoshida, and Y. Toyama. Proving confluence of term rewriting systems automatically. In R. Treinen, editor, *Proc. RTA*, volume 5595 of *Lecture Notes in Computer Science*, pages 93–102. Springer, 2009.

2. A. Bouhoula. Simultaneous checking of completeness and ground confluence for algebraic specifications. *ACM Transactions on Computational Logic*, 10(3):1–33, Apr. 2009.
3. R. Bruni and J. Meseguer. Semantic foundations for generalized rewrite theories. *Theoretical Computer Science*, 360(1-3):386–414, Aug. 2006.
4. M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. L. Talcott. *All About Maude - A High-Performance Logical Framework*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007.
5. N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Vol. B*, pages 243–320. North-Holland, 1990.
6. F. Durán, S. Lucas, C. Marché, J. Meseguer, and X. Urbain. Proving operational termination of membership equational programs. *Higher-Order and Symbolic Computation*, 21(1-2):59–88, June 2008.
7. F. Durán, S. Lucas, and J. Meseguer. Termination modulo combinations of equational theories. In *Frontiers of Combining Systems, 7th International Symposium, FroCoS 2009, Trento, Italy, September 16-18, 2009. Proceedings*, volume 5749 of *Lecture Notes in Computer Science*, pages 246–262. Springer, 2009.
8. F. Durán and J. Meseguer. On the Church-Rosser and coherence properties of conditional order-sorted rewrite theories. *J. Log. Algebr. Program.*, 81(7-8):816–850, 2012.
9. F. Durán, C. Rocha, and J. M. Álvarez. Towards a Maude formal environment. In G. Agha, O. Danvy, and J. Meseguer, editors, *Formal Modeling: Actors, Open Systems, Biological Systems*, volume 7000 of *Lecture Notes in Computer Science*, pages 329–351. Springer, 2011.
10. J. A. Goguen and J. Meseguer. Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations. *Theoretical Computer Science*, 105(2):217–273, 1992.
11. K. Hrbacek and T. J. Jech. *Introduction to Set Theory*. Number 220 in Monographs and textbooks in pure and applied mathematics. M. Dekker, New York, 3rd ed., rev. and expanded edition, 1999.
12. J. Meseguer. Membership algebra as a logical framework for equational specification. In G. Goos, J. Hartmanis, J. Leeuwen, and F. P. Presicce, editors, *Recent Trends in Algebraic Development Techniques*, volume 1376, pages 18–61. Springer, Berlin, Heidelberg, 1998.
13. J. Meseguer. Strict coherence of conditional rewriting modulo axioms. *Theor. Comput. Sci.*, 672:1–35, 2017.
14. J. Nagele, B. Felgenhauer, and A. Middeldorp. CSI: new evidence - A progress report. In L. de Moura, editor, *Proc. CADE*, volume 10395 of *Lecture Notes in Computer Science*, pages 385–397. Springer, 2017.
15. M. Nakamura, K. Ogata, and K. Futatsugi. Incremental Proofs of Termination, Confluence and Sufficient Completeness of OBJ Specifications. In S. Iida, J. Meseguer, and K. Ogata, editors, *Specification, Algebra, and Software*, volume 8373, pages 92–109. Springer, Berlin, Heidelberg, 2014.
16. C. Rocha and J. Meseguer. Constructors, Sufficient Completeness, and Deadlock Freedom of Rewrite Theories. In D. Hutchison and others, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 6397, pages 594–609. Springer, 2010.

A Church-Rosser Check of the HF-SETS-3 and its Submodules

```
                     \||||||||||||||||||/
                   --- Welcome to Maude ---
                     /||||||||||||||||||\
            Maude-mfe-hooks 2.7 built: Aug  1 2014 18:22:26
             With termination checker extension
            Copyright 1997-2014 SRI International
                   Mon Jan 22 14:03:21 2018

            Full Maude 2.7 March 10th 2015

The Maude Formal Environment 1.0c
    Inductive Theorem Prover - July 20th 2010
    Sufficient Completeness Checker 2a - August 2010
    Church-Rosser Checker 3n - December 17th 2012
    Coherence Checker 3n - December 17th 2012
    Maude Termination Tool 1.5j - August 11th 2014

set include BOOL off

set include TRUTH-VALUE on

Maude> (set include BOOL off .)
rewrites: 33 in 0ms cpu (0ms real) (~ rewrites/second)
set include BOOL off

Maude> (select tool CRC .)
rewrites: 76 in 1ms cpu (1ms real) (68345 rewrites/second)
The CRC has been set as current tool.

Maude> (ccr HF-SETS .)
rewrites: 14530655 in 17432ms cpu (20000ms real) (833542 rewrites/second)
Church-Rosser check for HF-SETS
The following critical pairs must be proved joinable:
  cp HF-SETS1123 for 11 and 15
    true
    = M':Magma in{M':Magma}.
  cp HF-SETS119 for 01 and 63
    augment({M':Magma},S:Set)U augment({#5:Magma},S:Set)
    = augment({M':Magma},S:Set)U augment({M':Magma},S:Set)U augment({#5:Magma},
    S:Set).
  cp HF-SETS123 for 01 and 63
    augment({#4:Magma},S:Set)U augment({#5:Magma},S:Set)U augment({#6:Magma},
    S:Set)
    = augment({#4:Magma},S:Set)U augment({#4:Magma},S:Set)U augment({#5:Magma},
    S:Set)U augment({#6:Magma},S:Set).
  cp HF-SETS131 for 01 and 63
    augment({#4:Magma},S:Set)U augment({#5:Magma},S:Set)U augment({#6:Magma},
    S:Set)U augment({#7:Magma},S:Set)
    = augment({#4:Magma},S:Set)U augment({#4:Magma},S:Set)U augment({#5:Magma},
    S:Set)U augment({#6:Magma},S:Set)U augment({#7:Magma},S:Set).
  cp HF-SETS136 for 01 and 63
    augment({#4:Magma},S:Set)U augment({#5:Magma},S:Set)U augment({#6:Magma},
    S:Set)U augment({#7:Magma},S:Set)U augment({#8:Magma},S:Set)
    = augment({#4:Magma},S:Set)U augment({#5:Magma},S:Set)U augment({#5:Magma},
    S:Set)U augment({#6:Magma},S:Set)U augment({#7:Magma},S:Set)U augment({
    #8:Magma},S:Set).
  cp HF-SETS5 for 53 and 53
    P({#6:Magma})U augment(P({#6:Magma}),S:Set)U augment(P({#6:Magma})U
    augment(P({#6:Magma}),S:Set),#1:Set)
    = P({#6:Magma})U augment(P({#6:Magma}),#1:Set)U augment(P({#6:Magma})U
    augment(P({#6:Magma}),#1:Set),S:Set).
  cp HF-SETS80 for 01 and 53
    P({#3:Magma})U augment(P({#3:Magma}),S:Set)
    = P({#3:Magma})U augment(P({#3:Magma}),S:Set)U augment(P({#3:Magma})U
    augment(P({#3:Magma}),S:Set),S:Set).
```

```
  cp HF-SETS88 for 01 and 63
    augment({M':Magma},S:Set)
    = augment({M':Magma},S:Set)U augment({M':Magma},S:Set).
        The module is sort-decreasing.

Maude> (ccr HF-SETS-0 .)
rewrites: 14944931 in 19319ms cpu (22399ms real) (773567 rewrites/second)
Church-Rosser check for HF-SETS-0
The following critical pairs must be proved joinable:
  cp HF-SETS-0105 for 01 and 53
    P({#5:Magma})U augment(P({#5:Magma}),S:Set)
    = P({#5:Magma})U augment(P({#5:Magma}),S:Set)U augment(P({#5:Magma})U
    augment(P({#5:Magma}),S:Set),S:Set).
  cp HF-SETS-0145 for 01 and 63
    augment({M':Magma},S:Set)U augment({#5:Magma},S:Set)
    = augment({M':Magma},S:Set)U augment({M':Magma},S:Set)U augment({#5:Magma},
    S:Set).
  cp HF-SETS-0149 for 01 and 63
    augment({#4:Magma},S:Set)U augment({#5:Magma},S:Set)U augment({#6:Magma},
    S:Set)
    = augment({#4:Magma},S:Set)U augment({#4:Magma},S:Set)U augment({#5:Magma},
    S:Set)U augment({#6:Magma},S:Set).
  cp HF-SETS-0157 for 01 and 63
    augment({#4:Magma},S:Set)U augment({#5:Magma},S:Set)U augment({#6:Magma},
    S:Set)U augment({#7:Magma},S:Set)
    = augment({#4:Magma},S:Set)U augment({#4:Magma},S:Set)U augment({#5:Magma},
    S:Set)U augment({#6:Magma},S:Set)U augment({#7:Magma},S:Set).
  cp HF-SETS-0162 for 01 and 63
    augment({#4:Magma},S:Set)U augment({#5:Magma},S:Set)U augment({#6:Magma},
    S:Set)U augment({#7:Magma},S:Set)U augment({#8:Magma},S:Set)
    = augment({#4:Magma},S:Set)U augment({#5:Magma},S:Set)U augment({#5:Magma},
    S:Set)U augment({#6:Magma},S:Set)U augment({#7:Magma},S:Set)U augment({
    #8:Magma},S:Set).
  cp HF-SETS-07 for 53 and 53
    P({#6:Magma})U augment(P({#6:Magma}),S:Set)U augment(P({#6:Magma})U
    augment(P({#6:Magma}),S:Set),#1:Set)
    = P({#6:Magma})U augment(P({#6:Magma}),#1:Set)U augment(P({#6:Magma})U
    augment(P({#6:Magma}),#1:Set),S:Set).
  cp HF-SETS-095 for 01 and 63
    augment({M':Magma},S:Set)
    = augment({M':Magma},S:Set)U augment({M':Magma},S:Set).
        The module is sort-decreasing.

Maude> (ccr HF-SETS-1 .)
rewrites: 16134529 in 20665ms cpu (23236ms real) (780741 rewrites/second)
Church-Rosser check for HF-SETS-1
The following critical pairs must be proved joinable:
  cp HF-SETS-1299 for 01 and 53
    P({#3:Magma})U augment(P({#3:Magma}),S:Set)
    = P({#3:Magma})U augment(P({#3:Magma}),S:Set)U augment(P({#3:Magma})U
    augment(P({#3:Magma}),S:Set),S:Set).
  cp HF-SETS-16 for 53 and 53
    P({#6:Magma})U augment(P({#6:Magma}),S:Set)U augment(P({#6:Magma})U
    augment(P({#6:Magma}),S:Set),#1:Set)
    = P({#6:Magma})U augment(P({#6:Magma}),#1:Set)U augment(P({#6:Magma})U
    augment(P({#6:Magma}),#1:Set),S:Set).
        The module is sort-decreasing.

Maude> (ccr HF-SETS-2 .)
rewrites: 16759193 in 21934ms cpu (24382ms real) (764044 rewrites/second)
Church-Rosser check for HF-SETS-2
The following critical pairs must be proved joinable:
  cp HF-SETS-2355 for 01 and 53
    P({#3:Magma})U augment(P({#3:Magma}),S:Set)
    = P({#3:Magma})U augment(P({#3:Magma}),S:Set)U augment(augment(P({
    #3:Magma}),S:Set),S:Set).
  cp HF-SETS-26 for 53 and 53
    P({#6:Magma})U augment(P({#6:Magma}),S:Set)U augment(P({#6:Magma}),#1:Set)U
```

```
    augment(augment(P({#6:Magma}),S:Set),#1:Set)
    = P({#6:Magma})U augment(P({#6:Magma}),S:Set)U augment(P({#6:Magma}),
    #1:Set)U augment(augment(P({#6:Magma}),#1:Set),S:Set).
        The module is sort-decreasing.

Maude> (ccr HF-SETS-3 .)
rewrites: 17061281 in 23124ms cpu (25660ms real) (737786 rewrites/second)
Church-Rosser check for HF-SETS-3
The following critical pairs must be proved joinable:
  cp HF-SETS-36 for 53 and 53
    P({#6:Magma})U augment(P({#6:Magma}),S:Set)U augment(P({#6:Magma}),#1:Set)U
    augment(augment(P({#6:Magma}),S:Set),#1:Set)
    = P({#6:Magma})U augment(P({#6:Magma}),S:Set)U augment(P({#6:Magma}),
    #1:Set)U augment(augment(P({#6:Magma}),#1:Set),S:Set).
        The module is sort-decreasing.
```

B Termination of the HF-SETS-3 and its Submodules

```
                     \||||||||||||||||||/
                   --- Welcome to Maude ---
                     /||||||||||||||||||\
           Maude-mfe-hooks 2.7 built: Aug  1 2014 18:22:26
            With termination checker extension
           Copyright 1997-2014 SRI International
                  Thu Jan 18 18:55:33 2018

           Full Maude 2.7 March 10th 2015

The Maude Formal Environment 1.0c
    Inductive Theorem Prover - July 20th 2010
    Sufficient Completeness Checker 2a - August 2010
    Church-Rosser Checker 3n - December 17th 2012
    Coherence Checker 3n - December 17th 2012
    Maude Termination Tool 1.5j - August 11th 2014

Maude> (set include BOOL off .)
rewrites: 33 in 0ms cpu (0ms real) (70063 rewrites/second)
set include BOOL off

Maude> (select tool MTT .)
rewrites: 76 in 30ms cpu (30ms real) (2509 rewrites/second)
The MTT has been set as current tool.

Maude> (select external tool aprove .)
rewrites: 39 in 36ms cpu (39ms real) (1065 rewrites/second)
aprove is now the current external tool.

Maude> (ct HF-SETS .)
rewrites: 246282 in 10627ms cpu (26371ms real) (23174 rewrites/second)
Success: The module HF-SETS is terminating.

Maude> (ct HF-SETS-0 .)
rewrites: 259780 in 9831ms cpu (24190ms real) (26422 rewrites/second)
Success: The module HF-SETS-0 is terminating.

Maude> (ct HF-SETS-1 .)
rewrites: 272724 in 11478ms cpu (26165ms real) (23759 rewrites/second)
Success: The module HF-SETS-1 is terminating.

Maude> (ct HF-SETS-2 .)
rewrites: 290161 in 14010ms cpu (28415ms real) (20710 rewrites/second)
Success: The module HF-SETS-2 is terminating.

Maude> (ct HF-SETS-3 .)
rewrites: 305282 in 17145ms cpu (32698ms real) (17805 rewrites/second)
Success: The module HF-SETS-3 is terminating.
```

C Sufficient Completeness of the HF-SETS Module

Let $\mathcal{R}_{\mathcal{E}} = (\Sigma, B, \overrightarrow{E})$ be obtained from $\mathcal{E} = (\Sigma, E \cup B)$, where Σ is B-preregular and the axioms B are sort-preserving. Let $f\colon s_1 \dots s_n \to s$ in Σ be such that the equations in E defining f, namely $\{f(\overrightarrow{u_i}) = v_i\}_{i=1\dots n}$ are all such that the $f(\overrightarrow{u_i}) = v_i$ are substitution instances of $f(x_1 : s_1, \dots, x_n : s_n)$. And let $\Omega \subseteq \Sigma$ be the subsignature that we claim is a constructor subsignature for the equations $\overrightarrow{E}$ modulo B. Then f will be *sufficiently complete* relative to Ω if and only if the following set containment holds:

$$[\![f(x_1 : s_1, \dots, x_n : s_n)]\!]_B^\Omega \subseteq [\![f(\overrightarrow{u_1}), \dots, f(\overrightarrow{u_n})]\!]_B^\Omega$$

where, by definition, for $t \in T_\Sigma(X)$ with $Y = vars(t)$ and $s = ls(t)$

$$[\![t]\!]_B^\Omega = \{u \in T_{\Sigma,s} \mid u =_B t\sigma \wedge \sigma \in [Y \to T_\Omega]\}$$

and where

$$[\![t_1, \dots, t_n]\!]_B^\Omega = [\![t_1]\!]_B^\Omega \cup \dots \cup [\![t_n]\!]_B^\Omega$$

Note that $[\![t]\!]_B^\Omega = [\![t\delta]\!]_B^\Omega$ for any δ that is a sort-preserving bijection of variables, so that it is immaterial in $[\![t_1, \dots, t_n]\!]_B^\Omega$ whether the t_i and t_j have disjoint variables or share any variables.

Finally, let $u \neq v$ be an inequality such that $vars(u \neq v) \subseteq vars(t)$. Then, we define

$$[\![t \mid u \neq v]\!] = \{w \in T_{\Sigma,s} \mid w =_B t\sigma \wedge \sigma \in [Y \to T_\Omega] \wedge u\sigma \neq_B v\sigma\}$$

In what follows, we show that the cover sets

$$[\![M]\!]_B^\Omega = [\![S, (S, M)]\!]_B^\Omega \tag{1}$$

$$[\![S]\!]_B^\Omega = [\![\{\}, \{M\}]\!]_B^\Omega \tag{2}$$

are sufficiently complete for each of the defined operators in the HF-SETS module. We use the SCC tool by J. Hendrix to prove sufficient completeness of the HF-SETS module without operators _in_ and _&_. The tool can only handle left-linear and conditional equations, and therefore it cannot handle the specification with these operations. We proceed modularly. First we prove the specification without these operators sufficiently complete using the SCC tool, and then prove sufficient completeness of the definitions of these operators in the following subsections. In the rest of this section we assume S, S', S'' variables of sort *Set* and M, M', M'' variables of sort *Magma*.

C.1 Sufficient completeness of the definition of *_in_*

Let us show sufficient completeness of the definition of *_in_* in the HF-SETS module. To show sufficient completeness of *_in_* we need to show

$$[\![M \mathit{\ in\ } S]\!]_B^\Omega \subseteq [\![S \mathit{\ in\ } \{S\},\ S \mathit{\ in\ } \{\},\ \{\} \mathit{\ in\ } \{\{M\}\},\ \{M\} \mathit{\ in\ } \{\{M'\}\},\ S \mathit{\ in\ } \{S', M\},\ (S, M) \mathit{\ in\ } S',\ \{M\} \mathit{\ in\ } \{\{\}\}]\!]_B^\Omega \tag{3}$$

where Ω are the constructors declared for `HF-SETS` and B its axioms.

By applying (1) to M on the left of (3), this is equivalent to

$$[\![S\ in\ S',\ \underline{(S, M)\ in\ S'}]\!]^{\Omega}_{B} \subseteq [\![S\ in\ \{S\},\ S\ in\ \{\},\ \{\}\ in\ \{\{M\}\},\ \{M\}\ in\ \{\{M'\}\},\ S\ in\ \{S', M\},\ \underline{(S, M)\ in\ S'},\ \{M\}\ in\ \{\{\}\}]\!]^{\Omega}_{B} \tag{4}$$

Since the underlined patterns define the *same* sets, (4) will hold if we can show

$$[\![S\ in\ S']\!]^{\Omega}_{B} \subseteq [\![S\ in\ \{S\},\ S\ in\ \{\},\ \{\}\ in\ \{\{M\}\},\ \{M\}\ in\ \{\{M'\}\},\ S\ in\ \{S', M\},\ \{M\}\ in\ \{\{\}\}]\!]^{\Omega}_{B} \tag{5}$$

By (2) applied on the left to S' this is equivalent to showing

$$[\![\underline{S\ in\ \{\}},\ S\ in\ \{M\}]\!]^{\Omega}_{B} \subseteq [\![S\ in\ \{S\},\ \underline{S\ in\ \{\}},\ \{\}\ in\ \{\{M\}\},\ \{M\}\ in\ \{\{M'\}\},\ S\ in\ \{S', M\},\ \{M\}\ in\ \{\{\}\}]\!]^{\Omega}_{B} \tag{6}$$

Since the underlined patterns define the *same* sets, (6) will hold if we can show

$$[\![S\ in\ \{M\}]\!]^{\Omega}_{B} \subseteq [\![S\ in\ \{S\},\ \{\}\ in\ \{\{M\}\},\ \{M\}\ in\ \{\{M'\}\},\ S\ in\ \{S', M\},\ \{M\}\ in\ \{\{\}\}]\!]^{\Omega}_{B} \tag{7}$$

By (1) applied on the left to M this is equivalent to showing

$$[\![S\ in\ \{S'\},\ \underline{S\ in\ \{S', M\}}]\!]^{\Omega}_{B} \subseteq [\![S\ in\ \{S\},\ \{\}\ in\ \{\{M\}\},\ \{M\}\ in\ \{\{M'\}\},\ \underline{S\ in\ \{S', M\}},\ \{M\}\ in\ \{\{\}\}]\!]^{\Omega}_{B} \tag{8}$$

By observing the corresponding underlined pattern in both sides it is enough to show

$$[\![S\ in\ \{S'\}]\!]^{\Omega}_{B} \subseteq [\![S\ in\ \{S\}, \{\}\ in\ \{\{M\}\}, \{M\}\ in\ \{\{M'\}\}, \{M\}\ in\ \{\{\}\}]\!]^{\Omega}_{B} \tag{9}$$

Since

$$[\![S\ in\ \{S'\}]\!]^{\Omega}_{B} = [\![S\ in\ \{S\}]\!]^{\Omega}_{B} \cup [\![S\ in\ \{S'\} \mid S \neq S']\!]^{\Omega}_{B} \tag{10}$$

it is enough to show

$$[\![S\ in\ \{S'\} \mid S \neq S']\!]^{\Omega}_{B} \subseteq [\![\{\}\ in\ \{\{M\}\}, \{M\}\ in\ \{\{M'\}\}, \{M\}\ in\ \{\{\}\}]\!]^{\Omega}_{B} \tag{11}$$

By (2) applied to S on the left this is equivalent to showing

$$[\![\{\}\ in\ \{S\} \mid \{\} \neq S]\!]^{\Omega}_{B} \cup [\![\{M\}\ in\ \{S\} \mid \{M\} \neq S]\!]^{\Omega}_{B} \subseteq [\![\{\}\ in\ \{\{M\}\}, \{M\}\ in\ \{\{M'\}\}, \{M\}\ in\ \{\{\}\}]\!]^{\Omega}_{B} \tag{12}$$

Let us show

$$[\![\{\}\ in\ \{S\} \mid \{\} \neq S]\!]^{\Omega}_{B} \subseteq [\![\{\}\ in\ \{\{M\}\}, \{M\}\ in\ \{\{M'\}\}, \{M\}\ in\ \{\{\}\}]\!]^{\Omega}_{B} \tag{13}$$

and

$$[\![\{M\}\ in\ \{S\} \mid \{M\} \neq S]\!]_B^\Omega \subseteq [\![\{\}\ in\ \{\{M\}\}, \{M\}\ in\ \{\{M'\}\}, \{M\}\ in\ \{\{\}\}]\!]_B^\Omega \tag{14}$$

By applying (2) to S on the left of (13) and $\{\} = \{\}$ we get

$$[\![\underline{\{\}\ in\ \{\{M\}\}}]\!]_B^\Omega \subseteq [\![\underline{\{\}\ in\ \{\{M\}\}}, \{M\}\ in\ \{\{M'\}\}, \{M\}\ in\ \{\{\}\}]\!]_B^\Omega \tag{15}$$

which always holds. Finally, by applying (2) to S on the left of (14) and $\{M\} \neq \{\}$ we get

$$[\![\{M\}\ in\ \{\{\}\}]\!]_B^\Omega \cup [\![\{M\}\ in\ \{\{M'\}\}\,|_{\{M\}\neq\{M'\}}]\!]_B^\Omega \subseteq [\![\{M\}\ in\ \{\{M'\}\}, \{M\}\ in\ \{\{\}\}]\!]_B^\Omega \tag{16}$$

Since $[\![\{M\}\ in\ \{\{\}\}]\!]_B^\Omega \subseteq [\![\{M\}\ in\ \{\{\}\}]\!]_B^\Omega$ and $[\![\{M\}\ in\ \{\{M'\}\}\,|_{\{M\}\neq\{M'\}}]\!]_B^\Omega \subseteq [\![\{M\}\ in\ \{\{M'\}\}]\!]_B^\Omega$ we can conclude that the definition of the *_in_* operation is sufficiently complete.

C.2 Sufficient completeness of the definition of _&_

Since *true* and *false* are irreducible constructor terms, we can use the transformation in [8] of the equations defining the operator _&_ as:

$$\{\}\ \&\ S' \to \{\}$$
$$\{S\}\ \&\ S' \to \{S\}\ if\ S\ in\ S' \to true$$
$$\{S\}\ \&\ S' \to \{\}\ if\ S\ in\ S' \to false$$
$$\{S, M\}\ \&\ S' \to \{S\} \cup (\{M\}\ \&\ S')\ if\ S\ in\ S' \to true$$
$$\{S, M\}\ \&\ S' \to \{M\}\ \&\ S'\ if\ S\ in\ S' \to false$$

Let us define $[\![\ t \mid x\ in\ y \to^* true]\!]_B^\Omega$ as the set

$$[\![\ t \mid x\ in\ y \to^* true]\!]_B^\Omega = \{\ w \in T_{\Omega,s} \mid w =_B t\sigma \wedge \sigma \in [Z \to T_\Omega] \wedge \sigma(x)\ in\ \sigma(y) \to^*_{\overrightarrow{E^{\equiv}},B} true\ \}$$

where $s = ls(t)$, $vars(t) = Z$, and $x, y \in Z$ have sort *Set*. And let us define $[\![\ t \mid x\ in\ y \to^* false]\!]_B^\Omega$ likewise.

By (2), we have

$$[\![S\ \&\ S']\!]_B^\Omega = [\![\{\}\ \&\ S\ ]\!]_B^\Omega \cup [\![\{S\}\ \&\ S']\!]_B^\Omega \cup [\![\{S, M\}\ \&\ S']\!]_B^\Omega \tag{17}$$

By $\_\ in\ \_$ being sufficiently complete, for any $u, v \in T_{\Omega,Set}$ we have either $u\ in\ v \to^* true$ or $u\ in\ v \to^* false$. Then,

$$[\![t]\!]_B^\Omega = [\![\ t \mid x\ in\ y \to^* true]\!]_B^\Omega \cup [\![\ t \mid x\ in\ y \to^* false]\!]_B^\Omega$$

And therefore, we can write 17 as

$$\begin{aligned}[\![S \,\&\, S']\!]^{\Omega}_{B} = {} & [\![\{\} \,\&\, S\,]\!]^{\Omega}_{B} \\ & \cup [\![\{S\} \,\&\, S' \mid x \,in\, y \to^{*} true]\!]^{\Omega}_{B} \\ & \cup [\![\{S\} \,\&\, S' \mid x \,in\, y \to^{*} false]\!]^{\Omega}_{B} \\ & \cup [\![\{S, M\} \,\&\, S' \mid x \,in\, y \to^{*} true]\!]^{\Omega}_{B} \\ & \cup [\![\{S, M\} \,\&\, S' \mid x \,in\, y \to^{*} false]\!]^{\Omega}_{B}\end{aligned}$$

This shows sufficient completeness of _ & _, since these are exactly the sets of ground terms for which each of the rules for _ & _ are enabled.

D Checking $\mathcal{R}_{\mathcal{E}} \vdash (\forall X)\, t \downarrow u$

Let $\mathcal{R}_{\mathcal{E}} = (\Sigma, B, \overrightarrow{E})$ with $\Sigma = (S, \leq, F)$ be the rewrite theory obtained from $\mathcal{E} = (\Sigma, E \uplus B)$, and let $\mathcal{R}^{\approx}_{\mathcal{E}} = (\Sigma^{\approx}, B, \overrightarrow{E}^{\approx})$ extend $\mathcal{R}_{\mathcal{E}}$ by:

1. extending $(S, \leq)$ to $(S^{\approx}, \leq^{\approx})$ by adding to each connected component $[s] \in S/\equiv_{\leq}$ a top sort $[s]$ with $s' \leq [s]$ for each $s' \in [s]$;
2. adding a fresh new sort *Prop* with constant *tt*;
3. adding for each $[s] \in S/\equiv_{\leq}$ an operator

$$\_ \approx \_ : [s]\ [s] \longrightarrow Prop$$

4. adding to $\overrightarrow{E}$ the rules

$$\{x : [s] \approx x : [s] \to tt \mid [s] \in S/\equiv_{\leq}\}.$$

Lemma 1. *For any $t, u \in T_{\Sigma}(X)$ with $[ls(t)] = [ls(u)]$:*

$$\mathcal{R}_{\mathcal{E}} \vdash (\forall X)\, t \downarrow u \quad \textit{iff} \quad \mathcal{R}^{\approx}_{\mathcal{E}} \vdash (\forall X)\, (t \approx u) \to^{*} tt.$$

For $\mathcal{R}_{\mathcal{E}}$ operationally terminating, $\mathcal{R}_{\mathcal{E}} \vdash (\forall X)\, t \downarrow u$ can be effectively checked in Maude by executing in the system module *mod* $\mathcal{R}^{\approx}_{\mathcal{E}}$ *endm* the search command:

$$search\ t \approx u \Rightarrow!\ tt\ .$$

giving us a decision procedure for deciding $\mathcal{R}_{\mathcal{E}} \vdash (\forall X)\, t \downarrow u$.

Note that the above result applies not just for $\mathcal{E} = (\Sigma, E \uplus B)$ an *unconditional* theory, but also for $\mathcal{E}$ *conditional* and satisfying the requirements in [8], namely, when $\mathcal{R}_{\mathcal{E}}$ is:

1. strongly deterministic;
2. strictly coherent modulo B; and
3. operationally terminating.

Therefore, reasoning about joinability in $\mathcal{R}_{\mathcal{E}}$ can be done under conditions (1)–(3) also for conditional theories and have the equivalence

$$\mathcal{R}_{\mathcal{E}} \vdash (\forall X)\, t \downarrow u \quad \text{iff} \quad \mathcal{R}^{\approx}_{\mathcal{E}} \vdash (\forall X)\, (t \approx u) \to^{*} tt.$$

and the implementation in Maude by *search* applying as well to conditional theories satisfying (1)–(3). In particular, this applies to the checking of joinability for the conditional theories of hereditarily finite sets in Section 4, which have some conditional equations for set intersection.