Congruence closure with ACI function symbols

Tanji Hu and Robert Givan

Purdue University ECE
{hut,givan}@purdue.edu

Abstract. Congruence closure is the following well known reasoning problem: given a premise set of equations between ground terms over uninterpreted function symbols, does a given query equation follow using the axioms of equality? Several methods have been provided for polynomial-time answers to this question. Here we consider this same setting, but where some of the function symbols are known to be associative, commutative, and idempotent (ACI). Given these additional axioms, does the query equation follow from the premise equations? We provide a sound and complete cubic-time procedure correctly answering such questions. The problem requires exponential space when adding only AC function symbols [18], but requiring idempotence restores tractability . Our procedure is defined by providing a sound and complete "local" rule set for the problem [11]. A "local formula" is a formula mentioning only terms appearing in the premises or query. A local rule set is one for which any derivable local formula has a derivation using only local intermediate formulas. Closures under local rule sets can immediately be constructed in polynomial time by refusing to infer non-local formulas. Finally, we present results on the integration of ACI function symbols and equality inference rules into more general local rule sets.

1 Introduction

Congruence closure is a well studied algorithmic problem: given a set of ground equations over a first-order term language of uninterpreted function symbols, which other ground equations between terms are entailed? Previous study has provided a small variety of approaches to efficiently solving this problem [23, 21, 16] as well as a great deal of work on related algorithms [2, 4, 3, 15]. The importance of this problem is apparent in the foundational nature of equality in almost every deductive setting, in systems representing expressive knowledge for almost any purpose. Keynote examples have been deductive support for advanced compilers and other program analysis tools (e.g., [9]), and reasoning systems for representing mathematics (e.g., [8, 19, 5]).

A range of prior work has been conducted on congruence closure in the presence of additional axioms about some or all of the function symbols that are present, resulting in the "uniform word problem" in a variety of algebraic structures [22, 13, 6, 7]. Much of this work does not allow arbitrary uninterpreted function symbols in addition to those affected by the axioms considered (typically commutativity and/or associativity). Notable among the prior related work

is [18] showing that the uniform word problem for commutative semigroups requires exponential space. This result implies that other methods discussing associative-commutative congruence closure that can solve this word problem, e.g. [2], will require at least exponential time to terminate in the worst case.

Our work here studies a case that we believe has been missed in the above panoply of results, but has a distinguished tradeoff between the desired expressiveness and the desired tractability. Here, we allow arbitrary uninterpreted function symbols, some subset of which is labeled as associative, commutative, and idempotent (ACI). ACI functions that arise naturally most obviously include intersection/union, and/or, and integer maximization/minimization. Program analysis frequently involves integer maximization or Boolean abstractions.

Here, we provide a cubic-time complete inference algorithm for congruence closure in this setting. Our procedure is defined using the technology of *local rule sets* [11, 20]; local rule sets are those for which the computed inference relation is unchanged by restricting inference to the terms mentioned in the premises and query. Any local rule set describes a polynomial-time decidable inference relation. In addition to providing a complete inference relation for the ACI congruence closure problem using a local rule set, we consider the problem of adding the ACI designation to function symbols in an arbitrary rule set. With no restrictions on the rule set, adding this designation may be expensive; however, we present a natural restriction on inference rules for which any local rule set can be augmented by ACI designations while retaining locality and thus polynomial-time decidability. Under this restriction, ACI function-symbol arguments can only be accessed by the inference rules in manners independent of the order or multiplicity of their presentation.

We proceed as follows: first, we present brief technical background on locality and on congruence closure, including a local rule set for congruence closure. Second, we present a simple rule set for reasoning about the ACI properties of terms resulting from ACI function symbols. Third, we prove this rule set together with the congruence closure rule set provides semantically sound and complete inference for the congruence closure problem with multiple ACI function symbols. Fourth, we prove that the resulting rule set is local, immediately providing a cubic-time procedure for the inference relation. Finally, we discuss integration with arbitrary rule sets restricted to access ACI terms via tuples of their arguments.

2 Technical Background

2.1 Locally Restricted Inference

We adopt the following definitions for reference almost directly from [11]. In the these definitions, Σ is any set of ground atomic formulas in first-order logic, and φ any single ground atomic formula.

Definition 1. *A* Horn clause *is a first order formula of the form* $(\psi_1 \wedge \ldots \wedge \psi_n) \rightarrow \psi$ *where* ψ *and the* ψ_i *are atomic formulas. For any* rule set *of Horn clauses* R*, we write* $\Sigma \vdash_R \varphi$ *whenever* $\Sigma \cup U(R) \vdash \varphi$ *in first-order logic, where* $U(R)$ *is the set of universal closures of Horn clauses in* R*.*

We can characterize $\vdash$ syntactically by defining derivations under the rules R.

Definition 2. *A* derivation *from Σ using rule set R is a sequence of ground atomic formulas $\psi_1, \ldots, \psi_n$ such that ψ_n is φ and for each ψ_i there exists a Horn clause $\theta_1 \wedge \ldots \wedge \theta_k \rightarrow \theta$ in R and a ground substitution σ such that $\sigma[\theta]$ is ψ_i and each formula of the form $\sigma[\theta_j]$ is either a member of Σ a formula ψ_j for $j < i$. The* length *of the derivation is the number n of ground formulas in the sequence.*

We then have $\Sigma \vdash_R \varphi$ if and only if there is a derivation of φ from Σ using R. Next, by restricting inference to terms mentioned in Σ or φ, we get a polynomial-time decidable inference relation that may or may not the same as $\vdash_R$.

Definition 3. *Let Υ be a set of terms that is closed under the subterm relation. We say that a ground atomic formula ψ is* local to Υ *if every term appearing in ψ is in Υ. For Γ a set of ground atomic formulas, let $\Upsilon(\Gamma)$ be the subterm-closed set of terms appearing in Γ. We write $\Sigma \Vdash_{R,\Upsilon} \varphi$ if there exists a local derivation of φ from Σ, i.e. one such that every atomic formula in the derivation is local to Υ. We omit Υ to get $\Sigma \Vdash_R \varphi$ when Υ is $\Upsilon(\Sigma \cup \{\varphi\})$.*

McAllester [20] provides a simple proof that the inference relation $\Vdash_R$ is polynomial-time decidable for any finite R; a straight-forward inference procedure can grow a set of derivable ground atoms from Σ by repeatedly considering each inference rule at polynomial time cost, staying always within the polynomially many ground atoms local to $\Upsilon(\Sigma \cup \{\varphi\})$.

Definition 4 (McAllester, 1993). *The rule set R is* local *if the restricted inference relation $\Vdash_R$ is the same as the unrestricted inference relation $\vdash_R$.*

Every inference relation that can be defined by a local rule set is then polynomial-time decidable. It has also been shown [11] that every polynomial-time predicate can be defined by a local rule set.

2.2 Congruence Closure

The following inference rules E define a complete, local inference relation for ground atomic equational premises and ground equational queries.

(eq-refl) $\rightarrow x = x$ (eq-symm) $x = y \rightarrow y = x$

(eq-trans) $x = y \wedge y = z \rightarrow x = z$ (eq-congr1) $x = y \rightarrow f(x) = f(y)$

(eq-congr2) $x_1 = y_1 \wedge x_2 = y_2 \rightarrow g(x_1, x_2) = g(y_1, y_2)$

For notational simplicity, we assume wlog that all function symbols have arity at most 2. We note that the eq-congr rules are actually abbreviations for finitely many rules, one for each function symbol of the given arity. When the equality rules are included with other rule sets, we will also assume eq-congr rules for each predicate symbol of the larger rule set, representing the same principle of substitution of equals: e.g., $x = y \wedge P(x) \rightarrow P(y)$.

The rule set E has been proven complete even when locally restricted ($\Vdash_E$ is complete), proving E is local and providing a polynomial-time decision procedure for this problem, called *congruence closure* [21]. We refer to any rule set containing the rules of E, among others, as an *equational* rule set.

3 A Congruence/ACI Rule Set

Here, we provide a set ACI of inference rules for binary function symbols known to be ACI (associative, commutative, and idempotent ($f(x,x) = x$)). We assume now that some subset of the function symbols have been designated as ACI. For each such function symbol f, we introduce a new binary predicate $\preccurlyeq_f$, and the rules in ACI are schemas providing one rule for each such function symbol. The rules can be best understood by thinking of the new atoms $x \preccurlyeq_f y$ as standing for "there exists w such that $f(x,w) = y$."

$$\begin{array}{rlrl}
\text{(ACI-trans)} & x \preccurlyeq_f y \ \wedge\ y \preccurlyeq_f z \ \rightarrow\ x \preccurlyeq_f z & \text{(ACI-refl)} & \rightarrow\ x \preccurlyeq_f x \\
\text{(ACI-sup)} & x \preccurlyeq_f z \ \wedge\ y \preccurlyeq_f z \ \rightarrow\ f(x,y) \preccurlyeq_f z & \text{(ACI-sub1)} & \rightarrow\ x \preccurlyeq_f f(x,y) \\
\text{(ACI-antisym)} & x \preccurlyeq_f y \ \wedge\ y \preccurlyeq_f x \ \rightarrow\ x = y & \text{(ACI-sub2)} & \rightarrow\ y \preccurlyeq_f f(x,y)
\end{array}$$

Here, the new predicates $\preccurlyeq_f$ implement inference from the ACI axioms, but these predicates are not intended to be part of premise sets or query formulas. We designate the new predicates as *hidden*, prohibiting them in premise sets and queries. This designation affects the claims we later make of rule-set locality and completeness. In each case, these claims refer to inference problems without the hidden predicates in the premises or query.

In the coming sections, we will prove this rule set, in combination with the equality rules E, is both correct ($\vdash_{E \cup \text{ACI}}$ is sound and complete) and local ($\Vdash_{E \cup \text{ACI}}$ is the same as $\vdash_{E \cup \text{ACI}}$), thus providing a polynomial time inference procedure for the congruence closure problem in the presence of known ACI function symbols.

4 Correctness of the Congruence/ACI Rule Set

In this section we prove that the inference rules $E \cup \text{ACI}$ are sound and complete for ground inference on equations. Since our rules and our premise sets are first-order formulas, completeness here refers to standard first-order logic interpretations. Note again that the introduced predicates $\preccurlyeq_f$ are considered hidden and do not appear in premise sets or queries.

Definition 5. *The* ACI-congruence theory $\mathcal{A}$ *is the set of first-order axioms of equality (reflexivity, symmetry, transitivity, and substitution of equals for equals) together with the axioms of associativity, commutativity, and idempotence for each ACI function symbol. A ground premise set Σ of equations entails a ground query equation φ in the ACI-congruence theory, written $\Sigma \cup \mathcal{A} \vDash \varphi$, when every first-order interpretation satisfying $\Sigma \cup \mathcal{A}$ also satisfies φ.*

Theorem 1 (Soundness and Completeness). *For any ground premise set of equations Σ and ground query equation φ, we have $\Sigma \cup \mathcal{A} \vDash \varphi$ if and only if $\Sigma \vdash_{E \cup \text{ACI}} \varphi$.*

Proof. We consider each direction of the theorem separately.

(Soundness) A simple induction on derivation length shows that every ground atom in a derivation from Σ using $E \cup \text{ACI}$ satisfies the following invariants:

$=$: Atoms $x = y$, for any terms x and y, satisfy $\Sigma \cup \mathcal{A} \vDash x = y$.

$\preccurlyeq_f$: Atoms $x \preccurlyeq_f y$, for any terms x and y and ACI function symbol f, satisfy $\Sigma \cup \mathcal{A} \vDash \exists z f(x, z) = y$.

Each rule preserves these invariants. The rule ACI-antisym is the hardest to check. We must check that $\mathcal{A} \cup \{\exists z f(x, z) = y, \exists w f(y, w) = x\} \vDash x = y$. Introducing Skolem constants we have $f(x, c_1) = y$ and $f(y, c_2) = x$. From these we can show $f(x, f(y, f(c1, c2)))$ is equal to x and also to y using the given premises. For instance $f(x, f(y, f(c1, c2))) = f(f(f(x, c1), y), c2) = f(f(y, y), c2) = f(y, c2) = x$. Thus the conclusion of the rule, $x = y$, is entailed as stated in the invariant for $=$. Soundness of all derived equation atoms is then immediate.

(Completeness) We prove completeness with a standard model-construction approach. We exhibit a first-order interpretation $\mathcal{I}$ that satisfies $\Sigma \cup \mathcal{A}$ and satisfies exactly equations φ that are derivable from Σ using rules $E \cup \text{ACI}$. We start by defining an equivalence relation $\triangleq$ on terms based on the derivable equations: $t_1 \triangleq t_2$ if and only if $\Sigma \vdash_{E \cup \text{ACI}} t_1 = t_2$. The rules in E imply that $\triangleq$ is an equivalence relation. We take the domain $D_{\mathcal{I}}$ of the interpretation to be the set $\{[t]_{\triangleq} \mid t \text{ a term}\}$ of all equivalence classes of terms under $\triangleq$.

Next, we define the interpretation under $\mathcal{I}[\![f]\!]$ of each function symbol f. We show the two argument case, which includes all ACI function symbols, but this definition is easily generalized to function symbols of any number of arguments. We define $I(f)$ on domain objects $[t_1]_{\triangleq}$ and $[t_2]_{\triangleq}$ to be $[f(t_1, t_2)]_{\triangleq}$. The inference rule (eq-congr) for the symbol f ensures that the defined output of f on two domain elements does not depend on which terms t_1 and t_2 are selected from the equivalence classes.

We next define the interpretation $\mathcal{I}$ for the predicate symbols. Like any first-order interpretation, $\mathcal{I}$ interprets equality as the identity function. We define $\mathcal{I}[\![\preccurlyeq_f]\!]$, for each ACI function-symbol f, to be true on domain objects $[t_1]_{\triangleq}$ and $[t_2]_{\triangleq}$ if and only if $\Sigma \vdash_{E \cup \text{ACI}} t_1 \preccurlyeq_f t_2$. Again, the (eq-congr) rule for the predicate $\preccurlyeq_f$ ensures the truth value does not depend on the choice of t_1 and t_2.

A straightforward induction on the compositional structure of an arbitrary term t shows that the interpretation $\mathcal{I}[\![t]\!]$ of t under $\mathcal{I}$ is $[t]_{\triangleq}$. It then follows that $\mathcal{I}$ satisfies an equation $t_1 = t_2$ if and only if $t_1 = t_2$ is derivable from Σ using $E \cup \text{ACI}$: i.e., we have $\Sigma \vdash_{E \cup \text{ACI}} t_1 = t_2$ if and only if $[t_1]_{\triangleq} = [t_2]_{\triangleq}$, by definition, if and only if $\mathcal{I}[\![t_1]\!] = \mathcal{I}[\![t_2]\!]$, by our inductive lemma, and thus if and only if $\mathcal{I} \vDash t_1 = t_2$, as desired. It is then immediate that $\mathcal{I}$ satisfies Σ and all equations derived from Σ, and no others.

It remains to argue that $I \vDash \mathcal{A}$. We have that $\mathcal{I}$ satisfies the first-order axioms of equality because $\mathcal{I}[\![=]\!]$ is the identify function. Consider the associativity axiom for f, $\forall x \forall y \forall z f(f(x, y), z) = f(x, f(y, z))$. For any terms t_1, t_2, t_3, the ACI rule set justifies the following derivation: $t_1 \preccurlyeq_f f(t_1, t_2)$, $t_2 \preccurlyeq_f f(t_1, t_2)$, $f(t_1, t_2) \preccurlyeq_f f(f(t_1, t_2), t_3)$, $t_1 \preccurlyeq_f f(f(t_1, t_2), t_3)$, $t_2 \preccurlyeq_f f(f(t_1, t_2), t_3)$, $t_3 \preccurlyeq_f f(f(t_1, t_2), t_3)$, $f(t_2, t_3) \preccurlyeq_f f(f(t_1, t_2), t_3)$, $f(t_1, f(t_2, t_3)) \preccurlyeq_f f(f(t_1, t_2), t_3)$, $\ldots$, $f(f(t_1, t_2), t_3) \preccurlyeq_f f(t_1, f(t_2, t_3))$, $f(f(t_1, t_2), t_3) = f(t_1, f(t_2, t_3))$. Thus, $[f(f(t_1, t_2), t_3)]_{\triangleq}$ is the same as $[f(t_1, f(t_2, t_3))]_{\triangleq}$. Now considering arbitrary do-

main members $[t_1]_{\triangleq}$,$[t_2]_{\triangleq}$, and $[t_3]_{\triangleq}$, the definition of $\mathcal{I}$ implies that

$$\mathcal{I}[\![f]\!](\mathcal{I}[\![f]\!]([t_1]_{\triangleq}, [t_2]_{\triangleq}), [t_3]_{\triangleq}) = [f(f(t_1, t_2), t_3)]_{\triangleq},$$

and likewise $\mathcal{I}[\![f]\!]([t_1]_{\triangleq}, \mathcal{I}[\![f]\!]([t_2]_{\triangleq}, [t_3]_{\triangleq}))$ is $[f(t_1, f(t_2, t_3))]_{\triangleq}$, and thus every instance of associativity is satisfied by $\mathcal{I}$, and so is the associativity axiom. Similar arguments justify commutativity ($\forall x \forall y f(x, y) = f(y, x)$) and idempotence ($\forall x f(x, x) = x$).

We have thus exhibited $\mathcal{I}$ satisfying $\Sigma \cup \mathcal{A}$ but not satisfying φ for any φ such that $\Sigma \not\vdash_{E \cup \mathrm{ACI}} \varphi$, as desired. Q.E.D.

5 Effective Decidability of the Congruence/ACI Rule Set

We next show that the rule set $E \cup \mathrm{ACI}$ is local ($\vdash_{E \cup \mathrm{ACI}} = \Vdash_{E \cup \mathrm{ACI}}$), implying immediately that the inference relation $\vdash_{E \cup \mathrm{ACI}}$ on ground equations is polynomial-time decidable. It then follows by Theorem 1 that entailment from ground premise sets in theory $\mathcal{A}$ is polynomial-time decidable.

The rule-set property of locality has been shown undecidable in general [11], but sub-classes of the local rule sets (inductively local rule sets [11] and bounded local rule sets [20]) have been shown to be automatically recognizable with procedures that fail to terminate on inputs representing local rule sets outside the sub-class. We have not been able to get these procedures to terminate on our rule set $E \cup \mathrm{ACI}$, leaving unresolved(prior to this work) the question of locality for this rule set.

A semantic proof of locality can be constructed by showing that the partial model constructed by locally restricted inference can always be extended (embedded) to a full model. This technique was first presented in [10] and exploited in analyzing theory combinations in [24, 14]. The latter work can be used to handle the uninterpreted function symbols present in our setting, and Dedekind-MacNeille completion [17] can be used to show the semantic embeddability needed to construct the full model in the case where we have only one ACI function symbol [1]. However, we do not know of work extending Dedekind-MacNeille completion to multiple function symbols (i.e. multiple partial orders). A single completion step on one partial order destroys the ordering information in the others. Due to the difficulty we encountered constructing a semantic proof along these lines, we instead present a direct syntactic proof of locality. This proof sheds syntactic insight on why local restriction does not change the inference relation, and is of value even if the semantic proof can be repaired. For the remainder of this section, we use the shorthand K to abbreviate $E \cup \mathrm{ACI}$ for readability.

One approach to proving locality of K is to prove semantic completeness of the restricted inference $\Vdash_K$. However, unlike in the pure congruence closure case ($\Vdash_E$), the model-theoretic completeness proof of Theorem 1 does not adapt directly to the $\Upsilon(\Sigma, \varphi)$-restricted inference of $\Vdash_K$. The proof encounters difficulties in defining ACI function symbols. In the pure congruence-closure case, function symbols can be arbitrarily defined in cases that produce terms in equivalence classes outside of $\Upsilon(\Sigma, \varphi)$, producing a simple finite model, but the ACI

theory restricts these definitions in complex ways. Due to the difficulties we encountered with a semantic approach, we instead prove locality with a syntactic analysis below.

For this proof, we first note that our definitions immediately imply that $\Sigma \vdash_K \varphi$ implies $\Sigma \Vdash_{K,\Upsilon} \varphi$ for some sufficiently large finite term set Υ containing $\Upsilon(\Sigma \cup \{\varphi\})$ and all terms appearing in the derivation underlying $\Sigma \vdash_K \varphi$. It follows that we can prove locality by proving that, for any ground premise set Σ and ground equation φ, and every finite subterm-closed term set Υ containing at least $\Upsilon(\Sigma \cup \{\varphi\})$, $\Sigma \Vdash_K \varphi$ if and only if $\Sigma \Vdash_{K,\Upsilon} \varphi$. $(*)$

We will prove $(*)$ by induction on the construction of Υ from $\Upsilon(\Sigma \cup \{\varphi\})$.

(Base case) $\Upsilon = \Upsilon(\Sigma \cup \{\varphi\})$. In this case, the claim $(*)$ holds trivially.

(Inductive case) We suppose the claim $(*)$ holds for a subterm-closed term set Υ containing $\Upsilon(\Sigma \cup \{\varphi\})$. Let α be an arbitrary term not in Υ such that every proper subterm of α is in Υ. We show that the claim $(*)$ holds with the set $\Upsilon' = \Upsilon \cup \{\alpha\}$ in place of Υ. Here, we will assume that α is $g(t_1, t_2)$ for some (ACI-labeled or not) g and terms t_1 and t_2; the other arity cases are similar but do not include the ACI possibility.

To show this, we consider all derivations from Σ using rules in K within Υ', characterizing every atomic formula that can appear in such a derivation. We show by a second, nested induction on the length of the derivation that the derived formula must fall into one of several classes. To define these classes of formula, we need some new notation. In order to use inference within Υ to characterize the terms t such that $t \preccurlyeq_g \alpha$ can be proven within Υ', if g is ACI we define the g-closure of the set $\{t_1, t_2\}$, written $\widehat{g}(\{t_1, t_2\})$, to be the closure of the set $\{t_1, t_2\}$ under the following two operations:

- For any term y in the set, add every z such that $\Sigma \Vdash_{K,\Upsilon} z \preccurlyeq_g y$.
- For any terms x and y in the set, add $g(x, y)$ if that term is in Υ.

In order to use inference within Υ to characterize those equations that are provable in Υ' between α and other terms e in Υ, we define $\langle\langle g(t_1, t_2) = e \rangle\rangle$ to hold if one of the following holds about inference with Υ:

- $\Sigma \Vdash_{K,\Upsilon} t_1 \preccurlyeq_g e \quad \wedge \quad \Sigma \Vdash_{K,\Upsilon} t_2 \preccurlyeq_g e \quad \wedge \quad e \in \widehat{g}(\{t_1, t_2\})$, and g is ACI, or
- $\Sigma \Vdash_{K,\Upsilon} x_1 = t_1 \quad \wedge \quad \Sigma \Vdash_{K,\Upsilon} x_2 = t_2 \quad \wedge \quad \Sigma \Vdash_{K,\Upsilon} g(x_1, x_2) = e$ for some terms x_1 and x_2.

These properties ensure that $\alpha = e$ will be inferred in Υ'. The first bullet characterizes equations by ACI inference and the second characterizes standard congruence inference. We will need the following lemmas about these definitions.

Lemma 1. *If* $\langle\langle g(t_1, t_2) = e_1 \rangle\rangle$ *then* $\langle\langle g(t_1, t_2) = e_2 \rangle\rangle$ *iff* $\Sigma \Vdash_{K,\Upsilon} e_1 = e_2$.

We are now ready to characterize the classes of formulas that can appear in derivations from Σ using rules K within Υ'. Every such formula must fall in one of the following classes[1]:

[1] In some cases class C2 will be contained in class C3; otherwise, all the classes are disjoint.

[C1] A formula derivable from Σ using rules K within Υ,

[C2] A reflexive $=$ or $\preccurlyeq_f$ formula about α, for any ACI f,

[C3] An atomic formula β, excluding $\preccurlyeq_g$, local to Υ' and mentioning α where the same atom is derivable within Υ with α replaced by some term $e \in \Upsilon$ (so that $\Sigma \Vdash_{K,\Upsilon} [e/\alpha]\beta$), where $\langle\!\langle g(t_1, t_2) = e \rangle\!\rangle$,

[C4] An atom $e \preccurlyeq_g \alpha$ for $e \in \widehat{g}(\{t_1, t_2\})$, or

[C5] An atom $\alpha \preccurlyeq_g e$ for which both $\Sigma \Vdash_{K,\Upsilon} t_1 \preccurlyeq_g e$ and $\Sigma \Vdash_{K,\Upsilon} t_2 \preccurlyeq_g e$.

Classes C4 and C5 are considered empty if g is not labeled ACI. These classes cover all formulas that can appear in derivations under $\Vdash_{K,\Upsilon'}$. This can be verified by induction on the length of the derivation, checking that each inference rule preserves the claim. We discuss the critical (ACI-antisym) inference rule here as an example. The rule antecedents are $x \preccurlyeq_f y$ and $y \preccurlyeq_f x$. We must show that the consequent $x = y$ falls in one of the five given classes. There are several cases to consider (up to symmetries):

1. $\alpha \notin \{x, y\}$. Both antecedents and then the conclusion must be in class C1.
2. $x = y = \alpha$. The conclusion is in class C2.
3. $x \neq y = \alpha$, $f \neq g$. Then $x \in \Upsilon$. Both antecedents must be in class C3. So there must be $e_1 \in \Upsilon$ such that $\langle\!\langle g(t_1, t_2) = e_1 \rangle\!\rangle$ and $\Sigma \Vdash_{K,\Upsilon} x \preccurlyeq_f e_1$. Likewise there must be $e_2 \in \Upsilon$ such that $\langle\!\langle g(t_1, t_2) = e_2 \rangle\!\rangle$ and $\Sigma \Vdash_{K,\Upsilon} e_2 \preccurlyeq_f x$. It follows from Lemma 1 that $\Sigma \Vdash_{K,\Upsilon} e_1 = e_2$. It then follows that $\Sigma \Vdash_{K,\Upsilon} x \preccurlyeq_f e_2$ using that (eq-congr) is in K, and that $\Sigma \Vdash_{K,\Upsilon} e2 = x$ using that (aci-antisym) is in K. Lemma 1 then implies $\langle\!\langle g(t_1, t_2) = x \rangle\!\rangle$. But then since $\Sigma \Vdash_{K,\Upsilon} x = x$, the rule conclusion $x = \alpha$ satisfies the C3 class invariant.
4. $x \neq y = \alpha$, $f = g$. Then $x \in \Upsilon$. Antecedent $x \preccurlyeq_g \alpha$ is in class C4, and antecedent $\alpha \preccurlyeq_g x$ is in class C5. We need to show the conclusion $x = \alpha$ is in class C3. To do so, we exhibit $e \in \Upsilon$ such that $\Sigma \Vdash_{K,\Upsilon} x = e$ and $\langle\!\langle g(t_1, t_2) = e \rangle\!\rangle$. From the antecedents, using the induction hypothesis about classes C4 and C5, we have $\Sigma \Vdash_{K,\Upsilon} t_1 \preccurlyeq_g x$ and $\Sigma \Vdash_{K,\Upsilon} t_2 \preccurlyeq_g x$, as well as $x \in \widehat{g}(\{t_1, t_2\})$. Then by definition, $\langle\!\langle g(t_1, t_2) = x \rangle\!\rangle$, so x is the desired e.

These cases together demonstrate that any instance of (ACI-antisym) in a derivation satisfying the claim up to that point will extend that derivation with a formula that also satisfies the claim. Together with similar analyses of all the other rules [12], we conclude that every formula in a derivation from Σ using K within Υ' falls in one of these five classes.

Since every formula in these classes either mentions α (classes C2 to C5) or is derivable within Υ (class C1), we have shown that for any φ local to Υ, $\Sigma \Vdash_{K,\Upsilon} \varphi$ if and only if $\Sigma \Vdash_{K,\Upsilon'} \varphi$, which together with our (outer) induction hypothesis implies that $(*)$ holds for Υ', as desired. We have thus shown the following theorem.

Theorem 2. *The rule set $E \cup \mathrm{ACI}$ is local. The inference relation $\vdash_{E \cup \mathrm{ACI}}$ is polynomial-time decidable.*

The techniques discussed in [20] provide a straightforward $\Theta(n^3)$ procedure via the locally restricted inference process.

6 Integrating ACI Functions into Other Rule Sets

We now turn our attention to general equational rule sets for which we may wish to designate some of the function symbols as ACI. We suppose an arbitrary equational rule set R is known to be local and consider the question of whether $R \cup \text{ACI}$, with some function symbol f labeled to be ACI, remains local.

In fact, it is easy to see that the addition of ACI in $R \cup \text{ACI}$ will not in general preserve locality. Rules in R may contain specific nested applications of the ACI-labeled function symbol f, and if the local expressions contain ACI-equivalent terms that don't match that specific nesting, the rules involved will not be part of derivations. For example, the single-rule local rule set $\{P(f(x, f(y, z))) \rightarrow Q(z)\}$ can draw no conclusions with local derivations on premise set $P(f(f(a, b), c))$, but upon expanding the rule set with the ACI rules, can conclude $Q(c)$. Rules like this one are using the function symbol f in a manner somehow inconsistent with the assumption that f is ACI.

As an example, consider the following local rule set for binary intersections and unions [11]:

$$\rightarrow\ x \subseteq x \qquad x \subseteq y\ \wedge\ y \subseteq z\ \rightarrow\ x \subseteq z$$

$$\rightarrow\ y \subseteq x \cup y \qquad x \subseteq z\ \wedge\ y \subseteq z\ \rightarrow\ x \cup y \subseteq z \qquad \rightarrow\ x \subseteq x \cup y$$

$$\rightarrow\ x \cap y \subseteq y \qquad z \subseteq x\ \wedge\ z \subseteq y\ \rightarrow\ z \subseteq x \cap y \qquad \rightarrow\ x \cap y \subseteq x$$

The techniques in this section are designed to enable the addition of complete inference from the ACI properties for intersection and union to rule sets like this without losing the locality property that ensures efficient inference, by reformulating the ACI function symbols (in this case $\cup$ and $\cap$) as being applied to single arguments that are tuples of terms, where ACI properties are managed by equating ACI-equivalent tuples. What we show here is that a rule set that is local before adding these ACI properties on tuple argument, will remain so after this addition is made. The remainder of this section formalizes this idea. As an introduction, consider the tuple formulation of the rule set just presented:

$$\rightarrow\ x \subseteq x \qquad x \subseteq y\ \wedge\ y \subseteq z\ \rightarrow\ x \subseteq z \qquad \forall x \in \lambda, x \subseteq z\ \rightarrow\ \bigcup \lambda \subseteq z$$

$$x \in \lambda\ \rightarrow\ x \subseteq \bigcup \lambda \qquad x \in \lambda\ \rightarrow\ \bigcap \lambda \subseteq x \qquad \forall x \in \lambda, z \subseteq x\ \rightarrow\ z \subseteq \bigcap \lambda$$

Here, the λ rule variable represents a tuple of terms, with implicitly ACI function symbols $\bigcup$ and $\bigcap$ being applied to such tuples. Rule antecedents involving $\in$ with tuples are abbreviations as discussed below. The rule set is restricted from accessing the tuple structure in any other way than with the types of $\in$ antecedents shown here and formalized below. What we prove here, with some substantial difficulty, is that if the rule set is local before this transformation, without the ACI properties on its tuples, it will remain local when the ACI properties are enforced.

6.1 Every/Some ACI Rule Sets

Here, we propose to limit the ways that the rule set R can mention the ACI function symbols to ensure that ACI inferences do not interact badly with the rules. Our proposal below supports two interactions between ACI terms and

rules: testing that every (nested) argument to an ACI function symbol satisfies a predicate, and forcing a variable to represent an arbitrary (nested) argument to the ACI symbol. In addition, we note that any number of ACI function symbols can be represented if the theory supports a single ACI "tupling" function symbol. So we limit consideration to implementing a single ACI function symbol for tupling.

Kinds. In order to enforce separation between the ACI inference and the arbitrary rule set R, we introduce the concept of "kind" into the representation. For simplicity here, without loss of generality, we assume there are just two kinds: basic-term ($\mathcal{B}$) and tuple ($\mathcal{T}$). Every function symbol and predicate symbol has a signature over the kinds, so that every well-formed term has a syntactic kind. Applications of functions and predicates to expressions of the wrong kind are considered ill-formed. Every variable in a rule also has a kind and rule instances can only be formed by substituting variables with terms of the matching kind. The equational rules E, contained in any equational R, are assumed duplicated for each kind, with (eq-congr) present for each signature of function symbol or predicate symbol. Throughout this section, t, x, y, and z are basic-term variables, and λ is a tuple variable.

Tuples. The only expressions of tuple kind are formed from two function symbols: $\langle t \rangle$ coerces basic-term t into a (singleton) tuple, and $(\lambda_1 \cdot \lambda_2)$ combines two tuples. The function symbol $(\cdot)$ is the only function symbol labeled ACI. Every predicate symbol that operates on tuple arguments is designated hidden, i.e. cannot appear in premise sets, and besides equality on tuples the only such predicate symbols are introduced by the rules for ACI inference ($\preccurlyeq_{(\cdot)}$, $Z_\exists$, and $Z_{\forall P}$, introduced below). All other predicate symbols are not hidden. The only function symbols that have tuple arguments have exactly one argument—it is these function symbols that are *implicitly ACI* by accepting their arguments in tuple form. We write $x \in \lambda$ by abuse of notation to mean that x is a maximal basic-term subexpression of the tuple λ.

Rule set restrictions. We restrict the rule set R to contain only basic-term variables and expressions, except for the equational rules in E for tuples, mentioned above, and for the specific enrichment to use tuples that we propose next. We allow tuple variables in rules in the following forms:

- as sub-expressions of a basic-term (i.e. as an argument tuple), or
- in "every" or "some" antecedents, as detailed below.

Moreover, we require each tuple variable in a rule to occur at least once as a sub-expression of a basic-term.

Some. We allow rule antecedents of the form $x \in \lambda$. The basic-term variable x may be used elsewhere in the rule; this antecedent can be thought of as binding x for use elsewhere in the rule. Semantically, the "some" antecedent abbreviates $\exists z\, s = (\langle x \rangle \cdot z)$.

Every. We allow rule antecedents of the form $\forall x \in \lambda\, P(x)$ where P is any one-argument predicate symbol on basic-terms. For notational simplicity here, we assume P has no other arguments, but the extension to allow arbitrary other

basic-term arguments not mentioning x is straightforward. The basic-term variable x must not appear in the rule outside of this antecedent. We think of the $\forall$ as binding x locally to this antecedent. Semantically, the "every" antecedent abbreviates $\forall x\, \exists z\, \lambda = (\langle x \rangle \cdot z) \;\rightarrow\; P(x)$.

Syntactically, we eliminate "every" and "some" antecedents as follows. Introduce a new hidden predicate $Z_\exists$ of signature (basic-term $\times$ tuple) for the implementation of "some". For each predicate P appearing in any "every" antecedent, introduce a new hidden predicate $Z_{\forall P}$ on tuples, i.e., of signature (tuple). Replace each antecedent $x \in s$ with the atom $Z_\exists(x, \lambda)$, and each antecedent $\forall x \in \lambda\, P(x)$ with the atom $Z_{\forall P}(\lambda)$. Then, the following rules are added to R; it is straightforward to verify that these rules are sound under the semantics just given for "every" and "some" antecedents:

$$(Z_\exists\text{-sub1})\;\; Z_\exists(x, \lambda_1) \;\rightarrow\; Z_\exists(x, (\lambda_1 \cdot \lambda_2)) \qquad (Z_\exists\text{-init})\;\; \rightarrow\; Z_\exists(x, \langle x \rangle)$$

$$(Z_\exists\text{-sub2})\;\; Z_\exists(x, \lambda_2) \;\rightarrow\; Z_\exists(x, (\lambda_1 \cdot \lambda_2)) \qquad (Z_{\forall P}\text{-init})\;\; P(x) \;\rightarrow\; Z_{\forall P}(\langle x \rangle)$$

$$(Z_{\forall \_}\text{combine})\;\; Z_{\forall P}(\lambda_1) \wedge Z_{\forall P}(\lambda_2) \;\rightarrow\; Z_{\forall P}((\lambda_1 \cdot \lambda_2))$$

where the $Z_{\forall P}$ rules are present once for each predicate P appearing in an "every" antecedent.

Definition 6. *An* every/some rule set *is an equational rule set R for the kinds basic-term and tuple in which every tuple sub-expression is a tuple variable that occurs within a basic-term expression, and may also occur within "every" antecedents and/or "some" antecedents. The only predicate and function symbols involving the tuple kind are the tuple-constructors $\langle\rangle$ and $(\cdot)$, implicitly ACI function symbols (signature tuple $\rightarrow$ basic-term), and those abbreviated by every/some. Each such rule set abbreviates a rule set with no every/some antecedents via the transformation just described.*

For our purposes here, it is important to note that an every/some rule set does not yet have any ACI properties enforced. The rule set will not be able to infer the equivalence of tuples that are ACI variants of each other, and so function applications of implicitly ACI function symbols (applied to tuples) will also not benefit from the ACI properties. All inference on such terms will depend on the argument order and multiplicity. What we wish to show is that we can *add* the ACI properties while preserving the locality of the rule set.

6.2 Adding ACI to Every/Some Rule Sets Preserves Locality

Here we prove that we can add the theory ACI for tuples to any local every/some rule set R, preserving locality and thus polynomial-time decidability. The proof is rather technical and carefully constructed. We provide the key major structure here and refer to our website supplement [12] for many technical verifications underlying the proof.

Definition 7. *For any set of rules R, subterm-closed set of terms Υ, premise set Σ local to Υ, and positive integer k, and we write $C_k(\Sigma, R, \Upsilon)$ for the set of all formulas derivable from Σ using R with a derivation local to Υ of length at*

most k. We write $B_k(\Sigma, R, \Upsilon)$ *for the non-hidden subset of* $C_k(\Sigma, R, \Upsilon)$*. Finally, we write* $C(\Sigma, R, \Upsilon)$ *for* $\bigcup_{k=1}^{\infty} C_k(\Sigma, R, \Upsilon)$*, and likewise* $B(\Sigma, R, \Upsilon)$*.*

We will also need some notation and invariants characterizing exactly when tuples become related by inference.

Definition 8. *Given two sets of basic-term expressions* S_1 *and* S_2 *and a set of formulas* Γ*, we write* $(S_1 \subseteq_{eq} S_2) \in \Gamma$ *if there are equations in* Γ *to make* S_1 *a subset of* S_2*, i.e. if for every* $x \in S_1$ *there is* $y \in S_2$ *such that* $x = y \in \Gamma$*. Extending this notation to tuple expressions* λ_1 *and* λ_2*, we write* $(\lambda_1 \subseteq_{eq} \lambda_2) \in \Gamma$ *to abbreviate* $(\{x \mid x \in \lambda_1\} \subseteq_{eq} \{y \mid y \in \lambda_2\}) \in \Gamma$*. Finally, we write* $(\lambda_1 =_{eq} \lambda_2) \in \Gamma$ *if both* $(\lambda_1 \subseteq_{eq} \lambda_2) \in \Gamma$ *and* $(\lambda_2 \subseteq_{eq} \lambda_1) \in \Gamma$*.*

Using this new notation, the locality of a rule set R implies for instance that $B(\Sigma, R, \Upsilon) =_\Upsilon B(\Sigma, R, \Upsilon')$ for every subterm-closed Υ, premise set Σ local to Υ, and Υ' containing Υ. Here, we introduce the notation $=_\Upsilon$ to represent equality between two sets of formulas after restricting each set to the formulas local to Υ. We likewise define $\subseteq_\Upsilon$.

We can now state the key invariants regarding ACI inference on tuples. These invariants are stated for any every/some rule set R, subterm-closed Υ, premise set Σ local to Υ and positive integer k. We temporarily abbreviate the set of consequences $C_k(\Sigma, R \cup \text{ACI}, \Upsilon)$ as Γ_k and $C(\Sigma, R \cup \text{ACI}, \Upsilon)$ as Γ_∞:

1. (ACI-1) $\lambda_1 = \lambda_2 \in \Gamma_k$ implies $(\lambda_1 =_{\text{eq}} \lambda_2) \in \Gamma_k$, and conversely,
   $(\lambda_1 =_{\text{eq}} \lambda_2) \in \Gamma_k$ implies $\lambda_1 = \lambda_2 \in \Gamma_\infty$.
2. (ACI-2) $\lambda_1 \preccurlyeq_{(\cdot)} \lambda_2 \in \Gamma_k$ implies $(\lambda_1 \subseteq_{\text{eq}} \lambda_2) \in \Gamma_k$, and conversely,
   $(\lambda_1 \subseteq_{\text{eq}} \lambda_2) \in \Gamma_k$ implies $\lambda_1 \preccurlyeq_{(\cdot)} \lambda_2 \in \Gamma_\infty$.
3. (ACI-3) $Z_{\forall P}(\lambda) \in \Gamma_k$ implies $P(t) \in \Gamma_k$ for every $t \in \lambda$, and conversely,
   $P(t) \in \Gamma_k$ for every $t \in \lambda$ implies $Z_{\forall P}(\lambda) \in \Gamma_\infty$.
4. (ACI-4) $Z_\exists(t', \lambda) \in \Gamma_k$ implies $t = t' \in \Gamma_k$ for some $t \in \lambda$, and conversely,
   $t = t' \in \Gamma_k$ for $t \in \lambda$ implies $Z_\exists(t', \lambda) \in \Gamma_\infty$.

These invariants are easily demonstrated by induction on the length of derivations for the forward directions, showing that each rule preserves these invariants, and induction on the tuple structures for the converses. The same invariants, dropping ACI-2, can be shown replacing $R \cup \text{ACI}$ by R, but in stating ACI-1 using a narrower definition of $(\lambda_1 \subseteq_{\text{eq}} \lambda_2)$ that requires directly matching tuple structure from λ_1 and λ_2 (for lack of ACI rules).

We are now ready to develop the main theorem of this section. We restrict consideration to a particular local every/some rule set R, arbitrary subterm-closed term set Υ, and premise set Σ. We first consider the effect on inference from Σ using $R \cup \text{ACI}$ when we add a single basic-term to Υ. It is in this case that the locality of the base rule set R comes into play.

Lemma 2. *(Basic-term Extension) For any basic-term expression* α*, where all proper subexpressions of* α *are in* Υ*,* $B(\Sigma, R \cup \text{ACI}, \Upsilon \cup \{\alpha\}) =_\Upsilon B(\Sigma, R \cup \text{ACI}, \Upsilon)$*.*

Proof. The backward containment is immediate because $B(,,\Upsilon)$ is clearly monotone in Υ, as every derivation local to Υ is local to any superset of Υ.

To show the forward containment, the central proof idea is to observe that if $R \cup \text{ACI}$ has a locality violation, then we can construct such a locality violation for R on a premise set that contains Σ along with some additional premises (those non-hidden formulas that could have been derived by ACI if ACI were in use). Since the property of rule-set locality is a property that applies to all premise sets, the larger premise set cannot generate a locality violation under R (i.e. there can be no R-derivable fact from the larger premise set that is not derivable by locally restricted inference from that premise set).

To formalize this idea, we introduce an enriched premise set $\Sigma' = B(\Sigma, R \cup \text{ACI}, \Upsilon)$ and consider the consequences under inference from Σ' using rule sets $R \cup \text{ACI}$, within the enlarged term set $\Upsilon \cup \{\alpha\}$. The key observation, discussed next, is that any derivation of a new consequence within Υ using $R \cup \text{ACI}$ will have to start by using R alone to get the first new consequence within Υ. But R alone cannot get new consequences within Υ as R is local.

We now show the desired forward containment, but for Σ': $B(\Sigma', R \cup \text{ACI}, \Upsilon \cup \{\alpha\}) \subseteq_\Upsilon B(\Sigma', R \cup \text{ACI}, \Upsilon)$. To show this, suppose for contradiction that the desired containment is false, so that there must be some formula φ local to Υ in $B(\Sigma', R \cup \text{ACI}, \Upsilon \cup \{\alpha\})$ but not in $B(\Sigma', R \cup \text{ACI}, \Upsilon)$. In this context, we refer to formulas local to Υ in $B(\Sigma', R \cup \text{ACI}, \Upsilon \cup \{\alpha\})$ but not in $B(\Sigma', R \cup \text{ACI}, \Upsilon)$ as *newly derivable formulas*; φ is a newly derivable formula. Also here we refer to formulas local to Υ as *local*; φ is a newly derivable local formula.

Consider any derivation of φ from Σ' using $R \cup \text{ACI}$ within $\Upsilon \cup \{\alpha\}$. We refer to formulas in the derivation as *earlier* or *later* in the derivation according to their index in the sequence (with lower index corresponding to earlier). Consider the earliest newly derivable local formula β in the derivation, which occurs at latest at φ. Then every non-hidden local formula in the (prefix) derivation of β is in $B(\Sigma', R \cup \text{ACI}, \Upsilon)$ and thus in Σ', as Σ' is already closed under $R \cup \text{ACI}$ within Υ. We show that there is a derivation of β from Σ' using only rules in R, which contradicts the locality of R (which can't derive new formulas local to Υ by using α), to conclude the proof, as argued below.

Observe no instance of a rule in ACI within $\Upsilon \cup \{\alpha\}$ can mention α, since every expression in the ACI rules is a tuple expression, and α is not a subexpression of any tuple expression in $\Upsilon \cup \{\alpha\}$. Thus every formula in any ACI rule instance used in the derivation of β is local to Υ.

We now show that β is in $B(\Sigma', R, \Upsilon \cup \{\alpha\})$. We show that there is no formula in the derivation of β being considered that can be justified only by an ACI rule, from the derivation to that point and the premises Σ'. To show this, we show that there can be no earliest such formula η. Supposing, for contradiction, there is such η, then η is local to Υ, as just argued for ACI rule instances in the derivation of β, and cannot be a member of Σ' (or ACI rules would not be the only justification for η). Formula η cannot be a $\preccurlyeq_{(\cdot)}$ formula or it could only be justified by another ACI rule and thus would not be the earliest choice. Formula η cannot be a $Z_\exists$ or $Z_{\forall P}$ formula or it would not be a possible consequent of

an ACI rule. So, η must not be hidden. Thus, η is a non-hidden local formula in the derivation of β—so, by our choice of β, we have $\eta \in \Sigma'$, the premise set, contradicting the choice of η as requiring justification by an ACI rule. We conclude from this contradiction that every formula in the derivation of β is either in Σ' or can be justified by a rule in R, so β is in $B(\Sigma', R, \Upsilon \cup \{\alpha\})$. Since β is newly derivable, $\beta \notin B(\Sigma', R \cup \text{ACI}, \Upsilon)$ and thus $\beta \notin B(\Sigma', R, \Upsilon)$. This violates the locality of R, completing our proof that $B(\Sigma', R \cup \text{ACI}, \Upsilon \cup \{\alpha\}) \subseteq_\Upsilon B(\Sigma', R \cup \text{ACI}, \Upsilon)$.

But Σ' can be replaced by Σ in this claim, as the inference under $R \cup \text{ACI}$ within Υ that constructs Σ' from Σ is already included in both $B(,,)$ closures being considered. This gives us the desired containment to conclude our proof of the lemma. Formally,

$$\begin{aligned} B(\Sigma', R \cup \text{ACI}, \Upsilon \cup \{\alpha\}) &= B(B(\Sigma, R \cup \text{ACI}, \Upsilon), R \cup \text{ACI}, \Upsilon \cup \{\alpha\}) \\ &= B(\Sigma, R \cup \text{ACI}, \Upsilon \cup \{\alpha\}) \end{aligned}$$

and likewise

$$\begin{aligned} B(\Sigma', R \cup \text{ACI}, \Upsilon) &= B(B(\Sigma, R \cup \text{ACI}, \Upsilon), R \cup \text{ACI}, \Upsilon) \\ &= B(\Sigma, R \cup \text{ACI}, \Upsilon). \end{aligned}$$

Q.E.D. (Basic-term Extension Lemma)

A separate induction on derivation length is needed to handle extensions of Υ by new tuple expressions, leveraging the invariants stated above on tuple inference ((ACI-1) to (ACI-4)).

Lemma 3. *(Tuple Extension) For any tuple expression α, where all proper subexpressions of α are in Υ, $B(\Sigma, R \cup \text{ACI}, \Upsilon \cup \{\alpha\}) =_\Upsilon B(\Sigma, R \cup \text{ACI}, \Upsilon)$.*

Proof. The backward containment is again immediate. We prove the forward containment by induction on k to show, for all k, $B_k(\Sigma, R \cup \text{ACI}, \Upsilon \cup \{\alpha\}) \subseteq_\Upsilon B(\Sigma, R \cup \text{ACI}, \Upsilon)$. Please see our website supplement [12] for technical verification that each inference rule preserves this property of derivations, leveraging the tuple invariants (ACI-1) to (ACI-4). Here we discuss in detail the argument for one example inference rule chosen to illustrates all the key ideas.

Suppose for induction that $B_{k-1}(\Sigma, R \cup \text{ACI}, \Upsilon \cup \{\alpha\}) \subseteq_\Upsilon B(\Sigma, R \cup \text{ACI}, \Upsilon)$. Consider a k-step derivation ending in a non-hidden formula φ justified by an instance of an inference rule from the basic rule set R, but not an equational rule from E. We show that the conclusion of this derivation is a member of $B(\Sigma, R \cup \text{ACI}, \Upsilon)$. We will show (a) every antecedent formula of the inference instance used is in $C(\Sigma, R \cup \text{ACI}, \Upsilon)$, and (b) the consequent φ is local to Υ and not hidden. From these two statements we can conclude that $\varphi \in B(\Sigma, R \cup \text{ACI}, \Upsilon)$ as desired. To see (b), observe that φ is not hidden, so it cannot mention the new tuple α and so, being local to $\Upsilon \cup \{\alpha\}$ must also be local to Υ.

Then, to argue for (a), consider an arbitrary antecedent formula β of the rule instance. We have $\beta \in C_{k-1}(\Sigma, R \cup \text{ACI}, \Upsilon \cup \{\alpha\})$ since β appears in a k-step derivation as an antecedent. Please refer to our website supplement [12] for details of the induction proof. Q.E.D. (Tuple Extension Lemma)

By a simple induction on the construction of subterm-closed set Υ' containing Υ, these two lemmas directly imply that $R \cup \text{ACI}$ is local.

Theorem 3. *For any local every/some rule set R, the rule set $R \cup \text{ACI}$ is local.*

It follows that $R \cup \text{ACI}$ defines a polynomial-time decidable inference relation.

7 Conclusion

We have shown a local rule set provably providing sound and complete congruence closure with ACI function symbols. We also provide a detailed example integrating ACI inference into other local rule sets preserving locality.

References

1. Anonymous Reviewer: Personal email communication (March 2013)
2. Bachmair, L., Ramakrishnan, I., Tiwari, A., Vigneron, L.: Congruence closure modulo associativity and commutativity. FroCoS pp. 245–259 (2000)
3. Bachmair, L., Tiwari, A.: Abstract congruence closure and specializations. In: Proceedings of the 17th International Conference on Automated Deduction. pp. 64–78. CADE-17, Springer-Verlag, London, UK, UK (2000)
4. Bachmair, L., Tiwari, A., Vigneron, L.: Abstract congruence closure. J. Autom. Reason. 31(2), 129–168 (Dec 2003)
5. Bertot, Y., Castéran, P.: Interactive Theorem Proving and Program Development — Coq'Art: The Calculus of Inductive Constructions. SpringerVerlag (2004)
6. Bloniarz, P., Hunt III, H., Rosenkrantz, D.: Algebraic structures with hard equivalence and minimization problems. J. ACM 31(4), 879–904 (1984)
7. Burris, S.: Polynomial time uniform word problems. Mathematical Logic Quarterly 41(2), 173–182 (1995)
8. Conchon, S., Contejean, E., Kanig, J., Lescuyer, S.: CC (X): Semantic combination of congruence closure with solvable theories. Elec. Notes in Theor. Comp. Science 198(2), 51–69 (2008)
9. De Moura, L., Bjørner, N.: Z3: An efficient smt solver. Tools and Algorithms for the Construction and Analysis of Systems pp. 337–340 (2008)
10. Ganzinger, H.: Relating semantic and proof-theoretic concepts for polynomial time decidability of uniform word problems. In: Logic in Computer Science, 2001. Proceedings. 16th Annual IEEE Symposium on. pp. 81–90. IEEE (2001)
11. Givan, R., McAllester, D.: Polynomial-time computation via local inference relations. ACM Transactions on Computational Logic 3, 521–541 (Oct 2002)
12. Hu, T., Givan, R.: Additional proof details for ADDCT 2013 paper submisison. https://engineering.purdue.edu/∼relation/addct13.html (2013)
13. Hunt III, H., Rosenkrantz, D., Bloniarz, P.: On the computational complexity of algebra on lattices. SIAM Journal on Computing 16(1), 129–148 (1987)
14. Ihlemann, C., Sofronie-Stokkermans, V.: On hierarchical reasoning in combinations of theories. In: Automated Reasoning, pp. 30–45. Springer (2010)
15. Kapur, D.: Shostak's congruence closure as completion. In: Proceedings of the 8th International Conference on Rewriting Techniques and Applications. pp. 23–37. RTA '97, Springer-Verlag, London, UK, UK (1997)
16. Kozen, D.: Complexity of finitely presented algebras. In: Proceedings of the ninth annual ACM symposium on Theory of computing. pp. 164–177. STOC '77, ACM, New York, NY, USA (1977)
17. MacNeille, H.M.: Partially ordered sets. Trans. Amer. Math. Soc 42(3), 416–460 (1937)
18. Mayr, E., Meyer, A.: The complexity of the word problems for commutative semigroups and polynomial ideals. Advances in mathematics 46(3), 305–329 (1982)
19. McAllester, D.: Ontic: A Knowledge Representation System for Mathematics. MIT Press, Cambridge, MA (1989)
20. McAllester, D.: Automatic recognition of tractability in inference relations. Journal of the ACM 40(2), 284–303 (April 1993)
21. Nelson, G., Oppen, D.C.: Fast decision procedures based on congruence closure. Journal of the ACM 27(2), 356–364 (April 1980)
22. Rehof, J., Mogensen, T.: Tractable constraints in finite semilattices. Science of Computer Programming 35(2), 191–221 (1999)
23. Shostak, R.E.: An algorithm for reasoning about equality. Communications of the ACM 21(7), 583–585 (Jul 1978), http://doi.acm.org/10.1145/359545.359570
24. Sofronie-Stokkermans, V.: Hierarchic reasoning in local theory extensions. In: Automated Deduction–CADE-20, pp. 219–234. Springer (2005)