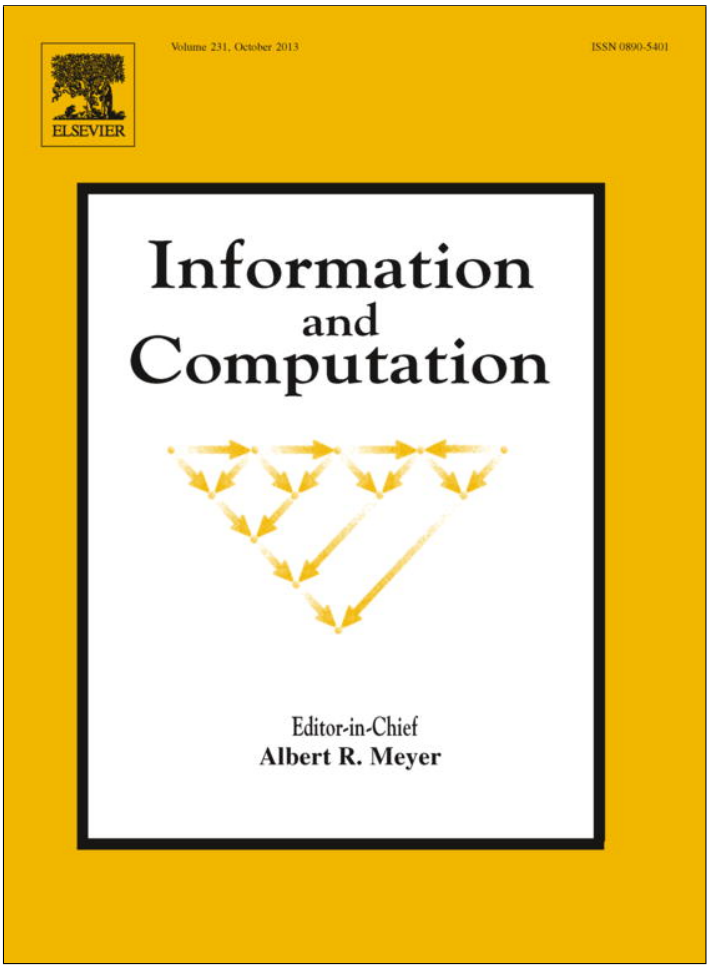

Information and Computation 231 (2013) 21–37

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Abstract Hilbertian deductive systems, infon logic, and Datalog

Andreas Blass [a,*,1], Yuri Gurevich [b]

[a] Mathematics Department, University of Michigan, Ann Arbor, MI 48109, USA
[b] Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA

ARTICLE INFO

Article history:
Available online 7 August 2013

Keywords:
Hilbertian
Deductive system
Infon logic
Datalog

ABSTRACT

In the first part of the paper, we discuss abstract Hilbertian deductive systems; these are systems defined by abstract notions of formula, axiom, and inference rule. We use these systems to develop a general method for converting derivability problems, from a broad range of deductive systems, into the derivability problem in a quite specific system, namely the Datalog fragment of universal Horn logic. In this generality, the derivability problems may not be recursively solvable, let alone feasible; in particular, we may get Datalog "programs" with infinitely many rules. We then discuss what would be needed to obtain computationally useful results from this method. In the second part of the paper, we analyze a particular deductive system, primal infon logic with variables, which arose in the development of the authorization language DKAL. A consequence of our analysis of primal infon logic with variables is that its derivability problems can be translated into Datalog with only a quadratic increase of size.

1. Introduction

This paper[2] grew out of the analysis of the infon logic introduced by the second author and Neeman [7] as a constituent of the Distributed Knowledge Authorization Language (DKAL) [2]. There are two versions of infon logic, full and primal (whose definitions we shall recall below). Among the main results of [7] are theorems delineating the computational complexity of the derivability problem (Given a set of hypotheses and a proposed conclusion, does the conclusion follow from the hypotheses?) for full and primal infon logics in the propositional case. The derivability problem is PSpace complete for full propositional infon logic and linear time for primal propositional infon logic, as long as quotation depths are bounded.[3] The latter result is particularly promising because primal infon logic (even with bounded quotation depth) is as good as full infon logic for many DKAL scenarios.

The restriction to propositional infon logic, on the other hand, is inappropriate in many applications. One often needs a version of infon logic that allows free variables, interpreted as universally quantified.

* Corresponding author.
E-mail addresses: ablass@umich.edu (A. Blass), gurevich@microsoft.com (Y. Gurevich).

[1] Partially supported by NSF grant DMS-0653696.

[2] This is a revised and corrected version of a paper with almost the same title that appeared in the *Bulletin of the European Association for Theoretical Computer Science*, vol. 102, October 2010, pp. 122–150. Because of the addition of Remarks 9 and 10 and the deletion of what used to be Definition 12, the numbering of this version differs from the previous one. Specifically, the numbers of old Remarks 9 to 11 have increased by 2, and the numbers of items from 13 up have increased by 1.

[3] A newer version of DKAL uses only one sort of quotation, "p `said`", where previously there was also "p `implied`". This simplification removes the need for bounding the quotation depth; the derivability problem for the new version of propositional primal logic is solvable in linear time, regardless of quotation depth [6].

0890-5401/$ – see front matter
http://dx.doi.org/10.1016/j.ic.2013.08.003

The question thus arises whether primal infon logic with variables also admits an efficient algorithm for the derivability problem, at least in typical cases.[4] In the present paper, we take the first steps toward an affirmative answer to this question. In the meantime, there has been further progress [1], building on the present work and making the method more practical.

Our approach is to transform the problem into Datalog, where efficient algorithms have already been developed. From the viewpoint of mathematical logic, Datalog can be described as universal Horn logic with two simplifications: First, there are no function symbols except for 0-ary ones (constants). That is, the vocabulary is *relational*. Second, although hypotheses are universal Horn formulas, the conclusions to be inferred from them are simply atomic formulas.

Convention 1. Throughout this paper, except where the contrary is explicitly stated, when we refer to Horn logic as a deductive system, we mean the limited version where only atomic formulas are to be deduced from Horn clauses.

To avoid possible ambiguity and to establish notation, we adopt the following convention concerning Horn clauses.

Convention 2. A *Horn clause* is a formula of the form $(\bigwedge X) \to \alpha$, where X is a finite set of atomic formulas, $\bigwedge X$ is the conjunction of all the formulas in X, and α is an atomic formula. Here X can be empty, in which case $\bigwedge X$ is to be understood as the always true formula $\top$; instead of $\top \to \alpha$, we usually write simply α, as they are semantically equivalent.

We begin by showing that a very broad class of deductive systems can be translated into Horn logic, in fact into its propositional fragment. For a slightly more restricted but still quite broad class of systems, there is a more useful translation into universal Horn logic and in fact into Datalog. In this generality, however, the translations suffer from a fatal (for computational purposes) defect: The translation of a (finite) derivability problem will generally involve infinitely many premises. That defect is, in general, unavoidable, but we shall show that, in the case of primal infon logic, the number of hypotheses in the Datalog translation of a given instance of the derivability problem can be reduced to a finite number and indeed to a number bounded by a linear function of the size of the given instance.

We thus show that any instance of the derivability problem for primal infon logic with variables can be converted into an equivalent Datalog problem, with only a quadratic increase in size.[5]

Although our translations do not satisfy the standard safety requirement that Datalog uses to limit the search for values to substitute for variables, we provide another effective way to limit that search.

The remaining difficulty, which we do not attempt to solve in this paper, is related to the following aspect of our translation: The arities of relations in the resulting Datalog instance may be substantially higher than the arities of relations in the given primal formula. The right way to deal with the derivability problem for primal logic with variables may be to deal with it directly rather than reduce it to the Datalog derivability problem. The direct approach may use some ideas of the best Datalog algorithms. That is future research. By the way, the current DKAL implementation [3] uses a direct method. Let us mention also that, in order to make this exposition simpler, we avoided various optimization tricks that could have been applied easily to our translation.

2. Abstract Hilbertian systems

In this section, we introduce the broad class of deductive systems for which there is a simple, natural translation into Horn logic.

Definition 3. An *abstract Hilbertian deductive system*[6] is given by

- a set F of (well-formed) *formulas*,
- a subset $\mathrm{Ax} \subseteq F$ called the set of *axioms*, and
- a set Ru of pairs $\langle P, \alpha \rangle$ where P is a finite subset of F and $\alpha \in F$; the elements of Ru are called *rules of inference*.

In the context of an abstract Hilbertian deductive system, a formula γ is said to be *deducible* from a set H of formulas if γ belongs to every set D of formulas that

- includes H,
- includes Ax, and
- is closed under Ru in the sense that, if $\langle P, \alpha \rangle \in \mathrm{Ru}$ and $P \subseteq D$ then $\alpha \in D$.

[4] We cannot expect much efficiency in the worst case, because derivability in Datalog is an exponential-time complete problem [4] and Datalog can be embedded in primal infon logic with variables; see Appendix A for more details.

[5] The increase in size is only linear if we measure the size of a set of formulas by the number (of occurrences) of connectives or of subformulas or of atomic formulas. If, however, we measure the number of symbols, then the increase can be quadratic, essentially because of relations with a large (linear) number of argument places.

[6] For brevity, we may omit "abstract" or "deductive" or both.

Quisani:[7] You use the terminology of logic – "formula", "axiom", "rule of inference" – but nothing in this definition requires the terminology to have anything like its usual logical meaning. You could let the "formulas" be real numbers, let the only "axiom" be just the number 0, and let the only "rule of inference" be $\langle\{x\}, x+1\rangle$. Then a formula deducible from the empty set would be just a natural number.

Authors: The overall drift of your comment is correct; abstract Hilbertian deductive systems are very general. In particular, the formulas in these systems need not have any meanings or truth values. They are purely syntactic objects, and the only required resemblance to logic is the presence of a notion of deducibility. But there's an error in your example, because what you call one rule of inference is actually a whole family of rules, one for each value of x.

Q: I see. So, returning to logic, you would call modus ponens a family of rules of inference, not a single rule.

A: That's right. Abstract Hilbertian systems as defined here make no provision for variables ranging over formulas, or, for that matter, any other variables. (We will, however, introduce "substitutional Hilbertian systems" in Section 3, and these will allow individual variables.)

One might reasonably call modus ponens a *rule scheme*, by analogy with axiom schemes. A rule scheme encodes a whole family of rules, obtained by inserting particular formulas into the scheme.

Q: It seems to me that all the usual deductive systems studied in various branches of logic would fall under your definition of "abstract Hilbertian".

A: Not quite. So-called natural deduction systems are not abstract Hilbertian. In such a system, to deduce a formula γ from a set H of hypotheses, you can temporarily introduce additional hypotheses and later cancel or "discharge" them. For example, to deduce an implication $\alpha \to \beta$ from H, you could temporarily assume α, deduce β from $H \cup \{\alpha\}$, and then infer $\alpha \to \beta$, discharging the assumption α.

Q: I see. Abstract Hilbertian systems don't allow discharging of hypotheses. But it seems that a natural deduction system could be converted into an abstract Hilbertian one rather easily. You could re-define "formula" to mean a formula in the original sense together with a set of temporary hypotheses. Then, what used to be called discharging becomes merely a change in the formula in the new sense. In your example, there would be a rule of inference allowing you to go from the formula "β with temporary hypotheses $\{\alpha\}$" to "$\alpha \to \beta$ with temporary hypotheses $\varnothing$".

A: That's essentially correct. You'd actually need to go from "β with temporary hypotheses $\{\alpha\} \cup X$" to "$\alpha \to \beta$ with temporary hypotheses X", for arbitrary finite sets X of formulas, because the inference in question might occur while some other temporary hypotheses are also in effect.

The "formulas in the new sense" that you propose are usually called *sequents* and written in the form $\Gamma \Rightarrow \alpha$ (or $\Gamma \vdash \alpha$), where α is the relevant formula in the original sense and Γ is the set of temporary hypotheses. Sequent calculi are intensively studied in proof theory; see for example [9] or [10]. They are abstract Hilbertian deductive systems, subject, as always, to the convention that what is usually called a rule of inference is, for us, a whole family of rules of inference.

We shall show next that an abstract Hilbertian deductive system is essentially a theory in propositional Horn logic.

Q: Wait a minute! You've agreed with me that just about any of the usual systems of logic is abstract Hilbertian, or at least becomes so when you convert natural deduction to sequents. That includes very complicated, undecidable systems. Propositional Horn logic, though, is decidable – in fact in linear time [8,5,11].[8] How can things with such vastly different complexities be essentially the same?

A: What, exactly, is the problem that you said is decidable in linear time?

Q: An instance consists of a set X of Horn clauses and an atomic formula γ; the question is whether γ is a logical consequence of X, or equivalently, whether γ is deducible from X in a standard axiomatization of propositional Horn logic.

A: Presumably, in this problem, X is finite; otherwise "linear time" would be forever.

Q: Sure; instances of problems should be finite.

Oh! Now I get it. You're going to translate arbitrary, possibly very complicated, abstract Hilbertian systems into propositional Horn logic, but with possibly an infinite number of hypotheses.

A: Exactly.

The translation works as follows. Given a Hilbertian system $S = \langle F, \mathrm{Ax}, \mathrm{Ru}\rangle$, introduce for each of its formulas α a propositional symbol [α is obtainable]. The *Horn version* $\hat{S}$ of S uses these propositional variables and has the following Horn clauses:

[7] Quisani is an attentive, inquisitive, and fictional graduate student of the second author.

[8] More information on the complexity of propositional logic programming is found in [4].

- for each $\alpha \in \mathrm{Ax}$, the clause [α is obtainable], and
- for each $\langle P, \alpha\rangle \in \mathrm{Ru}$ the *induced* clause

$$\left(\bigwedge_{\pi\in P} [\pi \text{ is obtainable}]\right) \to [\alpha \text{ is obtainable}].$$

Theorem 4. *A formula γ is deducible from a set H of hypotheses in an abstract Hilbertian system S if and only if* [γ *is obtainable*] *is deducible from* {[ξ *is obtainable*]: $\xi \in H$} *in the Horn version $\hat{S}$.*

Proof. Suppose first that γ is deducible in $S = \langle F, \mathrm{Ax}, \mathrm{Ru}\rangle$ from H. Let D be the set of those formulas $\delta \in F$ such that [δ is obtainable] is deducible from {[ξ is obtainable]: $\xi \in H$} in $\hat{S}$. Clearly, each $\delta \in H$ belongs to D. So does each axiom $\alpha \in \mathrm{Ax}$, since $\hat{S}$ contains the Horn clause [α is obtainable]. Furthermore, D is closed under Ru, because of the clauses of $\hat{S}$ induced by the rules in Ru. Therefore, by the definition of "deducible from H in S", we have $\gamma \in D$. This completes the proof of the "only if" half of the theorem.

For the converse, note that all the Horn clauses in $\hat{S}$ and all elements of {[ξ is obtainable]: $\xi \in H$} are true when we interpret each propositional variable [ξ is obtainable] as the assertion that ξ is deducible from H in S. Deduction in Horn logic preserves truth, so if [γ is obtainable] is deducible from {[ξ is obtainable]: $\xi \in H$} in $\hat{S}$ then [γ is obtainable] is true under this interpretation. □

Q: I have two objections to your notation [ξ is obtainable]. On the one hand, it's too long; on the other hand it's too short.

A: That sounds as if we got it pretty close to right. But seriously, what exactly are your objections?

Q: First, the "is obtainable" in your propositional variables seems superfluous. Why not just use the formulas of S as the propositional variables of $\hat{S}$?

A: We could indeed delete "is obtainable" but we would need to keep the brackets, at least if $\wedge$ or $\to$ occurred in formulas of S. That is, we need to maintain the distinction between a propositional variable of the form [$\alpha \to \beta$ is obtainable] (even if abbreviated as [$\alpha \to \beta$]) and the Horn clause [α is obtainable] $\to$ [β is obtainable] (even if abbreviated [α] $\to$ [β]), especially if the deductive system S treated $\to$ in some non-classical way (for example as relevant implication).

Q: OK, so you can't literally use formulas of S as formulas of $\hat{S}$; I concede that brackets are needed. But why "is obtainable"?

A: Those two words are not technically needed, but they are intended as an aid to intuition. In many cases, the formulas $\xi \in F$ are statements of some sort, and there is a notion of ξ being true.[9] The meaning of [ξ is obtainable], on the other hand, is intended to be that ξ is deducible in the formal system S, and that may be quite different from being true.

Q: I thought you might have something like that in mind. It brings me to my other objection: The words "is obtainable" are not specific enough, because they don't say what hypotheses are involved. "Obtainable" from what?

A: Well, as long as we're talking about a particular S, the axioms and rules of S are certainly available, so your problem seems to be with the additional hypotheses H.

Q: Right; to really make the intended meaning evident, your propositional variables should be [ξ is obtainable from H].

A: But then we'd have a different deductive system $\hat{S}$ for each set H. The point of our set-up is that we have one $\hat{S}$ that captures deducibility in S from any set H of hypotheses.

Remark 5. The notion of deducibility in a Hilbertian system admits an alternative characterization. A *deduction* of γ from H in S is a finite sequence of formulas, ending with γ, in which each formula either is an element of $H \cup \mathrm{Ax}$ or is inferred from previous formulas in the sequence via Ru. ("Inferred" means that the formula in question is the second component of a rule of inference whose first component consists of formulas that occur earlier in the sequence.) From this point of view, the proof of the "only if" direction in Theorem 4 can be recast as follows. Take a deduction Δ of γ from H in S. Replace each item ξ in Δ with [ξ is obtainable]. Finally, wherever a formula ξ was justified in Δ by applying a rule of inference, insert into the new, Horn deduction a few lines to justify [ξ is obtainable] using the Horn clause of $\hat{S}$ induced by the rule that justified ξ in Δ.

Remark 6. Another format for deductions will be useful later. A *tree deduction* of γ from H in S is a finite labeled tree, in which

- the labels are formulas of the Hilbertian system,

[9] Or, in the infon logic that motivated the present work, a notion of ξ being known by some agent.

- the root is labeled γ,
- if a non-leaf node is labeled α and the set of labels of its children is P, then $\langle P, \alpha\rangle$ is a rule of inference, and
- the labels of leaves are in $H \cup \mathrm{Ax}$.

We visualize these trees as growing upward, so the conclusion (root) γ is at the bottom and the hypotheses H at the top – the usual format for exhibiting proofs. Note that a hypothesis (or indeed any formula) that is among the premises in several inferences must, because of the tree structure, be repeated several times. Thus, a tree deduction may well be considerably larger than a (plain) deduction of the same conclusion from the same hypotheses, because in plain deductions a formula can be re-used without being repeated.

Remark 7. The notion of abstract Hilbertian system can be simplified by viewing any axiom α as a rule of inference $\langle \varnothing, \alpha\rangle$ with no premises. Under the usual convention that the empty conjunction is $\top$, everything we do in this paper continues to work under this alternative point of view.

3. Substitutional Hilbertian systems

In this section, we modify the preceding discussion by incorporating the use of free variables, interpreted as universally quantified, and rules of substitution for these variables.

Definition 8. A *substitutional Hilbertian system*[10] is an abstract Hilbertian system with the following additional properties:

1. Formulas are certain finite strings of symbols from a specified alphabet.
2. The alphabet includes a countably infinite set of symbols called *variables*.
3. A set (possibly empty) of non-variable symbols is designated as the set of *constants*.
4. If α is a formula and if β is obtained from α by uniformly replacing all occurrences of certain (distinct) variables $x_1, \ldots, x_k$ by variables or constants $u_1, \ldots, u_k$, respectively, then β is also a formula, and $\langle\{\alpha\}, \beta\rangle$ is a rule of inference. In this situation, we denote β by $\alpha(u_1/x_1, \ldots, u_k/x_k)$ or by $\alpha(\mathbf{u}/\mathbf{x})$ and call it a *substitution instance* of α. Rules of the form $\langle\{\alpha\}, \alpha(\mathbf{u}/\mathbf{x})\rangle$ are called *substitution rules*.
5. If $\langle\{\alpha_1, \ldots, \alpha_n\}, \beta\rangle$ is a rule of inference that is not a substitution rule, then, for any distinct variables $x_1, \ldots, x_k$ and any variables or constants $u_1, \ldots, u_k$, $\langle\{\alpha_1(\mathbf{u}/\mathbf{x}), \ldots, \alpha_n(\mathbf{u}/\mathbf{x})\}, \beta(\mathbf{u}/\mathbf{x})\rangle$ is also a rule of inference.

Remark 9. Our definition of substitution instances and substitution rules allows substitutions for any (finite) number k of variables simultaneously. This seems to be a natural definition, well adapted to Datalog where such multiple substitutions are used. Nevertheless, one could define a more frugal variant of substitutional Hilbertian systems by restricting clauses 4 and 5 of the definition to the case $k = 1$. In this case, we refer to *simple substitution instances* and *simple substitution rules*. Notice that, even if we only require the set of formulas to be closed under simple substitutions, it follows that it is closed under multiple substitutions as well. (This uses our assumption that infinitely many variables are available.) Similarly, the effect of a multiple substitution rule can be obtained by several successive applications of simple substitution rules.

Remark 10. Another inessential modification of the definition of substitutional Hilbertian system would be to require in clause 5 only that $\beta(u/x)$ is deducible from $\{\alpha_1(\mathbf{u}/\mathbf{x}), \ldots, \alpha_n(\mathbf{u}/\mathbf{x})\}$, possibly in several steps, rather than requiring that $\langle\{\alpha_1(\mathbf{u}/\mathbf{x}), \ldots, \alpha_n(\mathbf{u}/\mathbf{x})\}, \beta(u/x)\rangle$ is itself a rule of inference.

Remark 11. In applications, variables and constants usually have types, and substitution is required to preserve types. For simplicity of exposition, we omit types in this paper, but it is straightforward to incorporate them into all our results.

Remark 12. In a logical system with function symbols, variables usually admit substitution by arbitrary terms, not just by variables and constants. Our definition is designed to work under the assumption that there are no function symbols (except for constants if these are regarded as 0-ary function symbols). This assumption comes from Datalog and from our goal of translating our deductive systems into Datalog. If there were other function symbols, then our goal should be modified accordingly, and the preceding definition should provide for substituting compound terms for variables.

The key clauses 4 and 5 in the definition of "substitutional Hilbertian system" specify the role of variables in the system. Clause 4 says that variables behave as if they are universally quantified, in that they can be replaced by other terms. Clause 5 prevents other sorts of variable-manipulation in the system. Except for substitution, anything that can be done with variables can also be done with constants instead.

[10] A more accurate but more awkward name would be "Hilbertian system with free variables, understood as universally quantified".

Notice in this connection that what clause 5 requires of non-substitution rules cannot reasonably be required of substitution rules. That would require rules that substitute variables for constants, thereby essentially making the constants functionally indistinguishable from variables. For example, the presence of a substitution rule $\langle\{Ax\}, Ay\rangle$ would, if clause 5 applied to it, require the presence of a rule $\langle\{Ac\}, Ay\rangle$, where c is a constant. Worse, it would, in many systems, allow non-uniform substitutions. For example, if x, y are distinct variables, if a, b are distinct constants, and if Axy is a formula, then there is a substitution rule with premise Axy and conclusion Aab. If clause 5 applied to this substitution rule, it would provide a rule with premise Axx and conclusion Aab, which is certainly unwanted in almost any system.

Remark 13. If we were to adopt the convention that axioms are rules of inference with the empty set of premises (see Remark 7), then clause 5 of the definition of substitutional Hilbertian system would require that every substitution instance $\alpha(\mathbf{u}/\mathbf{x})$ of an axiom α is again an axiom. Without that convention, $\alpha(\mathbf{u}/\mathbf{x})$ need not be an axiom, but it will be deducible in one step by applying a substitution rule to α.

Q: Something bothers me about the exception in clause 5, saying that the clause doesn't apply to substitution rules. Your examples make it clear that such an exception is needed in typical logical systems. Nevertheless, it strikes me as awkward to have two sorts of rules treated differently, the substitution rules and all the rest. Is there a more conceptual way to understand the difference?

A: Yes, there is. Let's consider logical systems, where formulas have truth values (in appropriate structures). For simplicity of notation, let's consider a rule with a single premise, say the rule $\langle\{\alpha(x, y)\}, \beta(x, z)\rangle$, where $\alpha(x, y)$ just means a formula α in which only the variables x and y occur, and similarly for $\beta(x, z)$.

Now there are two possible semantical interpretations of such a rule. The first, which we'll call the weak interpretation, says that if the premise $\alpha(x, y)$ is universally true (as it would be if it has been proved) then so is the conclusion $\beta(x, z)$. Formally, the weak interpretation could be written as the implication

$$\big(\forall x \forall y\, \alpha(x, y)\big) \to \big(\forall x \forall z\, \beta(x, z)\big).$$

Q: Are you assuming that your Hilbertian system includes universal quantifiers?

A: Not necessarily. This implication is not being used here as a formula of the system but as an expression of the weak interpretation, using the ordinary means of logic.

The strong interpretation, in contrast, says that $\beta(x, z)$ should follow from $\alpha(x, y)$ for each assignment of values to x, y, z individually. Formally, it could be written as

$$\forall x \forall y \forall z \quad \big(\alpha(x, y) \to \beta(x, z)\big).$$

Q: I see that the strong interpretation implies the weak one, and the converse is not generally valid. So your terminology "strong" and "weak" is reasonable. But what does this have to do with my question about the exception in clause 5?

A: Substitution rules are, in general, sound only under the weak interpretation. For example, it is logically valid that

$$\big(\forall y\ P(y)\big) \to \big(\forall z\, P(z)\big),$$

but it is not valid that

$$\forall y \forall z \quad \big(P(y) \to P(z)\big).$$

On the other hand, it is only for the strong interpretation that we can be sure that the validity of a rule $\langle\{\alpha(x, y)\}, \beta(x, z)\rangle$ implies the validity of a substituted version $\langle\{\alpha(u, v)\}, \beta(u, w)\rangle$. So clause 5 makes good sense only for rules that are valid in the strong sense.

The implicit content of clause 5 is therefore that rules are supposed to be strongly valid except for the substitution rules, which are unavoidably only weakly valid.

Q: Why is this explanation left implicit in the actual definition of substitutional Hilbertian system? I think it would be clearer if it were stated explicitly.

A: The explanation depended on the assumption that formulas have meanings or at least truth values. The notion of substitutional Hilbertian system is more abstract. Clauses 4 and 5 are intended as an approximation, available at the abstract level, to the intuitive explanation at the more concrete level of logical formulas.

Q: OK, but I have another question. You said you were going to modify the treatment of abstract Hilbertian systems to apply to these substitutional Hilbertian systems. But the latter are, by definition, a special case of the former. So why do you need a modification; why not just apply the treatment of abstract Hilbertian systems directly to the substitutional case?

A: It is true that what we said about abstract Hilbertian systems in general applies in particular to substitutional ones. It produces a translation into propositional Horn logic. But that is not the translation we want in the substitutional case.

Substitutional Hilbertian systems admit a more useful translation into the universal Horn fragment of first-order logic. In this translation, the Horn theory $\hat{S}$ will not need axioms induced by the substitution rules, because the effect of those rules will be simulated by the substitutions available in universal Horn (or Datalog) inference. Furthermore, our translation will use, in first-order logic, a purely relational vocabulary (except for constant symbols).

Q: OK, but what's so great about universal Horn theories in purely relational vocabularies? Why do you want to translate into just such theories?

A: Computational efficiency. Universal Horn logic with no function symbols (except constants) is essentially the logic of Datalog, and much work has been done on building efficient Datalog software. By translating substitutional Hilbertian systems into Datalog, we can take advantage of this work to obtain more efficient implementations of appropriate substitutional Hilbertian systems.

Remark 14. In the application to DKAL, the target of the translation would, in general, be a version of Datalog with constraints. The reason is that deduction in a DKAL world is done by computational agents (usually called principals) that have states, and various sorts of information about the state can be used in the deductive process. Such information would be reflected in the Datalog approach in the form of constraints. The overall efficiency of deduction will, of course, depend on the efficiency with which these constraints can be evaluated.

The universal Horn translation of a substitutional Hilbertian system is defined very similarly to the (propositional) Horn translation $\hat{S}$ in the preceding section. The main differences are that, in place of the propositional variables [ξ is obtainable], we now have predicate symbols $|\xi|$ and that we do not include Horn clauses induced by the substitution rules.

Let S be a substitutional Hilbertian system. Its *universal Horn translation* $\tilde{S}$ is defined as follows. It has the same variables and constant symbols as S. It has a predicate symbol $|\alpha|$ for each formula α of S. If the distinct variables in α are $x_1, \dots, x_k$, in order of first occurrence in α, then the associated predicate symbol $|\alpha|$ is k-ary, and we say that an atomic formula $|\alpha|(u_1, \dots, u_k)$ of $\tilde{S}$ (where the u_i are terms, i.e., variables or constants) *points to* the formula $\alpha(u_1/x_1, \dots, u_k/x_k)$ of S. In particular, $|\alpha|(x_1, \dots, x_k)$ points to α, and we call $|\alpha|(x_1, \dots, x_k)$ the *standard translation* of α and denote it by $[\alpha]$. We emphasize, for future reference, that $[\alpha]$ points to α. With this terminology and notation, we can now list the axioms of $\tilde{S}$. They are of three sorts:

- If an atomic formula ξ (in the vocabulary of $\tilde{S}$) points to a formula α of S, then $\tilde{S}$ has the *bookkeeping* axioms

$$\xi \to [\alpha] \quad \text{and} \quad [\alpha] \to \xi.$$

- If α is an axiom of S, then $\tilde{S}$ has the corresponding axiom $[\alpha]$.
- If $\langle P, \gamma \rangle$ is a rule of S but not a substitution rule, then $\tilde{S}$ has the *induced* axiom

$$\left(\bigwedge_{\pi \in P} [\pi] \right) \to [\gamma].$$

Q: I see you've discarded the unnecessary words from the notation [α is obtainable].

A: Yes, but there's another reason, in addition to brevity, for changing the notation. In the proof of the "if" half of Theorem 4, we used a semantical interpretation in which a propositional variable [α is obtainable] meant that α is deducible in S from certain given hypotheses. In the corresponding argument for our new translation, the semantical situation will be somewhat different, and it seems helpful to indicate that difference with a new notation.

Q: What exactly is the difference?

A: Previously, when we dealt with propositional logic, the only atomic formulas were the propositional variables, [ξ is obtainable], whose meaning is suggested by the "is obtainable" notation. Now, in a first-order context, each predicate symbol $|\alpha|$ gives rise to lots of atomic formulas $|\alpha|(\mathbf{u})$, one of which is $[\alpha]$, namely $|\alpha|(\mathbf{x})$ (where $\mathbf{x}$ refers to the $x_1, \dots, x_k$ in the definition of universal Horn translation). If we retained the "is obtainable" notation, the general atomic formula would read [α is obtainable] ($\mathbf{u}/\mathbf{x}$), which is a bit awkward to read. Its intended meaning, as will be clear in the proof of Theorem 15 below, is that the formula to which it points, namely $\alpha(\mathbf{u}/\mathbf{x})$, is deducible.

It may be worth emphasizing that formulas of a substitutional Hilbertian system S (like any other Hilbertian system) do not necessarily represent assertions or have truth values. The formulas of the universal Horn translation $\tilde{S}$ do have meanings, and these concern only deducibility in S.

Theorem 15. *A formula γ is deducible from a set H of formulas in a substitutional Hilbertian system S if and only if $[\gamma]$ is deducible from $\{[\xi]\colon \xi \in H\}$ in the universal Horn theory $\tilde{S}$.*

Q: Before embarking on the proof, please remind me of the axioms and inference rules of universal Horn logic.

A: Recall (from Convention 1) that we care only about deductions where the conclusion is atomic. So we'll describe rules of inference for this situation. (For the more general case, where the conclusions can be non-atomic universal Horn formulas, see Appendix B at the end of the paper.)

Given a set H of universal Horn clauses, i.e., formulas of the form

$$\left(\bigwedge_{\pi\in P}\pi\right)\to\eta,$$

where all $\pi\in P$ and η are atomic, the set of atomic formulas deducible from H is defined to be the smallest set D with the following closure property:

- If, for some substitution σ (of constants or variables for variables) and some universal Horn clause (as above) in H, all the instances $\sigma(\pi)$ of the antecedent are in D, then so is the corresponding instance $\sigma(\eta)$ of the consequent.

The basis of this induction is given by clauses in H with empty antecedent, i.e., $\top\to\eta$, which we write as just η (without changing the meaning). When we speak, as in the theorem, of deducibility from some hypotheses in some system of universal Horn formulas, we mean that both the hypotheses and the axioms of the system are to be used as hypotheses in the definition of deducibility. When atomic formulas occur as hypotheses, they are to be treated as universal Horn clauses with the empty conjunction $\top$ as antecedent.

Q: These axioms and rules are obviously sound. I suppose they're known to be complete as well.

A: Yes, in fact, given H, one can construct a model M such that the atomic formulas that hold in M (when the free variables are interpreted as universally quantified) are exactly those deducible from H. To obtain M, temporarily introduce a countable infinity of new constants. Then interpret the predicate symbols by making a ground atomic formula true if and only if it is deducible from H.

Q: Why do you need the new constants here?

A: To provide counterexamples for unprovable atomic formulas that contain variables. Suppose, for example, that there were just two constant symbols and that H contains Pa and Pb, where P is a unary predicate. Then Px, with a variable x, is not deducible from H, but it holds in every model of H whose only elements are a and b.

Q: OK; I see why you need an additional element in your model, but why infinitely many? Couldn't a single new element serve as the counterexample for all the formulas that need one?

A: No. Suppose again that a and b are the only constants, but now suppose H contains Qax, Qbx, Qxa, Qxb, and Qxx for a binary predicate Q. Then Qxy is not deducible from H but holds in any model of H containing at most one element other than a and b. There are similar counterexamples, using an n-ary predicate, to show that you might need n elements beyond those named by the constants in H.

Q: Are relations of high arity the only reason for needing many new constants?

A: Yes. If all the predicates are at most n-ary, then n new constants suffice. To see this, consider a model as described above, where ground atomic formulas are true just in case they're deducible from H, and suppose we have an atomic formula α in the original vocabulary (so variables and original constants are allowed, but not the n new constants) that isn't deducible from H. Because of the arity bound, at most n variables occur in α, so we can consider a formula α' obtained from α by replacing all its variables by distinct fresh constants (from the n new constants that are available). This α' won't be deducible from H – any deduction of it could be converted to a deduction of α by changing the new constants back to variables (and perhaps renaming some other variables in the middle of the deduction). So, being an atomic ground formula, α' won't be true in our model; that is, the model contains a counterexample to α.

Lemma 16. *If an atomic formula γ is deducible in universal Horn logic from a set H of Horn clauses, then so is every substitution instance of γ.*

The lemma can be expressed by saying that substitution rules, though not officially among the rules of inference in Datalog, are admissible for Datalog.

Proof of Lemma 16. This is immediate from the soundness and completeness of the deductive system, but it can also be proved by a direct syntactic argument. The set of atomic formulas γ such that all their substitution instances are deducible from H is easily seen to enjoy the closure property in the definition of universal Horn deducibility. □

Proof of Theorem 15. For the "only if" direction, we proceed as in the proof of Theorem 4. We consider the set D of those formulas δ of the system S for which $[\delta]$ is deducible from $\{[\xi]\colon\ \xi\in H\}$ in $\tilde{S}$. As before, this set contains H and all the axioms of S (because of the corresponding axioms in $\tilde{S}$) and is closed under all rules of inference of S except substitution rules (because of the induced axioms of $\tilde{S}$). We must still check that D is also closed under substitution rules. Once this is done, it follows, from the definition of "deducible", that D contains all the formulas that are deducible in S from H, and so the "only if" proof will be complete.

Consider, therefore, a substitution rule of S, say $\langle\{\alpha\}, \alpha(\mathbf{u}/\mathbf{y})\rangle$, where $\mathbf{y}$ is a sequence of variables and $\mathbf{u}$ is the sequence of variables or constants being substituted for them. Suppose $\alpha \in D$; that is, $[\alpha]$ is deducible from $\{[\xi]\colon\ \xi \in H\}$ in $\tilde{S}$. Recall that $[\alpha]$ is $|\alpha|(\mathbf{x})$ where $\mathbf{x}$ is $x_1, \ldots, x_k$ as in the definition of $\tilde{S}$. We may assume, without loss of generality, that the sequences of variables $\mathbf{x}$ and $\mathbf{y}$ are the same. Indeed, any y_j that is not in $\mathbf{x}$ could be omitted (along with the corresponding u_j) from our substitution, since it has no effect there. If any x_i is not in the sequence $\mathbf{y}$, it could be appended to it, say as y_r, with $u_r = y_r$, since such a vacuous substitution has no effect. The only remaining possible difference between $\mathbf{x}$ and $\mathbf{y}$ would be the order of the terms, but we can permute $\mathbf{y}$ as long as we permute $\mathbf{u}$ the same way. So we have reduced the problem to showing that (with possibly modified $\mathbf{u}$), $\alpha(\mathbf{u}/\mathbf{x}) \in D$. That is, we must show that $[\alpha(\mathbf{u}/\mathbf{x})]$ is deducible from the hypotheses $\{[\xi]\colon\ \xi \in H\}$ in $\tilde{S}$. Since we already know that $[\alpha]$ is deducible from these hypotheses, Lemma 16 tells us that the formula $[\alpha](\mathbf{u}/\mathbf{x})$ is deducible as well. But this formula points to $\alpha(\mathbf{u}/\mathbf{x})$, so $\tilde{S}$ contains the bookkeeping axiom $[\alpha](\mathbf{u}/\mathbf{x}) \to [\alpha(\mathbf{u}/\mathbf{x})]$. Therefore, $[\alpha(\mathbf{u}/\mathbf{x})]$ is also deducible, as required, and the "only if" half of the theorem is proved.

For the "if" half, we prove the following stronger statement: If an atomic formula η is deducible in universal Horn logic from $\tilde{S}$ and $\{[\xi]\colon\ \xi \in H\}$, and if η points to the formula $\bar{\eta}$, then $\bar{\eta}$ is deducible from H in S. This includes the desired result because, as noted earlier, $[\gamma]$ points to γ.

Let E be the set of atomic formulas η with the desired property that the formula $\bar{\eta}$ to which they point is deducible from H in S. It suffices to show that E is closed under the inference rule of universal Horn logic with respect to the given hypotheses $\tilde{S}$ and $\{[\xi]\colon\ \xi \in H\}$. Suppose, therefore, that we have one of these hypotheses, say

$$\left(\bigwedge_{\kappa \in Q} \kappa\right) \to \lambda,$$

and a substitution σ such that all the instances $\sigma(\kappa)$ of the antecedent are in E; we must show that $\sigma(\lambda) \in E$. Using, as above, the bar notation to indicate the formula of S to which an atomic formula points, we have that each $\overline{\sigma(\kappa)}$ is deducible from H in S, and we must show the same for $\overline{\sigma(\lambda)}$. There are several cases, depending on the provenance of the hypothesis $(\bigwedge_{\kappa \in Q} \kappa) \to \lambda$.

The first possibility is that this hypothesis is $[\xi]$ for some $\xi \in H$. So Q is empty and λ is $[\xi]$. It follows immediately from the definitions that $\sigma(\lambda)$ points to $\sigma(\xi)$, so what we must show is that $\sigma(\xi)$ is deducible from H in S. But this is obvious; since $\xi \in H$, we obtain $\sigma(\xi)$ by one application of a substitution rule of S.

The second possibility is that the hypothesis is a bookkeeping rule of $\tilde{S}$. In this case, Q consists of a single atomic formula κ pointing to the same formula as λ does. Then $\sigma(\kappa)$ points to the same formula as $\sigma(\lambda)$. Thus, what we must prove is already covered by the induction hypothesis.

The third possibility is that the hypothesis is $[\alpha]$ for an axiom α of S. This case is handled exactly like the first case above.

The fourth and final possibility is that the hypothesis is the axiom induced by some rule R of S. In view of the definition of induced axioms, the rule R has as its premises the formulas $\bar{\kappa}$ for $\kappa \in Q$ and as its conclusion $\bar{\lambda}$. We know that R is not a substitution rule (because substitution rules don't induce axioms of $\tilde{S}$), and so, by clause 5 in the definition of substitutional Hilbertian system, S also has a rule of inference with premises $\sigma(\bar{\kappa})$ (for $\kappa \in Q$) and conclusion $\sigma(\bar{\lambda})$. By the definition of "points to", these premises are formulas $\overline{\sigma(\kappa)}$ that we already know are deducible from H in S, and the conclusion is $\overline{\sigma(\lambda)}$. So the latter is also deducible from H in S, as required. □

4. Complexity

Q: You've suggested that you're aiming for a computational benefit by translating Hilbertian systems with variables into Datalog. But the translations you actually produced, in Theorems 4 and 15, end up with infinitely many hypotheses in $\hat{S}$ or $\tilde{S}$, so I don't see any computational benefit, or indeed any relevance to computation.

A: What we have done so far is just setting up a general framework. It covers not only situations like the infon logic of [7], for which we expect a computational benefit, but also other systems whose worst-case behaviors are computationally hopeless. It is entirely possible to mirror, in universal Horn logic, the computations of arbitrary Turing machines.

To obtain computational benefits, one must analyze the system S and show that the translation $\tilde{S}$ can be limited in a way that precludes combinatorial explosions. Specifically, there are two problems that must be solved.

The first is that what looks like a simple rule in S, say modus ponens, is, from our viewpoint in the preceding sections, a rule schema, i.e., an infinite collection of rules, each inducing an axiom of $\tilde{S}$. For computational tractability, we need to bound the collection of axioms somehow, limiting Datalog's attention to the "relevant" ones.

The second is that universal Horn logic allows substitution of arbitrary constants for variables. If there are many constants available, these substitutions may be too numerous for practical computation.

The benefit of our translations, in particular $\tilde{S}$, is not that they immediately give us efficient algorithms but rather that they show us what must still be done in order to get such algorithms. The translations have, in essence, cleared away the underbrush and allowed us to focus on the essential properties of a substitutional Hilbertian system that can lead to efficient algorithms.

In the next section, we shall work out the particular example, primal infon logic with variables, that motivated the present work.

Note that Datalog has two built-in mechanisms to limit the second problem mentioned above, the excessive number of substitutions available in deductions. One mechanism is the prohibition of function symbols other than constants. Even a single, unary function symbol would produce, by iterated application to a constant, an infinite number of ground terms. By allowing no ground terms other than constants, Datalog at least keeps the number of available substitutions finite. (It is known that the notion of deducibility in universal Horn logic is unchanged if one allows in substitutions only the symbols occurring in the hypotheses and the conclusion.)

The other mechanism is the notion of safety, which means that any variable occurring in the head of a rule (i.e., in the consequent of a Horn clause) should also occur in the body (i.e., in the antecedent). When a Datalog program is safe in this sense, the usual way of evaluating Datalog queries will make fewer substitutions in any particular rule, because the substitutions into the body (limited by what has already been deduced) determine the substitutions into the head.

Unfortunately, the universal Horn translations of the various forms of primal logic are not safe in this sense,[11] so we shall need other methods to limit the substitutions. Those methods are the essential content of the next section.

Convention 17. We assume from now on that the systems we deal with (substitutional Hilbertian systems in general and Datalog in particular) are equipped with efficient ways to tell which symbols are variables, which are constants, and which are something else. This is needed for efficient implementation of substitution, which is taken for granted in Datalog.

5. Primal infon logic with variables

In this section, we apply the preceding results to a particular substitutional Hilbertian system, namely the primal infon logic of [7] extended to allow variables. Our purpose is to discuss the structure of this system and its universal Horn translation in enough detail to show how the threatened combinatorial explosions can be avoided.

We begin by recalling the relevant definitions of primal infon logic from [7, Section 5], while making the obvious modifications to allow (free) variables.

Atomic formulas are formed as usual from predicate symbols, a countably infinite supply of variables, and constants. There is also the propositional constant $\top$. Compound formulas are built from atomic formulas and $\top$ by the binary connectives of conjunction ($\wedge$) and implication ($\rightarrow$) and unary connectives "p `said`" and "p `implied`", where p can be a variable or constant. (For simplicity, we ignore typing of variables and constants; had we not ignored it, we should say that p is of type "principal".)

A *quotation prefix* is a finite concatenation of operators of the forms "p `said`" and "p `implied`", with possibly different p's. So by putting a quotation prefix in front of a formula one obtains again a formula. We use `pref`, sometimes with subscripts, to denote a quotation prefix. One quotation prefix, $\texttt{pref}_1$, is said to *dominate* another, $\texttt{pref}_2$, if they differ only in that some occurrences of `said` in $\texttt{pref}_1$ become `implied` in $\texttt{pref}_2$. (In particular, the two prefixes contain the same number of unary connectives, with the same p's in the same order.)

The only axioms of primal infon logic are $\texttt{pref}\,\top$ for arbitrary quotation prefixes `pref`.

The rules of inference are the substitution rules plus the following, for arbitrary formulas α and β and arbitrary quotation prefixes:

prefix deflation $\dfrac{\texttt{pref}_1\,\alpha}{\texttt{pref}_2\,\alpha}$ provided $\texttt{pref}_1$ dominates $\texttt{pref}_2$.

$\wedge$ *elimination* $\dfrac{\texttt{pref}(\alpha\wedge\beta)}{\texttt{pref}\,\alpha}$ and $\dfrac{\texttt{pref}(\alpha\wedge\beta)}{\texttt{pref}\,\beta}$.

$\wedge$ *introduction* $\dfrac{\texttt{pref}\,\alpha \qquad \texttt{pref}\,\beta}{\texttt{pref}(\alpha\wedge\beta)}$.

$\rightarrow$ *elimination* $\dfrac{\texttt{pref}\,\alpha \qquad \texttt{pref}(\alpha\rightarrow\beta)}{\texttt{pref}\,\beta}$.

$\rightarrow$ *introduction* $\dfrac{\texttt{pref}\,\beta}{\texttt{pref}(\alpha\rightarrow\beta)}$.

It is clear that this system of primal infon logic[12] with variables is a substitutional Hilbertian system. We call this system *PIV* (abbreviating "Primal-Infon-Variables") and, following the notation introduced earlier for arbitrary substitutional Hilbertian systems, we write $\widetilde{PIV}$ for its universal Horn translation satisfying Theorem 15. Our goal in this section is to

[11] The lack of safety arises from the implication introduction rule; see the next section.

[12] The full infon logic is formulated in a different style in [7]. Its most important difference from the primal logic is that it admits the familiar (non-Hilbertian) rule for implication introduction: When β has been deduced from α, infer $\alpha\rightarrow\beta$ and discharge the hypothesis α.

show that the notion of deducibility in *PIV* is unchanged by certain restrictions on the variables and formulas allowed in a deduction. These restrictions will permit, in specific applications, the replacement of $\widetilde{PIV}$ with a subsystem that is computationally efficient.

We concentrate, for the time being, on the *derivability problem* for *PIV*: Given a set H of formulas and another formula γ, is γ deducible from H in *PIV*? (The multiple derivability problem discussed, for the propositional case, in [7], can be treated similarly. It allows several γ's to be given and the problem is to decide which of them are deducible from the given H.) For given H and γ, we call a variable or constant u *native* if it occurs in H or in γ; otherwise u is *foreign*. Our first result about *PIV* is that, as far as the derivability problem is concerned, foreign variables and constants are irrelevant. The following theorem makes this precise and gives some additional simplifying information about the sorts of derivations that are needed. It implies, for example, that one can require all uses of substitution rules to precede all uses of other rules.

Theorem 18. *Let H be a set of formulas and γ a formula of PIV. Then γ is deducible from H in PIV if and only if there is a set H' of formulas such that*:

- *Each formula in H' is obtainable from some formula in H by substituting native terms* (*i.e., variables and/or constants*) *for some variables* (*possibly none*).
- *γ is deducible from H' in PIV without using substitution rules.*

Proof. The "if" direction is obvious, so we prove the "only if" direction. Let D be a tree deduction of γ from H in *PIV*. We shall modify D to obtain a tree deduction of the sort required by the theorem.

We begin by disposing of the case that there are no native variables or constants. In this case, we have no choice about H'; it can only be H itself. All the predicate symbols (if any) in H and γ must be 0-ary, but there might be predicate symbols of higher arity occurring in D, with (necessarily foreign) variables or constants as their arguments. If this occurs, then replace all atomic formulas involving such predicate symbols either with $\top$ or with any (necessarily 0-ary) predicate symbol from γ. The result is still a tree derivation; indeed, every step is still justified by the same rule as in D. Furthermore, the hypotheses H and conclusion γ are unchanged. That almost gives us the desired result with $H' = H$. The only remaining problem is that, if D used a substitution rule at some step, say inferring $\alpha(\mathbf{u}/\mathbf{x})$ from α, then in the new derivation the corresponding step is still officially justified by the same substitution rule, even though the formulas α and $\alpha(\mathbf{u}/\mathbf{x})$ have become identical and both $\mathbf{x}$ and $\mathbf{u}$ have disappeared. To get a derivation without substitution rules, as required by the theorem, just remove such repetitions from the tree deduction. (More pedantically: Replace the two identical formulas and the edge joining them by a single occurrence of that formula, maintaining the rest of the tree structure.)

From now on, we assume that there is at least one native variable or constant; we fix one and call it g (because it will be used as garbage). We consider various cases, depending on how the final conclusion γ was justified in D, and we proceed by induction (on the size of D), assuming as an induction hypothesis that the theorem is correct for the premises of the last inference in D.

Before we begin considering cases, we make two general remarks. First, the good news: It follows from clause 5 in the definition of substitutional Hilbertian systems (or, in the present case, from inspection of the axioms of *PIV*) that, whenever we have a tree derivation that doesn't use substitution rules, then we can uniformly replace variables by any other variables or constants, throughout the tree, and still have a tree derivation. In fact, for *PIV*, the situation is even better than clause 5 requires. We can also replace constants uniformly by variables or constants throughout a substitution-free derivation, and the result remains a derivation.

Second, the bad news: When we apply our induction hypothesis to a premise β of the final step in D, the relevant notion of "native" is "occurring in H or in β". We have β here, where we would prefer to have γ, so we shall have to include some steps to resolve the discrepancy.

We now begin to consider the possibilities for the final step in D. If γ is an axiom of *PIV* or a hypothesis from H, then we don't need to do anything (except to officially say that H' should be H). So from now on, we are concerned only with the case that there is a genuine step leading to γ at the root of D.

Suppose that step is prefix deflation. Then its premise β contains the same variables and constants as γ (they differ only in that γ may have "`implied`" at some places where β has "`said`"). So the induction hypothesis gives us an appropriate H' and a deduction, without substitution rules, from H' to β. Retaining the final prefix deflation, we get an appropriate deduction of γ.

The case of $\rightarrow$ introduction is equally easy, because here again every variable in the premise occurs also in the conclusion, and so the H' given by the induction hypothesis is as desired.

The same discussion applies to $\wedge$ introduction, except that now there are two premises. Apply the induction hypothesis to both.

The elimination rules are a bit more complicated. Consider the case of $\wedge$ elimination. Say γ is $\mathtt{pref}\alpha$, obtained from the premise $\mathtt{pref}(\alpha \wedge \xi)$. (The case of the other $\wedge$ elimination rule, giving the second conjunct, is exactly analogous.) Now the induction hypothesis gives us a substitution-free deduction D' of $\mathtt{pref}(\alpha \wedge \xi)$ from a set H' of substitution instances of H by native variables and constants. But these native symbols could include ones that occur only in ξ, not in H or γ, and these are not permitted in the deduction we want. Fortunately, they are easy to eliminate. Obtain D'' from D' by replacing all the unwanted variables or constants by the g that we fixed earlier. The result is a tree derivation (because there are no

substitution rules in D') of $\mathtt{pref}(\alpha \wedge \xi')$ (where ξ' is a substitution instance of ξ) from hypotheses H'' that are substitution instances of those in H'. Thus the formulas in H'' are obtainable from those in H by substitution, and in fact by substitution of native (to H and γ) variables or constants, because the foreign ones arising from ξ have been eliminated in favor of the native g. Thus, D'' followed by $\wedge$ elimination to produce γ is as required by the theorem.

The case of $\rightarrow$ elimination is handled the same way. Here γ is $\mathtt{pref}\,\alpha$ and it is obtained from the premises $\mathtt{pref}\,\xi$ and $\mathtt{pref}(\xi \rightarrow \alpha)$. In the new deductions of $\mathtt{pref}\,\xi$ and $\mathtt{pref}(\xi \rightarrow \alpha)$ provided by the induction hypothesis, replace all the variables and constants foreign to H and γ (hence arising from ξ) with g. The resulting two tree deductions, followed by $\rightarrow$ elimination to produce γ, witness the theorem in this case.

Finally, there is the case that γ is obtained from some β by substitution; say γ is $\beta(\mathbf{u}/\mathbf{x})$. We may assume that each variable in the sequence $\mathbf{x}$ actually occurs in β and therefore each term in $\mathbf{u}$ occurs in γ, because any x_i not occurring in β could simply be omitted from the substitution operation, along with the corresponding u_i, without affecting the result γ. So each term u_i in $\mathbf{u}$ is native for H and γ, but x_i might not be. The induction hypothesis gives us a substitution-free tree deduction D' of β from hypotheses H' obtained from H by native substitution – where "native" refers to β and may therefore include variables from $\mathbf{x}$. Obtain D'' from D' by replacing each x_i everywhere by the corresponding u_i. This D'' is still a tree deduction, because D' didn't use substitution. Its hypotheses H'' are like H' except that $\mathbf{x}$ is no longer there, having been changed to $\mathbf{u}$; so H'' is obtainable from the original H by substitutions native for H and γ. Finally, the conclusion of D'' is $\beta(\mathbf{u}/\mathbf{x})$, i.e., the desired γ. □

Remark 19. The proof of Theorem 18 applies to many substitutional Hilbertian systems, not just to *PIV*. The only special properties of *PIV* that were used in the proof are the availability of $\top$ or some 0-ary predicate symbol and the possibility of renaming constants in a deduction.

The proof used $\top$ or a 0-ary predicate symbol from γ in the degenerate case where there are no native variables or constants. The essential property that is needed here is that any occurrence of a variable or constant in a formula is part of a segment (an atomic subformula) that can be replaced by $\top$ or a 0-ary predicate symbol, resulting in a formula, and that systematic replacement of all such segments in the premises and conclusion of any rule results in a rule.

The proof also used, in the case of elimination rules, the fact that one can uniformly replace constants by constants or variables in any rule and obtain again a rule. In effect, substitution-free derivations treat constants the same as variables.

Any substitutional Hilbertian system that shares the two essential properties just described will also share the property of *PIV* asserted in Theorem 18.

Remark 20. A universal Horn clause [or "rule", in Datalog terminology] $(\bigwedge X) \rightarrow \eta$ is *safe* if every variable occurring in the conclusion [or "head"] η also occurs in the antecedent [or "body"] $\bigwedge X$. A set of Datalog rules [a "program"] is safe if all its rules are so. When Datalog computes the consequences of given rules by forward inference, it considers an instantiation of a rule only when the corresponding instantiations of the atomic formulas in its body have already been established. Safety implies that, for any such instantiation of the body, only one instantiation of the head is generated. Thus safety provides a measure of control over the substitutions used during such a computation.

The safety condition need not hold in instances of the *PIV* derivability problem, nor in the Datalog programs obtained by the translation into $\widetilde{PIV}$. Instead, we obtain a similar measure of control over substitutions by Theorem 18. This theorem assures us that we need only substitute native terms for variables. So it provides an *a priori* bound on what can be substituted for variables.

Q: By using only native variables and constants in forming H', this theorem addresses the second of the two problems you mentioned in connection with the combinatorial explosion. Datalog won't have to substitute arbitrary variables and constants, only those native to the query. But what about the first problem? Modus ponens, now renamed "$\rightarrow$ elimination", is still an infinite collection of rules, and nothing in your theorem tells us that a reasonable subcollection will suffice.

A: There is indeed nothing about that issue in this theorem, but there is something in [7]. It is shown there that, when a formula γ is derivable from a set H of formulas in primal infon logic, then there is a deduction that uses only formulas local to the query. This notion of locality is very similar to the traditional notion of subformula, except that some manipulation of quotation prefixes is also allowed. In any case, given a query, consisting of H and γ, the locality theorem [7, Theorem 5.11] puts useful bounds on which rules are actually needed for derivations.

Q: That sounds good, but [7] is only about the propositional case. How do you know that locality applies to *PIV*, where variables are allowed and there is a substitution rule.

A: That's where the "substitution-free" part of the conclusion of our Theorem 18 comes in (in addition to being used in the proof to make the induction work). The deduction of γ from H', since it doesn't use substitution rules, amounts to a deduction in *propositional* primal infon logic. We simply view all the atomic formulas as propositional variables; all the rules of *PIV* except substitution are rules of propositional primal infon logic. Thus, the locality theorem from [7] applies to this deduction.

In fact, we can import some more information from [7]. Theorem 5.8 of that paper (in combination with the already cited Theorem 5.11) lets us convert any deduction in propositional primal infon logic into one that splits into two parts:

First there is a segment involving only formulas local to the hypotheses. Second, from the conclusions of this local part, one obtains the desired conclusion by means of only introduction rules. As before, this result for propositional primal infon logic carries over immediately to the substitution-free part of deduction in *PIV*.

Note that, if the conclusion is an atomic formula, then the second part, consisting of introduction rules, can be taken to be vacuous. So an atomic conclusion must be local to the hypotheses, and then the whole deduction can be made local to the hypotheses.

Here's the result we obtain, for *PIV*, in the light of this information.

Corollary 21. *If γ is deducible from H in PIV, then there is a set H' of formulas, each obtainable from a formula in H by a native (to H and γ) substitution, such that γ can be deduced from H' by a deduction that uses no substitution rules and uses only formulas local to $H' \cup \{\gamma\}$. If γ is atomic, then it must be local to H', and the whole deduction of γ from H' uses only formulas local to H'. If γ is not atomic, then its deduction from H' can be taken to be, first, a part consisting of formulas local to H', and second, a part building up γ from these local formulas by means of introduction rules.*

Q: There might be a great many substitution instances in H' even if there are only a few formulas in H. Can you cut H' down to something manageable?

A: In many cases, H' can indeed be reduced, but manageability depends on the particular problem. Here's one result in this direction.

Corollary 22. *In the situation of Corollary 21, assume in addition either that at least one variable occurs in γ or that at least one constant occurs in H or in γ. Then the conclusion of Corollary 21 remains true with the additional requirement that every variable in H' also occurs in γ. In particular, if γ is a ground formula (i.e., without variables), then H' can be taken to consist of ground formulas.*

Proof. Let g be either a variable occurring in γ or a constant occurring in H or in γ. Fix a deduction D of γ from some H' as in Corollary 21. In this deduction D, put g in place of all occurrences of all variables that are not in γ. The result is still a deduction (as D didn't use substitution rules), and its conclusion is still γ (as we replaced only the variables that didn't occur in γ). The hypotheses H' have been changed by replacing variables not in γ with g, so they now satisfy the desired conclusion. (Note that not only the new hypotheses H' but the whole new deduction of γ from them uses only variables from γ.) □

6. From primal infon logic to Datalog

We combine the preceding results with an additional simplification and with observations from [7] to obtain the following reduction of the derivability problem of *PIV* to the same problem for Datalog.

Theorem 23. *There is an algorithm that converts any instance of the derivability problem for PIV into an instance of the derivability problem for Datalog with the same answer. Furthermore, when this algorithm is applied to PIV formulas of bounded quotation depth, the size of the resulting Datalog problem is quadratically bounded in terms of the size of the PIV problem.*

Proof. The basic idea of the proof is to translate the *PIV* derivability problem "Is γ deducible from H?" to the Datalog derivability problem "Is $[\gamma]$ deducible from $\widetilde{PIV}$ and $\{[\xi]\colon\ \xi \in H\}$?" The difficulty is that the output here, specifically the $\widetilde{PIV}$ part, is infinite.

Part of this difficulty is removed by Corollary 21. For each non-substitution rule schema of *PIV*, contributing infinitely many individual rules to *PIV* and thus inducing infinitely many axioms of $\widetilde{PIV}$, we need only those individual rules whose premises and conclusion are local to H and γ.

Q: A prime seems to have gotten lost. Corollary 21 speaks about formulas local to H' and γ, but now you want to use formulas local to H and γ.

A: Fortunately, every formula local to H' and γ is a substitution instance of a formula local to H and γ. So the Datalog program corresponding to Corollary 21, with formulas local to H' and γ, can be obtained from the H-and-γ version by substitution. And the way Datalog deduction is defined, any step justified by substitution instances of a universal Horn formula is equally justified by that formula itself.

It is pointed out in [7] that the number of local formulas is linearly bounded in terms of the size of H and γ as long as the quotation depth is bounded. It follows that each rule schema contributes only a linearly bounded number of induced axioms to the local part of $\widetilde{PIV}$.

The remaining part of the problem with $\widetilde{PIV}$ concerns the bookkeeping axioms. Our definitions allow far too many of these, because of the possibility of many different tuples of arguments for a predicate symbol $|\alpha|$. Fortunately, we can eliminate the problem by being more frugal with our predicate symbols.

Given H and γ, call two local formulas *similar* if they become equal when a particular variable, say z, is substituted for all variables and all constants. Another way to say this is that the two formulas agree, symbol by symbol, except that they might have different variables or constants in corresponding positions. For each similarity class, one can obtain a formula (not necessarily local) ξ such that each formula in the similarity class is a substitution instance of ξ. One general way to do this is to let ξ agree, symbol for symbol, with some α in the similarity class, except that in all the places where α has variables or constants, ξ has distinct variables. (It may be, however, that ξ can use the same variable at some pair of positions, namely if no α in the similarity class has different variables or constants there. It may also happen that ξ can use a constant, if all formulas in the similarity class have the same constant at that position.) Choose one such ξ for each similarity class of local formulas and call it the *generic form* for that class. Then introduce predicates $|\xi|$ not for all local formulas but rather for these generic forms. If α is a local formula, obtained from the generic form ξ of its similarity class by replacing the variables $x_1, \ldots, x_k$ of ξ (in order of first occurrence) by $u_1, \ldots, u_k$, then define $[\alpha]$ to be $|\xi|(u_1, \ldots, u_k)$.

With this revised definition of $[\alpha]$, we no longer need bookkeeping axioms. If η, in the new, smaller vocabulary of $\widetilde{PIV}$, points to α then η is $[\alpha]$, so we no longer need axioms to say that they are equivalent.

These modifications produce a translation from the *PIV* derivability problem into the Datalog derivability problem, that behaves well with respect to the size of the instances. The number of formulas in the output (an instance of Datalog derivability) is linear with respect to the size of the input (an instance of *PIV* derivability). Since the relevant Datalog formulas are, by inspection of the translation, combinations of at most three atomic formulas, we still have a linear bound if we measure the size of the output as the number of atomic subformulas or as the number of connectives. Atomic formulas can, however, contain many symbols because the predicates $|\alpha|$ can have many arguments – as many as in any formula in the input. So these arities are, in general, only linearly bounded, and we get a quadratic bound for the size of the output measured in symbols. □

Q: In this proof, you've made some modifications to the original method, direct translation to Datalog, because that method, though rather easy to describe, was inappropriate for computation. Could you summarize how the new method, after all the modifications, works?

A: Sure. Let's restrict attention to the important case where γ is atomic. Given an instance of the derivability problem for *PIV*, say the instance "Is γ deducible from H?" where γ is atomic, here's what you do. First, using the definitions from [7], make a list of the local formulas with respect to H. Check which of them are similar, in the sense of the proof of Theorem 23. As in that proof, choose a generic form for each similarity class, and invent predicate symbols to correspond to those generic forms. That determines the formulas $[\alpha]$ for all your local formulas α. Now form a set of universal Horn formulas (to be used as Datalog rules) as follows.

- For each local axiom $\mathtt{pref}\,\top$ of *PIV*, include $[\mathtt{pref}\,\top]$.
- For each rule schema of *PIV*, include the axioms

$$\left(\bigwedge_{\pi\in P}[\pi]\right)\to[\alpha]$$

  generated by those rules of the schema that consist entirely of formulas local to H (i.e., α and all $\pi \in P$ are local to H).
- For each hypothesis $\eta \in H$, include $[\eta]$.

(Note that there are no bookkeeping axioms here.) Finally ask Datalog whether $[\gamma]$ is deducible from this set of rules.

Q: I like that the resulting Datalog program depends only on H.

A: Right, it is often the case that H is fixed while γ varies.

Q: But what if γ is not atomic?

A: Then, for each introduction rule schema of *PIV*, include the axioms

$$\left(\bigwedge_{\pi\in P}[\pi]\right)\to[\alpha]$$

generated by those rules of the schema that consist entirely of formulas local to γ.

Appendix A. Primal infon logic subsumes Datalog

In this appendix, we explain briefly a limitation on any attempt to solve efficiently the derivability problem for *PIV*, namely that the worst-case complexity of this problem is exponential-time hard. The reason is a combination of two facts: First, it is exponential-time hard to determine, given a Datalog program and a ground atom, whether the atom is derivable from the program; this is Theorem 4.5 of [4]. Second, this Datalog derivability problem can easily be reduced to the *PIV* derivability problem, as we now explain.

The reduction consists of replacing each rule in a Datalog program with the universal Horn formula expressing it, and then deleting the universal quantifiers. Thus, a rule

$$\alpha :- \beta_1, \ldots, \beta_n$$

becomes the *PIV* formula, which we call the implicational form of the rule,

$$(\beta_1 \wedge \cdots \wedge \beta_n) \to \alpha$$

(with some standard choice of parenthesization of the iterated conjunction). The reduction leaves the goal, the ground atom in the Datalog derivability problem, unchanged. We claim that this reduction is correct, i.e., that a ground atom γ is deducible in Datalog from certain rules if and only if it is deducible in *PIV* from the implicational forms of those rules.

The "only if" direction is easy, because every step in a Datalog derivation can be simulated in *PIV* by means of substitution rules (to produce the implicational forms of the instances of rules used by Datalog), $\wedge$ introduction (to assemble the instances of the body of a rule), and $\to$ elimination (to infer the instance of the head of a rule).

For the "if" direction, the point is to show that the additional apparatus available in *PIV* (quotation prefixes and the axioms and rules not used in the preceding paragraph) are conservative, i.e., that no ground atoms become deducible with this apparatus that were not already deducible by the method of the preceding paragraph. For this purpose, we adopt a semantical approach, as follows. Consider an arbitrary first-order structure for the vocabulary of an instance of the derivability problem of Datalog or of *PIV*. Interpret *PIV* formulas in such a structure by interpreting atomic formulas and propositional connectives ($\wedge, \to, \top$) in the standard way, ignoring quotation prefixes, and interpreting all variables as universally quantified (with the entire formula as the scope of the universal quantifier). Interpret Datalog rules as synonymous with their implicational forms. It is well known that Datalog is complete for this semantics, in the sense that, if a ground atom holds in all structures that satisfy certain rules, then that atom is deducible from those rules. Furthermore, it is easy to check that *PIV* is sound for this semantics, i.e., anything deducible in *PIV* from some hypotheses is satisfied in all structures that satisfy those hypotheses. Combining the preceding two sentences, we find that, if a ground atom is deducible in *PIV* from the implicational forms of certain Datalog rules, then it is semantically a consequence of those implicational forms and therefore of those rules, and then it is deducible in Datalog from those rules. This completes the verification of the correctness of our reduction of Datalog to *PIV*.

Appendix B. Universal Horn logic

In accordance with Convention 1, our treatment of universal Horn logic was restricted to deducing atomic formulas from universal Horn formulas. This part of the logic is what Datalog uses, and it suffices for our application to the derivability problem for substitutional Hilbertian systems. For the sake of completeness, we summarize in this appendix some information about the general version of universal Horn logic, where conclusions are not required to be atomic. So we repeal Convention 1; we are now concerned with deducing universal Horn conclusions from universal Horn hypotheses. As before, we do not write universal quantifiers; free variables are always to be interpreted as universally quantified.

If we do not insist on a substitutional Hilbertian deductive system but allow the use of temporary hypotheses, then this general version of universal Horn logic can be reduced to the Datalog case, with atomic conclusions, considered earlier. Specifically, a universal Horn formula φ of the form

$$\left(\bigwedge_{\pi \in P} \pi \right) \to \eta$$

is deducible from a set H of universal Horn hypotheses if and only if for some (or, equivalently, for every) substitution σ that substitutes new, distinct constants for the free variables in φ, the atomic formula $\sigma(\eta)$ is deducible from $H \cup \{\sigma(\pi) \colon \pi \in P\}$.

If, on the other hand, we want a substitutional Hilbertian formulation of universal Horn logic, then the following seems to be the simplest and most natural such system.

The only axioms are $\alpha \to \alpha$ for atomic formulas α.

The rules of inference are of three sorts:

- Substitution: From φ infer $\varphi(\mathbf{u}/\mathbf{x})$ for any list of variables $\mathbf{x}$ and any variables or constants[13] $\mathbf{u}$.
- Weakening: From $(\bigwedge X) \to \gamma$ infer $(\bigwedge Y) \to \gamma$ when $X \subseteq Y$.
- Cut: From $(\bigwedge X) \to \gamma$ and $(\bigwedge Y) \to \xi$ for all $\xi \in X$, infer $(\bigwedge Y) \to \gamma$.

By writing the antecedent in a Horn clause as the conjunction $\bigwedge X$ of a set of atomic formulas, we've implicitly built in the convention that the order of atomic formulas in the antecedent doesn't matter. Had we used conjunctions of sequences, rather than sets, of atomic formulas, then the rules should allow us to permute these atomic formulas and to add or delete repetitions, either by additional explicit rules or by a suitable formulation of cut.

[13] If we had non-constant function symbols, then arbitrary terms should be allowed in the list $\mathbf{u}$.

When this general version of universal Horn logic is used with the universal Horn translation $\tilde{S}$ of a substitutional Hilbertian system S, it produces derived rules of inference for S. More precisely, consider a Horn clause φ in the vocabulary of $\tilde{S}$, say

$$\left(\bigwedge_{\beta\in B}\beta\right)\to\eta$$

where η and the elements β of the body B are atomic formulas. Write $\bar{\eta}$ for the formula of S to which η points, and analogously for $\bar{\beta}$. ("Points to" was defined during the definition of $\tilde{S}$ shortly before Theorem 15.) We say that the Horn clause φ is a *strong derived rule* for the system S with a set H of S-formulas as hypotheses if and only if $\bar{\eta}$ is deducible from $\{\bar{\beta}\colon\ \beta\in B\}$ and substitution instances of formulas from $H\cup\mathrm{Ax}$ (where Ax is, as before, the set of axioms of S) using only the non-substitution rules of S.

Q: What would be a weak derived rule?

A: Well, the usual meaning of "derived rule", in a context like this, would be that $\bar{\eta}$ is deducible from $\{\bar{\beta}\colon\ \beta\in B\}\cup H\cup\mathrm{Ax}$ via the inference rules of S. We added the adjective "strong" because of the additional requirements that the deduction should use substitution only on elements of $H\cup\mathrm{Ax}$ and should do so before applying any of the non-substitution rules.

Proposition 24. *A Horn clause φ is deducible from $\{[\xi]\colon\ \xi\in H\}$ in $\tilde{S}$ if and only if φ is a strong derived rule for S with hypotheses H.*

Proof. We begin by proving the "only if" direction. It suffices to show that the set D of Horn clauses that are strong derived rules for S with H contains all the assumptions $[\xi]$ for $\xi\in H$, that it contains all the axioms of $\tilde{S}$ and of universal Horn logic, and that it is closed under the deduction rules of universal Horn logic.

Many of these items are trivial. In the first place, if $\xi\in H$, then $[\xi]$ is in D because it points to ξ (which is a substitution instance of itself).

Consider next the axioms of $\tilde{S}$. The bookkeeping axioms are in D because in such an axiom the antecedent and the consequent point to the same formula of S.

The axioms $[\alpha]$ corresponding to axioms α of S are clearly in D because α is a (trivial) substitution instance of itself.

Finally, the axiom of $\tilde{S}$ induced by a non-substitution rule $\langle P,\alpha\rangle$ of S is in D, by one application of that very rule $\langle P,\alpha\rangle$.

We turn next to the axioms and rules of universal Horn logic. The axioms $\alpha\to\alpha$ are clearly in D, so we need only consider the rules. The case of weakening is trivial, because we don't lose deducibility when more hypotheses become available (i.e., the system S is a monotone logic). The case of cut is also trivial, as we can just combine the given deductions to produce the desired one.

It remains to consider the substitution rules. It suffices to consider, for notational simplicity, substitution for a single variable, because multiple substitutions can be obtained by iterating single substitutions. Suppose therefore that φ is in D, that x is a variable, and that u is a variable or constant; we must show that $\varphi(u/x)$ is in D. We use the notation from the definition of "strong derived rule", so φ is $(\bigwedge_{\beta\in B}\beta)\to\eta$ and therefore $\varphi(u/x)$ is $(\bigwedge_{\beta\in B}\beta(u/x))\to\eta(u/x)$. As we already saw in another context, $\overline{\eta(u/x)}$ is the same as $\bar{\eta}(u/x)$, and similarly for each $\beta\in B$.

Our assumption that φ is in D means that we have a deduction, say Δ, of $\bar{\eta}$ from $\{\bar{\beta}\colon\ \beta\in B\}$ plus substitution instances of formulas from $H\cup\mathrm{Ax}$ using the non-substitution rules of S. We must show that we also have a deduction Δ' of $\bar{\eta}(u/x)$ from $\{\bar{\beta}(u/x)\colon\ \beta\in B\}$ plus substitution instances of formulas from $H\cup\mathrm{Ax}$ using the non-substitution rules of S. We obtain Δ' from Δ by replacing every occurrence of x by u. Notice that the conclusion $\bar{\eta}$ of Δ becomes the desired conclusion $\bar{\eta}(u/x)$ in Δ'. The hypotheses $\bar{\beta}$ $(\beta\in B)$ used in Δ become legitimate hypotheses $\bar{\beta}(u/x)$ for Δ'. Any substitution instances of formulas from $H\cup\mathrm{Ax}$ that were used as hypotheses in Δ become in Δ' (possibly different) substitution instances of the same formulas. And the rules of inference used in Δ were not substitution rules, so they remain correct in Δ' because of clause 5 in the definition of substitutional Hilbertian systems.

This completes the proof of the "only if" half of the theorem. We turn to the "if" half.

We first claim that it suffices to prove:

$(*)$ if α is deducible, by non-substitution rules of S, from a set Q of formulas plus substitution instances of $H\cup\mathrm{Ax}$, then $(\bigwedge_{\kappa\in Q}[\kappa])\to[\alpha]$ is deducible in universal Horn logic from $\tilde{S}$ plus $\{[\xi]\colon\ \xi\in H\}$.

To see that this suffices, suppose $(\bigwedge_{\beta\in B}\beta)\to\eta$ is a strong derived rule for S and H. Take Q to be $\{[\bar{\beta}]\colon\ \beta\in B\}$, take α to be $[\bar{\eta}]$, and notice that, by definition of "strong derived rule", they satisfy the hypothesis of $(*)$. Therefore, by $(*)$, we have a deduction from $\tilde{S}$ plus $\{[\xi]\colon\ \xi\in H\}$ of

$$\left(\bigwedge_{\beta\in B}[\bar{\beta}]\right)\to[\bar{\eta}].$$

But $\tilde{S}$ also has the bookkeeping axioms $\beta\to[\bar{\beta}]$ and $[\bar{\eta}]\to\eta$. So two applications of the cut rule produce a deduction of

$$\left(\bigwedge_{\beta\in B}\beta\right)\to\eta,$$

as required.

We therefore turn our attention to proving $(*)$, and we proceed by induction on the deduction of α. There are several cases to consider, none particularly difficult.

Suppose first that $\alpha\in Q$. Then the required $(\bigwedge_{\kappa\in Q}[\kappa])\to[\alpha]$ follows from the axiom $[\alpha]\to[\alpha]$ by weakening.

Second, suppose α is a substitution instance of an axiom of S; say $\zeta\in \mathrm{Ax}$ and α is $\sigma(\zeta)$ for some substitution σ. Then in $\tilde{S}$, we have the axiom $[\zeta]$, and we can deduce, by a substitution rule, $\sigma([\zeta])$, which points to α. So we also have the bookkeeping axiom $\sigma([\zeta])\to[\alpha]$, and we obtain $[\alpha]$ by cut. Finally, we obtain the required $(\bigwedge_{\kappa\in Q}[\kappa])\to[\alpha]$ by weakening.

The third case is that α is a substitution instance of an element ξ of H. This case is handled exactly like the preceding one except that, instead of the axiom $[\zeta]$ of $\tilde{S}$, we use the hypothesis $[\xi]$.

Finally, there remains the case that α was obtained by a non-substitution rule of S, say $\langle P,\alpha\rangle$, from a set P of premises. By induction hypothesis, there are deductions in universal Horn logic, from $\tilde{S}$ and $\{[\xi]\colon\ \xi\in H\}$, of

$$\left(\bigwedge_{\kappa\in Q}[\kappa]\right)\to[\pi]$$

for all $\pi\in P$. We also have in $\tilde{S}$ the axiom induced by the rule $\langle P,\alpha\rangle$, namely

$$\left(\bigwedge_{\pi\in P}[\pi]\right)\to[\alpha].$$

Combining these by the cut rule, we get the required

$$\left(\bigwedge_{\kappa\in Q}[\kappa]\right)\to[\alpha].\qquad\square$$

References

[1] Nikolaj Bjørner, Guido de Caso, Yuri Gurevich, From primal infon logic with individual variables to Datalog, in: E. Erdem, J. Lee, Y. Lierler, D. Pearce (Eds.), Correct Reasoning: Essays on Logic-Based AI in Honour of Vladimir Lifschitz, in: Lecture Notes in Computer Science, vol. 7265, Springer, 2012, pp. 72–86.

[2] Andreas Blass, Yuri Gurevich, Michał Moskal, Itay Neeman, Evidential authorization, Microsoft Research Technical Report MSR-TR-2010-92, July 2010, Sebastian Nanz (Ed.), The Future of Software Engineering, Springer, 2011, pp. 77–99.

[3] DKAL at CodePlex, http://dkal.codeplex.com/.

[4] Evgeny Dantzin, Thomas Eiter, George Gottlob, Andrei Voronkov, Complexity and expressive power of logic programming, ACM Computing Surveys 33 (3) (2001) 374–425.

[5] William Dowling, Jean Gallier, Linear-time algorithms for testing the satisfiability of propositional Horn formulae, The Journal of Logic Programming 1 (1984) 267–284.

[6] Yuri Gurevich, Two notes on propositional primal logic, Microsoft Research Technical Report MSR-TR-2011-70, May 2011.

[7] Yuri Gurevich, Itay Neeman, Logic of infons: The propositional case, ACM Transactions on Computational Logic 12 (2) (January 2011), Article 9, Microsoft Research Technical Report MSR-TR-2011-90, July 2011, corrects an oversight in the published version.

[8] Alon Itai, Janos Makowsky, On the complexity of Herbrand's theorem, Technical Report 243, Dept. of Computer Science, Technion – Israel Institute of Technology, May 1982, later published as Unification as a complexity measure for logic programming, The Journal of Logic Programming 4 (1987) 105–117.

[9] Stephen C. Kleene, Introduction to Metamathematics, Van Nostrand, 1952.

[10] Grigori Mints, A Short Introduction to Intuitionistic Logic, Kluwer/Plenum, 2000.

[11] Michel Minoux, LTUR: A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation, Information Processing Letters 29 (1) (1988) 1–12.