Clausal Logic and Logic Programming in Algebraic Domains*

William C. Rounds

Artificial Intelligence Laboratory
University of Michigan
Ann Arbor, MI 48109, U.S.A.
E-mail: rounds@umich.edu

and

Guo-Qiang Zhang

Department of Electrical Engineering and Computer Science
Case Western Reserve University
Cleveland, OH 44106, U. S. A.
E-mail: gqz@eecs.cwru.edu

We introduce a domain-theoretic foundation for disjunctive logic programming. This foundation is built on clausal logic, a representation of the Smyth powerdomain of any coherent algebraic dcpo. We establish the completeness of a resolution rule for inference in such a clausal logic; we introduce a natural declarative semantics and a fixed-point semantics for disjunctive logic programs, and prove their equivalence; finally, we apply our results to give both a syntax and semantics for default logic in any coherent algebraic dcpo.

Key Words: Domain theory and applications, logic programming, logics in artificial intelligence.

1. INTRODUCTION

Domain theory, as introduced by Scott in the 1970's, has many connections with logic. Such connections are usually made by extracting an appropriate *language/syntax* from a category of domains. To name a few examples, we have Abramsky's "domain theory in logical form" [Abr91], Scott's own representation of Scott domains as information systems [Sco82], extended to other domains by Zhang [Zha91], and Smyth's treatment of observable properties as open sets [Smy83], expanded to the ideas of topological systems and locales in Vickers [Vic89]. There has been much investigation into the use of domain logics as

* Research supported by NSF grant IRI-9509067.

logics of types and of program correctness, mainly for imperative and functional languages. Surprisingly, though, there has been relatively little work tying domain theory to the semantics of logic programming [Apt90], in particular disjunctive logic programming [LMR92]. Most of this work focuses on first-order logic. Extensions to higher types have been made — for example, λ-Prolog [NM98] — but domain theory has not played as much of a role here as it has for functional programming.

In this paper we offer some potentially appropriate spaces — coherent algebraic domains — for the semantics of disjunctive logic programs, perhaps even of higher type, though much work would be required to substantiate this. Our idea is that disjunctive logic programs can be seen as non-deterministic programs in the standard sense of non-deterministic computation, and not so much as declarative representations. For this purpose we use the *Smyth powerdomain* construction [Smy78].

In this paper we show how to represent the Smyth powerdomain of a coherent algebraic dcpo using an elementary logic built over such a domain. This is a *clausal* logic, different from the modal logic introduced by Winskel [Win83] for the Smyth powerdomain. We obtain the logic by regarding finite sets of compact elements of a domain disjunctively as clauses of an abstract partial logic, and sets of clauses conjunctively as theories. We prove our representation theorem (Theorem 3.2) using the Hofmann-Mislove theorem [HM81]. This proof makes clear the basic Galois connection (duality) between theories in the clausal logic, and sets of models as Scott-compact saturated sets. The main result yields a compactness theorem for any clausal logic over a coherent algebraic domain. We prove the usual compactness theorem in classical logic as a corollary.

Next we show that the resolution rule, appropriately generalized to clauses over coherent algebraic dcpos, gives a complete inference procedure for the clausal logic. We apply these results to give a fixed-point semantics for abstract disjunctive logic programs. We then establish the agreement between this semantics and the natural declarative semantics (i.e. the set of logical consequences).

Our research into these questions was initiated by using domain theory to provide a model for non-monotonic reasoning in AI, as exemplified by Reiter's *default logic* [Rei80]. We used the Smyth powerdomain, in its model-theoretic incarnation as a domain of compact saturated sets, as a setting for a semantic version of default reasoning called *power default reasoning* [ZR97a]. In the final section of this paper, we provide a simple syntactic version of default logic whose semantics is exactly the semantics of power defaults. This logic is derived naturally from disjunctive logic programs, and its proof-theoretic semantics is essentially that of the fixed-point semantics of disjunctive logic programs, modified to include a so-called *consistency check.*

The paper is organized as follows. Section 2 contains preliminary definitions from domain theory. For those readers unfamiliar with these concepts we use the running example of Kleene's three-valued propositional logic with strong negation. The representation result is the subject of Section 3. Section 4 is a study of the resolution rule, and Section 5 treats logic programs. Finally, Section 6 gives applications to default logic.

2. PRELIMINARIES

This section contains standard definitions from domain theory. For those readers unfamiliar with the theory, we provide the running example of Kleene's three-valued propositional logic. For our compactness results we use the Hofmann-Mislove theorem. A lemma providing the essence of the proof of this theorem can be found in Keimel and Paseka [KP94]; see also Vickers [Vic89].

DEFINITION 2.1. *A* poset *is a pair* $(D, \sqsubseteq)$*, where* D *is a nonempty set and* $\sqsubseteq$ *is a reflexive, antisymmetric, and transitive relation on* D*. If* D *has a least element we write this as* $\bot$*. When* $x \sqsubseteq y$ *we sometimes say "*x *subsumes* y*".*

DEFINITION 2.2. *A nonempty subset* X *of a poset* D *is* directed *iff for any* x, y *in* X *there is a* z *in* X *with* $x \sqsubseteq z$ *and* $y \sqsubseteq z$*.*

A typical example of a directed set is a nonempty totally ordered subset (chain) of D. Chains and directed sets provide us with an abstract notion of "approximating sequence", where "approximation" is in the sense of learning more and more specific information. We require approximating sequences to "converge" to a "limit", which is the least upper bound of the directed set or chain. This is captured by

DEFINITION 2.3. *A* directed complete partial order *(dcpo) is a poset* $(D, \sqsubseteq)$ *with a bottom element* $\bot$ *and such that for any directed* $X \subseteq D$ *there is a least upper bound* $\bigsqcup X \in D$*.*

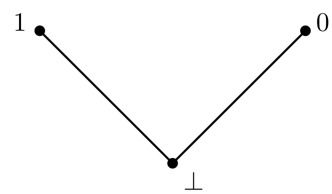
EXAMPLE 2.1. We begin our running example by introducing the syntax for Kleene's three-valued logic. This is given by the following recursive specification (where Var is a given countable set of propositional variables)::

$$\phi ::= x(\in Var) \mid \mathbf{true} \mid \mathbf{false} \mid (\phi \wedge \psi) \mid (\phi \vee \psi) \mid \neg\phi.$$

Strings in this language will be called propositional formulas or just formulas. This syntax is the same as that of standard propositional logic; the difference lies in the semantics. The meaning of a propositional formula is specified by its behavior on *partial truth assignments* (ptas). A pta is a function $e : Var \to \{0, 1, \bot\}$, with 1 understood as "true", 0 as "false", and $\bot$ as "undefined". Note that we use truth values 0 and 1 in truth assignments, instead of **false** and **true**. This is to keep clear the distinction, at least temporarily, between syntax and semantics.

It is customary to describe a pta by a string of *literals*: variables or their complements. For example, $ab\overline{cd}$ represents the pta which maps a and b to 1, and c and d to 0 (with all other variables mapping to $\bot$).

The set of all ptas forms a dcpo $[Var \to \{0, 1, \bot\}]$ by defining the order pointwise: $e \sqsubseteq e'$ if $e(x) \sqsubseteq e'(x)$ for every $x \in Var$, where the truth values are ordered by letting $\bot \sqsubseteq 0$ and $\bot \sqsubseteq 1$ (0 and 1 will thus be incomparable). We write T for this cpo, pictured below.



Definition 2.4. *Let $(D, \sqsubseteq)$ be a poset, and $A \subseteq D$. The* upper closure *of A, written $\uparrow A$, is the set of all $d \in D$ such that for some $a \in A$, $a \sqsubseteq d$. A is* upward closed *if $A = \uparrow A$. Similarly for lower closure (use $\downarrow$). The set of minimal elements of a subset X of D is written μX.*

Example 2.2. A typical example of an upper-closed set is the set of satisfiers (models) of a 3-valued propositional formula. To fix terminology, we introduce the truth tables for the connectives $\vee, \wedge$ and $\neg$, usually attributed to Kleene.

$\wedge$	**1**	**0**	$\bot$
1	**1**	**0**	$\bot$
0	**0**	**0**	**0**
$\bot$	$\bot$	**0**	$\bot$

$\vee$	**1**	**0**	$\perp$
1	**1**	**1**	**1**
0	**1**	**0**	$\perp$
$\perp$	**1**	$\perp$	$\perp$

	$\neg$
1	**0**
0	**1**
$\perp$	$\perp$

These truth tables provide a method to evaluate any formula under a partial truth assignment. Given any pta $e : Var \to \mathsf{T}$, write $[\![\phi]\!](e)$ for the evaluation of the formula ϕ under the pta e. We have, inductively,

1. $[\![\mathbf{true}]\!](e) = 1$; $[\![\mathbf{false}]\!](e) = 0$;
2. $[\![v]\!](e) = e(v)$ for $v \in Var$;
3. $[\![\phi \wedge \psi]\!](e) = \wedge([\![\phi]\!](e), [\![\psi]\!](e))$, where $\wedge$ is obtained from the first truth table above.
4. Similarly for the connectives $\neg$ and $\vee$.

Given a pta e and a formula ϕ, we say that e is a (positive) satisfier or model of ϕ if $[\![\phi]\!](e) = 1$. In this case we also write $e \models \phi$. (Similarly, e is a negative satisfier of ϕ if $[\![\phi]\!](e) = 0$.) It is easy to show that for any formula ϕ, the set

$$[\![\phi]\!] = \{e \mid e \models \phi\}$$

is upward-closed. This property is also sometimes called the *persistence* property.

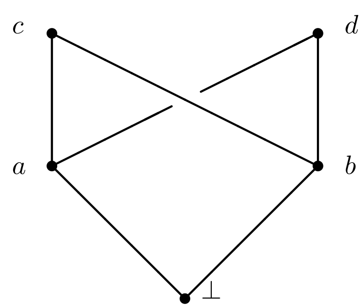
DEFINITION 2.5. *Let $(D, \sqsubseteq)$ be a dcpo. A* compact (finite) element *e of D is one such that whenever $e \sqsubseteq \bigsqcup L$ with L directed, we also have $e \sqsubseteq y$ for some $y \in L$. The set of compact elements of a dcpo D is written as $\mathsf{K}(D)$.*

EXAMPLE 2.3. In the dcpo $D = [Var \to \mathsf{T}]$ of ptas, an element e is compact if and only if it maps all but a finite number of variables to $\perp$.

DEFINITION 2.6. *An* algebraic dcpo *is a directed complete partial order such that every element is the directed join of the finite elements that subsume it. A* Scott domain *is an algebraic dcpo in which every consistent (bounded from above) subset has a least upper bound.*

The idea of an algebraic domain is that roughly, one can obtain information about any element by taking a suitable approximating sequence of prespecified finite elements. A Scott domain has this property, and in a sense is like a lattice as well.

EXAMPLE 2.4. The domain $D = [Var \to \mathsf{T}]$ is in fact a Scott domain. Here is a picture of a non-Scott domain:



We turn to the important notion of the *Scott topology* on a dcpo.

DEFINITION 2.7. *Let* $(D, \sqsubseteq)$ *be a dcpo. A set* $U \subseteq D$ *is said to be* Scott open *if (i) it is upward closed; and (ii) for any directed* $L \subseteq D$*, we have* $\bigsqcup L \in U$ *iff* $U \cap L \neq \emptyset$*. The set of Scott-opens of* D *is denoted* ΩD*. An open set is* compact open *if it is a compact element of the lattice* $(\Omega D, \subseteq)$.

A Scott-open set captures, first of all, the notion of an *affirmative predicate* on D. A predicate (set) U is said to be affirmative if, whenever an element $x \in U$, it is somehow possible to test that $x \in U$ with a finite number of tests. That is, if $x = \bigsqcup\{d \in \mathsf{K}(D) \mid d \sqsubseteq x\}$ and $x \in U$ then for some $d \sqsubseteq x$, d is compact and $d \in U$.

The collection of affirmative predicates on a set should therefore be closed under finite intersections and arbitrary unions, and contain the empty set and the whole space, thus forming a *topological space*. But this is not all that is captured by the Scott topology. One can reclaim the order $\sqsubseteq$ on D because $x \sqsubseteq y$ iff for all

Scott-open U, if $x \in U$ then $y \in U$. Furthermore, an element $e \in D$ is compact if and only if $\uparrow e$ is compact open.

EXAMPLE 2.5. In the domain $D = [Var \to \mathsf{T}]$, a set M is compact open iff it is the set of satisfiers of some propositional formula. The easiest way to see this is to notice that any formula is equivalent (in the sense of having the same positive and negative satisfiers) to a formula in disjunctive normal form (DNF). One can get a unique DNF by discarding disjuncts which are non-minimal; for example in $a \vee (a \wedge b)$ one can discard the disjunct $(a \wedge b)$, because $a \sqsubseteq ab$. Each remaining disjunct then corresponds to a minimal pta in the set of satisfiers of the formula; the upward closure of the set of minimal satisfying pta's is a compact open set, because it is a finite union of the compact opens determined by taking the upward closure of each minimal pta. Conversely, in any algebraic domain, one can prove that a set M is compact open iff it is the upward closure of a finite number of compact elements. So if a set of elements of $[Var \to \mathsf{T}]$ is compact open, one can read off a DNF formula by obtaining the minimal elements of M.

In general, an open set in this domain can be thought of as the set of satisfiers of an infinite disjunction of finite conjunctions of literals (variables or their complements.)

DEFINITION 2.8. *An algebraic dcpo is said to be* coherent *if the intersection of any two compact open sets is compact open.*

EXAMPLE 2.6. The non-Scott domain in Example 2.4 is coherent. In fact, every finite poset with $\perp$ will be coherent algebraic. For an infinite counterexample, let N be the domain

$$\{a_0 \sqsupseteq a_1 \sqsupseteq \ldots\} \cup \{c, d, \perp\},$$

where c and d are incomparable elements below all the a_is, and $\perp$ is the bottom element. The intersection U of $\uparrow c$ with $\uparrow d$ is open, because it is the infinite union of the compact open sets $\uparrow a_i$. But this union is not compact open, for the collection $\{\uparrow a_i \mid i \in \omega\}$ is directed, and its union is U; but U is not a subset of any $\uparrow a_i$.

In this paper all domains will be coherent algebraic.

DEFINITION 2.9. *A* Scott-open filter *is a collection J of Scott open subsets of D which is closed under intersection and superset; and such that if a directed union of opens is in J, then some open in the directed collection is also in J. A filter is proper if it does not contain the empty set (iff it is not the lattice of all Scott opens).*

EXAMPLE 2.7. In three-valued logic, a filter corresponds to a collection of logical equivalence classes of finite formulas in some set T of formulas. The formulas are represented first as DNF clauses and then as compact open sets, together with "infinite DNF formulas" – arbitrary disjunctions of finite conjunctions, represented by general Scott open sets. Scott-open filters are those for which arbitrary Scott open sets always correspond to infinite disjunctive formulas which are weakenings of finite formulas in the theory T.

In somewhat more detail, we can think of Scott-open filters as being determined by *logically closed theories*. Recall that a theory T entails a formula ϕ means that every pta e satisfying (positively) every formula in T must also satisfy ϕ. A theory T is logically closed if every formula entailed by T is in T. For a logically closed theory T let

$$J(T) = \{U \in \Omega D \mid (\exists \phi \in T)([\![\phi]\!] \subseteq U)\}.$$

We claim that $J(T)$ is a Scott-open filter. Clearly it is closed under superset. Let $U \supseteq [\![\phi]\!]$ and $V \supseteq [\![\psi]\!]$ be members of $J(T)$. Then $U \cap V \supseteq Z = [\![\phi \wedge \psi]\!]$. Clearly $\phi \wedge \psi$ is entailed by T and so this formula is in T. This shows $J(T)$ is closed under intersection. Lastly, we check the Scott-open property. Suppose the directed union $\bigcup_i U_i$ is a member of $J(T)$. Then for some $\phi \in T$, $[\![\phi]\!] \subseteq \bigcup_i U_i$. Since $[\![\phi]\!]$ is compact open, $[\![\phi]\!] \subseteq U_i$ for some i. This proves $U_i \in J(T)$ for some i.

We use the notion of Scott-open filter to help relate the logical notion of compactness to the topological notion. The logical notion is often taken to be that if a formula is a consequence of a collection T of formulas, then it is a consequence of a finite subcollection F of T. The following gives the topological notion:

DEFINITION 2.10. *A subset C of a dcpo D is said to be* compact *if the collection $\{U \in \Omega D \mid C \subseteq U\}$ is a Scott-open filter.*

It is an easy exercise to show that (working in the Scott topology) this definition is equivalent to the standard topological definition: every open covering of C has a finite subcovering. We will show in the next section how to connect this topological definition to the logical one.

We need one more technical definition, which we use below to characterize the possible sets which can be the set of models of a theory:

DEFINITION 2.11. *A subset S of a dcpo is said to be* saturated *iff it is the intersection of all Scott-open sets which contain it. (This is equivalent to its being upward-closed.)*

Finally, we recall some definitions relevant to the Smyth powerdomain of an algebraic dcpo. The original idea of powerdomains was to lift the underlying order in a domain to the powerset level, thus providing a domain-theoretic semantics for non-determinism. There are several natural ways to do this, depending on how we relate order in sets of domain elements to the underlying order of the domain. For us, the appropriate tool is the Smyth preorder, defined below. This preorder is defined on finite sets of compact elements of the underlying domain, and then extended to a partial order using the technique of *ideal completion*. The relevant definitions are as follows.

DEFINITION 2.12. *Let $(P, \preceq)$ be a preordered set (i.e., $\preceq$ is reflexive and transitive). An* ideal *of P is a subset I of P which is downward closed and directed (these notions make sense for the preorder $\preceq$). An ideal is* proper *iff it is not equal to D. The (proper)* ideal completion *of a preordered set $(P, \preceq)$ is the set of all (proper) ideals of P, ordered by inclusion.*

DEFINITION 2.13. *The* Smyth preorder *on a collection of subsets of a dcpo $(D, \sqsubseteq)$ is defined by setting $X \sqsubseteq^\sharp Y$ iff for every element $y \in Y$ there is an element $x \in X$ with $x \sqsubseteq y$.*

DEFINITION 2.14. *Let $(D, \sqsubseteq)$ be an algebraic dcpo. The* Smyth powerdomain *$P^\sharp(D)$ is defined to be the proper ideal completion of the collection $P_f(\mathsf{K}(D))$ of finite sets of compact elements of D, under the Smyth preorder.*

This definition of the Smyth powerdomain usually proves to be cumbersome, which is why we do not provide examples. One way to simplify matters is given by the following result. For a proof see [AJ94].

THEOREM 2.15. *The Smyth powerdomain $P^\sharp(D)$ of an algebraic dcpo $(D, \sqsubseteq)$ is isomorphic to the domain consisting of all nonempty Scott-compact saturated subsets of D, ordered by reverse set inclusion. The compact elements of the powerdomain are the nonempty compact open subsets of D.*

Our representation result, which works in any coherent algebraic dcpo D, is that the Smyth powerdomain of D can be viewed as the set of all logically closed consistent theories over D, where a "formula" in D is just a set of compact elements, and a theory is a set of formulas. The relevant ordering on logically closed theories is just set inclusion. In fact, logically closed consistent theories are nothing more or less than proper ideals using the Smyth preorder (cf. Proposition 3.5.)

3. THEORIES

The following definition contains ideas essential for the rest of the paper.

DEFINITION 3.1. *Let D be a coherent algebraic dcpo (with $\bot$) and let $\mathsf{K}(D)$ be the set of compact elements of D. We define a* clause *to be a finite subset of $\mathsf{K}(D)$. A theory is a set of clauses. For $w \in D$, and a clause X, we say that $w \models X$ if there is a $c \in X$ with $c \sqsubseteq w$. For a theory T we say that w is a model of T, and write $w \models T$, if $w \models X$ for all $X \in T$. We say that $T \models X$ if for all w, if $w \models T$ then $w \models X$. (If T is a singleton set, say $\{Z\}$, then we just write $Z \models X$.) A theory T is* closed *if for all X, $T \models X \Rightarrow X \in T$. A theory is* consistent *if it does not entail ($\models$) the empty clause. Equivalently, there is a w such that $w \models T$.*

We allow a theory to contain no clauses. Every w will satisfy this theory; it is in fact logically equivalent to the theory consisting of one clause $\{\bot\}$.

EXAMPLE 3.1. For three-valued logic, we adopt an abstract syntax where a disjunction of a conjunction of literals is simply represented as an abstract clause obtained by taking a conjunction of literals to be the corresponding pta and the disjunction of these conjuncts to be a set of these ptas. For example, $(a \vee (a \wedge \neg b) \vee (b \wedge \neg c))$ translates domain-theoretically to $\{a, a\bar{b}, b\bar{c}\}$. With this translation, the other parts of the above definition should be very familiar.

Theories over a domain D are another way of talking about the ideal completion of D under the Smyth preorder. The essential connection is that a clause X entails a clause Y if and only if $Y \sqsubseteq^\sharp X$ in the Smyth preorder. With all of this in mind, we can state our main representation theorem.

THEOREM 3.2. *Let D be a coherent algebraic dcpo. The collection of all nonempty Scott-compact saturated subsets of D, ordered by reverse inclusion, is isomorphic to the set of all consistent closed theories over D, ordered by inclusion.*

To prove this theorem, we provide an order-isomorphism between the collection of theories and the collection of compact saturated sets. We can then use Theorem 2.15 to connect compact saturated sets to the original definition of the Smyth powerdomain as an ideal completion, or we can use Proposition 3.5 to see this directly. For the proof we need the Hofmann-Mislove theorem [HM81]. Refer to the definitions of Scott-open filters (Definition 2.9) and compact saturated sets (Definitions 2.10, 2.11.)

LEMMA 3.1 (Hofmann-Mislove). *Let D be an algebraic dcpo. There is a 1-1 order-preserving correspondence between the collection of (proper) Scott-open filters over D, ordered by inclusion, and the (nonempty) compact saturated subsets of D, ordered by reverse inclusion. This is given by the map sending* *a filter F to $\bigcap_{U \in F} U$; the inverse map sends the compact saturated set K to $\{U \in \Omega D \mid K \subseteq U\}$.*

Proof (Proof of Theorem 3.2.). We show that there is an order isomorphism between the closed theories and the compact saturated sets over D. We start with traditional maps in logic.

DEFINITION 3.3. *Let K be compact saturated. Define $Th(K)$ to be the set $\{X \mid (\forall m \in K)(m \models X)\}$. For any theory T let $[\![T]\!]$ denote the set of models of T.*

These maps are inverses of each other, provided their domains are appropriate:

LEMMA 3.2. *If T is logically closed, then $Th([\![T]\!]) = T$.*

Proof. Clearly $T \subseteq Th([\![T]\!])$. For the converse, we need to show that $T \models X$ whenever $X \in Th([\![T]\!])$. But if $m \models T$ then $m \in [\![T]\!]$. Thus $m \models X$ by definition of Th. ▮

LEMMA 3.3. *If K is compact saturated, then $[\![Th(K)]\!] = K$.*

Proof. Clearly $K \subseteq [\![Th(K)]\!]$. For the converse, let $m \in [\![Th(K)]\!]$. Then $m \in \uparrow X$ for all $\uparrow X \supseteq K$. We claim that $w \in U$ for all Scott-open $U \supseteq K$. The reason is that U is the union of all $\uparrow d$ for d a minimal element of U, so this collection forms an open cover of K. By compactness there is a finite subcover; let $\uparrow X$ be the union of this finite subcover. Then $m \in \uparrow X$ and so $m \in U$. The conclusion follows since K is saturated. ▮

It is easy to check that $Th(K)$ is logically closed for any K. Also, $[\![\cdot]\!]$ is clearly order-reversing. Therefore, to prove Theorem 3.2, it suffices to prove that for any (consistent) theory T, the set $[\![T]\!]$ is (nonempty) compact saturated. We do this by factoring the map $[\![\cdot]\!]$ through the collection of (proper) Scott-open filters, and then applying the Hofmann-Mislove theorem. For a logically closed theory T let

$$J(T) = \{U \mid (\exists X \in T)(\uparrow X \subseteq U)\}.$$

We claim that in a coherent algebraic domain, this collection is a Scott-open filter. The proof mirrors the reasoning in Example 2.7. We need to check closure under superset and intersection, together with the Scott-open property. The first and third of these checks follow exactly as in Example 2.7. For closure under intersection, let $U \supseteq \uparrow X$ and $V \supseteq \uparrow Y$ be members of $J(T)$, with $X, Y \in T$. Then $U \cap V \supseteq Z = \uparrow X \cap \uparrow Y$. Since D is coherent, Z is compact open and the set

μZ of minimal elements of Z is a clause entailed by T (a satisfier of X and Y must be in Z, so above some minimal element of Z). Therefore $\mu Z \in T$ and so $U \cap V \in J(T)$. This shows $J(T)$ is closed under intersection, and establishes our claim.

But we have that $\bigcap J(T) = [\![T]\!]$. To see this, let $w \in [\![T]\!]$, and U be such that $\uparrow X \subseteq U$ for some $X \in T$. Then $w \in \uparrow X \subseteq U$. Conversely, let $w \in \bigcap F$. Take an arbitrary $X \in T$. Then $\uparrow X$ is a set $U \supseteq \uparrow X$, so $w \in \uparrow X$. Thus $w \models X$ for all $X \in T$ and so $w \in [\![T]\!]$. We conclude that $[\![T]\!]$ is compact saturated, by the Hofmann-Mislove theorem. This completes the proof of our representation result. ▮

Remark 3. 1. The assumption of coherence is necessary in this theorem. To see this, consider the domain N constructed in Example 2.6:

$$\{a_0 \sqsupseteq a_1 \sqsupseteq \ldots\} \cup \{c, d, \bot\}.$$

The compact saturated sets in this domain are exactly the upper closures of all these finite elements, together with the upward closure of $\{c, d\}$. However, the *conjunction* of the two clauses $\{c\}$ and $\{d\}$ is not represented as a compact saturated set, though it makes perfect sense as a theory. It is easy to draw the Hasse diagrams of both the set of compact saturated sets and the set of theories, to see that these two domains are not isomorphic.

COROLLARY 3.4 (Compactness). *A clause is a logical consequence of a theory T iff it is a logical consequence of a finite subset of T.*

Proof. We claim that the compact elements of the space of theories are exactly the logical closures of finite theories. To see this, we recall that the compact elements of the domain of compact saturated sets are indeed the compact open subsets in the Scott topology of that domain (Theorem 2.15). By the proof of Theorem 3.2 we have that

$$[\![T]\!] = \bigcap\{U \mid (\exists X \in T)(\uparrow X \subseteq U)\}.$$

If T is finite then $[\![T]\!] = [\![\overline{T}]\!]$ is compact open, by the above equation. (Here $\overline{T}$ is the closure of T under logical consequence.) Conversely, if K is compact open then K is $[\![T]\!] = [\![\overline{T}]\!]$ for some finite T, namely $\{\mu K\}$, where μK is the set of minimal elements of K. Thus the order-isomorphism $[\![\cdot]\!]$ restricts to the closures of finite theories and the compact opens. This establishes our claim.

Now suppose that the theory $T \models X$. The collection of closures of finite subcollections of T is a directed set of compact elements of the space of theories. and its least upper bound is the closure of T. Since the closure of X

is contained in the closure of T, and is itself a compact element, this means that X is in the closure of one of the finite subcollections of T, as desired. ▮

Remark 3. 2. One can avoid the use of Theorem 2.15 in the foregoing proof by making the space of Scott-open filters over D into a dcpo $SOF(D)$ under the inclusion ordering. Then it is easy to check that the finite elements of $SOF(D)$ are the principal filters generated by compact open subsets of D, and by showing that the map $\lambda T.J(T)$ is an isomorphism sending the logical closure of a finite theory to the principal filter generated by the intersection of the compact opens corresponding to each clause in the finite theory.

We use Corollary 3.4 to characterize theories as ideals in the $\sqsubseteq^\sharp$ preorder.

PROPOSITION 3.5. *A theory over D is logically closed iff it is an ideal.*

Proof. Suppose T is a logically closed theory. We show that T is an ideal. The downwards closedness of T follows directly from logical closure. We claim that T is directed. For a finite subcollection F of T, put

$$Z := \bigcap \{\uparrow X \mid X \in F\}.$$

By coherence, μZ, the set of minimal elements of Z, is a clause; moreover, $T \models \mu Z$. By logical closure μZ is a member of T, and it is clearly an upper bound for F.

Conversely, let I be an ideal. If $I \models X$, then by Corollary 3.4, there is a finite subcollection $F = \{X_1, \ldots, X_n\}$ of I such that $F \models X$. By directedness of I, there is a $Y \in I$ with $X_1, \ldots, X_n \sqsubseteq^\sharp Y$. It follows that $Y \models X$, or $X \sqsubseteq^\sharp Y$. But $Y \in I$ and so $X \in I$ by downward closure of I. ▮

EXAMPLE 3.2. We close this section by applying Corollary 3.4 to give a proof of the compactness theorem in classical propositional logic. Clearly it gives a compactness result for Kleene's 3-valued propositional logic already. For the classical result, just notice that a theory T entails a clause X classically if and only if T together with $\{(v \vee \neg v) \mid v \in Var\}$ entails X in Kleene's logic.

4. A RESOLUTION RULE

We now investigate a rule of inference for logical entailment in the space of theories over D. Some notation will be useful.

DEFINITION 4.1. *Let $\{a_1, \ldots, a_n\}$ be a finite subset of $\mathsf{K}D$. By $\mathsf{mub}\{a_1, \ldots, a_n\}$ we mean the set of minimal elements of $\uparrow\{a_1\} \cap \ldots \cap \uparrow\{a_n\}$.*

The set $\mathsf{mub}\{a_1, \ldots, a_n\}$ is in fact the set of minimal upper bounds of all of the a_is. It is a finite set of compact elements, by the definition of algebraic coherence.

EXAMPLE 4.1. Refer to Example 2.4. We have $\mathsf{mub}\{a, b\} = \{c, d\}$.

DEFINITION 4.2. *Let $\{X_1, \ldots, X_n\}$ be a clause set. The* clausal hyper-resolution rule *(HR) is the following, where Y is a clause:*

$$\frac{X_1 \;\; X_2 \;\; \cdots X_n; \quad a_i \in X_i \textit{ for } 1 \leq i \leq n; \quad \mathsf{mub}\{a_1, \ldots, a_n\} \models Y}{Y \cup \bigcup_{1 \leq i \leq n} (X_i \setminus \{a_i\})}$$

This rule applies to a clausal theory whenever $X_1, \ldots, X_n$ are clauses of the theory, and we can find $a_i \in X_i$ satisfying the condition $\mathsf{mub}\{a_1, \ldots, a_n\} \models Y$. We allow the set $\{a_1, \ldots, a_n\}$ to be inconsistent. In this case, $\mathsf{mub}\{a_1, \ldots, a_n\} = \emptyset \models \emptyset$, so that we get the usual notion of resolution (generalized to partial logic).

We need to adjoin two special rules to (HR) in order to get a complete proof system. First, we notice that when one of the clauses X_i is empty, then the rule (HR) does not apply. However, classical logic allows the inference of any clause from the empty clause. So we need to adjoin to (HR) a special rule which allows the inference of any clause from a finite set of clauses, one of which is empty. Second, we need a "getting-started" rule allowing the inference of the clause $\{\bot\}$ from no assumptions. We summarize the two special rules:

$$\text{(EC)} \; \frac{X_1 \;\; X_2 \;\; \cdots \emptyset \cdots X_n}{Y} \qquad \text{and} \qquad \text{(GS)} \; \frac{}{\{\bot\}}.$$

The rule (HR) is sound. By this, we mean that if $x \in D$ satisfies each of $X_1, \ldots, X_n$, then for any choice of $\{a_1, \ldots, a_n\}$ and Y fulfilling the antecedent, x will satisfy the consequent. This is clear considering the two cases for x: either $a_i \sqsubseteq x$ for all i, in which case x is above some minimal upper bound of the a_i, and therefore satisfies Y, or some a_i does not subsume x, in which case $x \models X_i \setminus \{a_i\}$.

Write $X_1, \ldots, X_n \vdash^* X$ if X can be derived from $X_1, \ldots, X_n$ using the rule (HR) or the special inference rules (EC) and (GS). More generally, for a theory T write $T \vdash^* X$ if $X_1, \ldots, X_n \vdash^* X$ for some $X_1, \ldots, X_n \in T$. Our first objective is to show that if $\{X_1, \ldots, X_n\} \models X$, then $X_1, \ldots, X_n \vdash^* X$. We use this to prove the Completeness Theorem: if $T \models X$ then $T \vdash^* X$.

LEMMA 4.1 (Weakening). *If E is a nonempty clause and $E \models X$ then $E \vdash^* X$.*

Proof. First note that if $E \models X$ and $f \in E \setminus X$ then there is $m \in X$ with $f \sqsubseteq m$, so $E \vdash (E - f) \cup \{m\}$. Iterating this process removes all elements of $E \setminus X$ so that $E \vdash^* D$ for some $D \subseteq X$. Without loss of generality $D \neq \emptyset$, since otherwise $D \vdash X$ by the special inference rule. If now there is a $c \in X \setminus D$ then choose a $d \in D$. Then $D \vdash (D - d) \cup \{d, c\}$. ▮

LEMMA 4.2. *Let P and Q be nonempty clauses, and let $P \bowtie Q$ be the clause consisting of the union of all the sets* $\mathsf{mub}\{p, q\}$*, where $p \in P$ and $q \in Q$. Then $P, Q \vdash^* P \bowtie Q$.*

Proof. Let $P = \{p_1, \ldots, p_m\}$ and $Q = \{q_1, \ldots, q_n\}$. We seek a way to prove the clause $P \bowtie Q$. We do this using a combinatorial argument. For $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$ let $X(i, j)$ be the clause

$$\{p_1, \ldots, p_i, q_1, \ldots, q_j\} \cup \bigcup \{\mathsf{mub}\{p_k, q_l\} \mid m \geq k > i, n \geq l > j\}$$

and in addition let $X(m, 0) = P$ and $X(0, n) = Q$. Let S be the set of pairs (i, j) such that $X(i, j)$ is in the resolution closure of $\{P, Q\}$. Then it is easy to check that

- $(m-1, n-1) \in S$;
- $(i+1, j)$ and $(i, j+1) \in S \Rightarrow (i, j) \in S$;
- $(m-1, j+1) \in S \Rightarrow (m-1, j) \in S$;
- $(i+1, n-1) \in S \Rightarrow (i, n-1) \in S$.

It follows by downward induction on $i + j$ that $(0, 0) \in S$, or that $P \bowtie Q$ is in the resolution closure of $\{P, Q\}$, as we wanted. ▮

Remark 4. 1. Notice that $P \bowtie Q$ is a clause which is logically equivalent to the theory $\{P, Q\}$ – they have the same set of models. Up to logical equivalence, therefore, $\bowtie$ is associative. In view of the next corollary, this says that binary resolution — resolving on at most two clauses in the antecedent — is sufficient for deriving logical consequences.

COROLLARY 4.3 (Completeness Theorem). *If $T \models X$ then $T \vdash^* X$.*

Proof. Suppose that $T \models X$. By the compactness theorem (Corollary 3.4), there are finitely many clauses $X_1, \ldots, X_n$ such that $\{X_1, \ldots, X_n\} \models X$. First assume these clauses are consistent. By Lemma 4.2 and induction,

$$X_1, \ldots, X_n \vdash^* E = X_1 \bowtie \ldots \bowtie X_n.$$

Notice that E is logically equivalent to the conjunction of $X_1, \ldots, X_n$. Thus X must be a logical consequence of E, so that by Lemma 4.1, $E \vdash^* X$.

If the collection $\{X_1, \ldots, X_n\}$ is inconsistent, then choose a minimal subcollection of nonempty clauses which is inconsistent. Then by induction the empty clause is derivable from this subcollection, and then by the rule (EC), any clause will then be derivable. ▮

5. CLAUSAL LOGIC PROGRAMMING IN COHERENT DOMAINS

As proved in the previous section, the hyper-resolution rule is complete for deriving the logical consequences of a consistent theory over the Smyth powerdomain. Our first goal is to state a similar rule for deriving logical consequences of a disjunctive logic program over a coherent algebraic domain D. We begin with this latter definition.

DEFINITION 5.1. *A* disjunctive logic program *over D is a set P of rules of the form $\theta \leftarrow \tau$, where θ, τ are clauses over D.*

The idea of this definition is that a rule $\theta \leftarrow \tau$ adds a clause θ to a theory whenever τ is entailed by the theory.

EXAMPLE 5.1. A standard form of a disjunctive propositional logic program is a collection of rules of the form

$$b_1, \ldots b_m \leftarrow a_1, \ldots, a_n$$

where the a's and b's are propositional literals. The set of a's is taken conjunctively and the set of b's is taken disjunctively. A particular example might be

$$p, \neg q, r \leftarrow s, \neg p.$$

In our formulation this corresponds to the rule

$$\{p, \overline{q}, r\} \leftarrow \{s\overline{p}\}.$$

Here we are referring to our running example of three-valued logic. The left side of such standard rules will always translate to a domain-theoretic rule with a singleton clause in its "body"; more generally, we allow disjunctive clauses as both body and head.

DEFINITION 5.2. *Let P be a disjunctive logic program over D. An element $e \in D$ is said to be a model of P if for every rule $\theta \leftarrow \tau$ in P, if $e \models \tau$, then $e \models \theta$. A clause Y is a logical consequence of P if every model of P satisfies Y. We write $\overline{P}$ for the set of all clauses which are logical consequences of P.*

EXAMPLE 5.2. In the previous example, $s\overline{p}r$ is a model of the program P consisting of just the one given rule. However, the only logical consequences of P will be clauses containing the null pta $\bot$. This is because $\bot$ itself is a model of P – it does not satisfy $\{s\overline{p}\}$. This brings out an important difference between 3-valued and 2-valued (classical) logic programs. Let R be the program $q \leftarrow p$. In classical terms, this is equivalent to $q \vee \neg p$, and so has this formula as a consequence. But as in the case of the program P, the program R has only trivial logical consequences. To obtain the classical case, adjoin to R all rules of the form

$$\{v, \overline{v}\} \leftarrow \{\bot\}$$

for v a propositional variable. This forces any model of the augmented program R' to be a total truth assignment. So $\{q, \overline{p}\}$ will now in fact be a logical consequence of R'.

We will introduce an operator T_P on the space of logically closed theories over D which will provide the fixpoint semantics of the program P, and then show that the least fixpoint of T_P gives us the set of logical consequences of P. In order to define T_P we formulate the hyper-resolution rule HR(P) determined by P:

$$\frac{X_1 \quad X_2 \quad \cdots X_n; \quad a_i \in X_i \ \text{ for } 1 \leq i \leq n; \quad \theta \leftarrow \tau \in P; \quad \mathsf{mub}\{a_i \mid 1 \leq i \leq n\} \models \tau}{\theta \cup \bigcup_{1 \leq i \leq n} (X_i \setminus \{a_i\})}.$$

We say that $\theta \cup \bigcup_{1 \leq i \leq n} (X_i \setminus \{a_i\})$ is an HR(P)-consequence of $\{X_1, \ldots, X_n\}$; further, we say that Y is a HR(P)-consequence of a theory T if it is an HR(P)-consequence of some $\{X_1, \ldots, X_n\} \subseteq T$.

In the next definition, we write $Cn(T)$ for the closure of the theory T under the clausal inference rule. By the completeness theorem, this coincides with the logical closure of T.

DEFINITION 5.3. *Let T be a logically closed theory over D, and let P be a program. We define*

$$T_P(T) = Cn\{Y \mid \ Y \textit{ is an HR(P)-consequence of } T\}.$$

The intent of using the operator T_P is as follows. We are interested in those clauses derivable from the clause $\{\bot\}$ using a finite sequence of applications of the rule HR(P), interleaved with applications of the clausal inference rule. This set of clauses will be what we get by iterating the T_P operator some finite number of times. Technically, $T_P(T)$ is the entire set of logical consequences (Cn) of one application of the HR(P). This makes the computation of clauses generated by iterating the T_P operator technically problematic. It is easy to see, though, that any clause derivable by iterating the operator will in fact be derivable (because of compactness) in the interleaved fashion we have suggested.

The operator T_P maps logically closed theories to logically closed theories; it may be that $T_P(T)$ is inconsistent, and so not an official element of the Smyth powerdomain of D. We do have, however, the following result:

PROPOSITION 5.4. *T_P is a continuous function on the space of logically closed theories.*

Proof. Let Λ be a directed set of theories. We have to show that

$$T_P(\bigcup \Lambda) = \bigcup_{T \in \Lambda} T_P(T).$$

Since T_P is clearly monotone, the inclusion of right in left is obvious. To go from left to right, suppose that Z is a logical consequence of the set of HR-consequences of $\bigcup \Lambda$. Then by compactness, Z is a logical consequence of a finite number of these HR-consequences. This means that in fact Z is a logical consequence of the set of HR-consequences of some finite number $X_1, \ldots, X_m$ of clauses of $\bigcup \Lambda$. and therefore of some particular $T \in \Lambda$, since Λ is directed. This shows that Z is a member of the right hand side as well. ▮

By the Scott-Knaster-Kleene-Tarski theorem, T_P has a least fixpoint fixT_P. We are going to show that $\overline{P} = \text{fix}T_P$. Several lemmas will prepare the way. The first two lemmas are from [RZ97].

LEMMA 5.1. *In a coherent algebraic domain D, every compact saturated set is the upward closure of the set of its minimal elements.*

Proof. This is a generalization of the proof in [RZ97] from Scott domains to coherent algebraic domains.

Let N be compact saturated. Since N is saturated, N is the intersection of all Scott opens O such that $O \supseteq N$. However, each Scott open set can be expressed as an (infinite) union of compact open sets. Since N is compact, we know that N is in fact the intersection of all compact Scott opens O such that $O \supseteq N$.

For any element $x \in N$, we want to show that there exists a minimal element $y \in N$ such that $x \sqsupseteq y$. Fix an arbitrary element $x \in N$, and consider a chain C in $\downarrow x \cap N$. We claim that the set of all compact lower bounds of C is nonempty and directed. Indeed, let m and n be compact lower bounds for C. There are a finite number of minimal upper bounds for m and n. Let b_1 be such a minimal upper bound. Consider the set $C_1$of elements of C above b_1. If this is not all of C, then there are elements of C below everything in C_1, and not above b_1, but some of these must be above some other minimal upper bound b_2. Fix such a b_2 and let the set of elements of C above b_2 but not b_1 be C_2. If $C_1 \cup C_2$ is not all of C then we repeat this argument. The argument terminates because there are only finitely many minimal upper bounds of m and n, so that all of C must be above some minimal upper bound b of m and n, whence b is the required element needed for directedness.

>From the claim it follows that $m_C = \bigsqcup\{m \mid m \text{ is a compact lower bound of } C\}$ exists in D. Clearly m_C is a lower bound of C, for each c in C is the least upper bound of all compact elements subsuming it. We show that m_C is a member of $\downarrow x \cap N$, as well. Let O be any compact open set containing N. Clearly, $C \subseteq O$. Since O is compact, μO is a finite set, and $O = \uparrow \mu O$. Using the argument of the previous paragraph, we have that there exists some $a \in \mu O$ such that $C \subseteq \uparrow a$, and hence $m_C \in \uparrow a$ (because a is a compact lower bound of C), which implies $m_C \in O$. Therefore, m_C is a member of every compact open set containing N, and so is in N by saturation.

To summarize, we have shown that for any element $x \in N$, the nonempty set $\downarrow x \cap N$ has the property that every chain in it has a lower bound. By Zorn's lemma, $\downarrow x \cap N$ has a minimal element, say, y. Such a y is a minimal element of N subsuming x. ▮

LEMMA 5.2 (Interpolation Theorem). *Let E be a compact saturated set over a coherent algebraic dcpo, g be a minimal element of E, and d be a compact element subsuming g. For any compact open set K such that $K \supseteq E$, there exists a compact open set L, such that*

1. $K \supseteq L \supseteq E$,
2. the minimal element l of L subsuming g is unique, and
3. $d \sqsubseteq l$.

Proof. Once again the proof of this result in [RZ97] is for Scott domains, The following argument generalizes our previous proof and corrects a minor error in it.

Choose a minimal element f of K subsuming g. Let l be a minimal upper bound of d and f, such that $l \sqsubseteq g$. Let $R = \{x \in K \cap \mathsf{K}(D) \mid x \not\sqsubseteq g\}$. Consider the collection C of all $\uparrow x$ for $x \in R$ together with $\uparrow l$. We claim that this is an open cover of E. In proof, let $e \in E$. Without loss of generality, g does not subsume e. Hence by algebraicity there is a compact element $n \in D$ such that $n \sqsubseteq e$ but n does not subsume g. Since $e \in E$, then there is a $b \in \mu K$ with $b \sqsubseteq e$. By coherence, there must be a minimal upper bound m of b and n in $K \cap \mathsf{K}(D)$, so that m subsumes e but not g. Thus C is an open cover of E. By compactness, C has a finite subcover for E. The union of the sets of this subcover has the desired properties. ▮

LEMMA 5.3. *Let m be a minimal model of* $\mathsf{fix}T_P$. *Then m is a model of P.*

Proof. Let m be a minimal model of $\mathsf{fix}T_P$. We must show that m is a model of P. Let $\theta \leftarrow \tau$ be a rule of P with $m \models \tau$. Since $\mathsf{fix}T_P$ is a logically closed theory, we know from Theorem 3.2 that the set of all models of $\mathsf{fix}T_P$ is a compact saturated set, which we choose as E in Lemma 5.2. In the same lemma, take $K = \uparrow\perp$, $g = m$, and take d to be an element of τ with $d \sqsubseteq m$. Consider the clause μL consisting of the minimal elements of the compact open set L in the conclusion of the lemma. Then $\mathsf{fix}T_P \models \mu L$, because $[\![\mathsf{fix}T_P]\!]$ is a subset of L. Since $\mathsf{fix}T_P$ is logically closed, we have $\mu L \in \mathsf{fix}T_P$. Now consider the following (unary) instance of the hyper-resolution rule:

$$\frac{\mu L; \quad l \in \mu L; \quad \theta \leftarrow \tau \in P \quad \{l\} \models \tau}{\theta \cup (\mu L \setminus \{l\})}$$

where l is the unique generator of L subsuming m. The clause $\theta \cup (\mu L \setminus \{l\})$ is in $T_P(\mathsf{fix}T_P)$ and so is $\mathsf{fix}T_P$. Therefore $m \models \theta \cup (\mu L \setminus \{l\})$. But l is the only element of μL subsuming m, so it must be that $m \models \theta$. ▮

LEMMA 5.4. $T_P(\overline{P}) \subseteq \overline{P}$.

Proof. If $Y \in T_P(\overline{P})$, then Y is a logical consequence of the HR-consequences of $\overline{P}$. We want to show that every model of P satisfies Y. Let e be a model of P. It is enough to show that e satisfies each clause of of the form $\theta \cup \bigcup_{1 \leq i \leq n}(X_i \setminus \{a_i\})$, where $X_i \in \overline{P}$, $\theta \leftarrow \tau \in P$, $a_i \in X_i$, and $\mathsf{mub}\{a_i \mid 1 \leq i \leq n\} \models \tau$, because Y is a logical consequence of these clauses. Since each of the X_i are in $\overline{P}$, $e \models X_i$ by definition. If $e \models \tau$ then $e \models \theta$ because e is a model of P, so we are done. So suppose that it is not the case that $e \models \tau$. Then, because $\mathsf{mub}\{a_i \mid 1 \leq i \leq n\} \models \tau$, there must be some a_i so that that a_i does not subsume e. Since $e \models X_i$, it must be the case that $e \models X_i \setminus \{a_i\}$, and we are again done. ▮

THEOREM 5.5. $\overline{P} = \mathsf{fix}T_P$.

Proof. The inclusion $\mathsf{fix}T_P \subseteq \overline{P}$ follows from Lemma 5.4 by mathematical induction and the familiar formula for the least fixed point of T_P.

For the other direction, we prove that if $X \in \overline{P}$, then $\mathsf{fix}T_P \models X$. (This implies that $X \in \mathsf{fix}T_P$ since $\mathsf{fix}T_P$ is logically closed.) It is sufficient, by Lemma 5.1 and Theorem 3.2, to show that any minimal element of the compact saturated set $[\![\mathsf{fix}T_P]\!]$ (i.e., a minimal model of $\mathsf{fix}T_P$) is a model of X. So let $m \in D$ be a minimal model of $\mathsf{fix}T_P$. By Lemma 5.3, m is a model of P, so $m \models X$ because $X \in \overline{P}$. ▮

6. NON-MONOTONIC REASONING

In defining the concept of logical consequence of a program, we treated rules of the form $\theta \leftarrow \tau$ as *material implications*. Such material implications cannot be translated into clauses, because they do not have the persistence property – the set of models of a program need not be upward closed. This "non-monotonicity" is the essence of Zhang-Rounds *power default reasoning* [ZR97a]. Consider, for example, the following program, expressing that birds fly, penguins are birds, but penguins do not fly:

$$\begin{aligned} \{b\} &\leftarrow \{p\}; \\ \{f\} &\leftarrow \{b\}; \\ \{\overline{f}\} &\leftarrow \{p\}. \end{aligned}$$

This program is in the shorthand for the logic of partial truth assignments, as in Section 2. The partial assignment bf is a model of the program, but the more specific bpf is not. Moreover, in nonmonotonic reasoning, we would like the clause $\{bf\}$ to be a non-monotonic consequence of the clause $\{b\}$ (i.e., birds normally fly).

In this section we introduce a non-monotonic version of the T_P operator, called the *extension* operator, owing its first incarnation to Reiter [Rei80]. This will allow us to define the concept of non-monotonic consequence, and in addition, will handle problems with inconsistent programs such as the two-rule program

$$\{p\} \leftarrow \{\bot\}; \;\; \{\overline{p}\} \leftarrow \{\bot\}.$$

We give both a syntactic and a semantic version of the operator. The semantic version, restricted to Scott domains, appeared (in a different guise) in [RZ97] and [ZR97a]; the syntactic version is new.

We now think of a logic program as a set of *default rules*. One starts with a logically closed theory as "given information", applies a version of the hyper-resolution rule modified to check for consistency of rule application, and thereby derives possibly several logically closed theories called *extensions* of the given theory. A clause is a non-monotonic consequence of the given theory iff it is in all extensions of the theory.

To begin, consider a *unary* version of the hyper-resolution rule

$$\frac{X; \quad a \in \mu X; \quad \theta \leftarrow \tau \in P; \quad a \models \tau}{\theta \cup (X \setminus \{a\})}$$

We define the associated unary program-resolution operator U_P on theories as follows:

$$U_P(T) = Cn\{(\theta \cup (X \setminus \{a\})) \mid \; X \in T, \theta \leftarrow \tau \in P, a \in \mu X, \text{and } a \models \tau\}.$$

By the proof of Lemma 5.3, U_P has the same least fixed point as T_P, namely $\overline{P}$. Note that in particular we can choose the element a to be minimal in the clause X. We focus on this version of the unary rule for the rest of the section.

EXAMPLE 6.1. Let T be the theory in 3-valued logic generated by the clause $p \vee q$ (officially, the clause $\{p, q\}$). Consider the program given by the two rules $r \leftarrow p$ and $r \leftarrow q$. Then $U_P(T)$ contains the clauses $p \vee r$ and $q \vee r$, and $U_P^2(T)$ contains the clause r.

DEFINITION 6.1. *Say that a clause Y is* consistent *with a theory E if they have a model in common. Write $Y : E$ in this case.*

In the next definition we limit the set of consequences of the unary rule to those which happen to be consistent with a theory E.

DEFINITION 6.2. *Fix a program P and let W and E be logically closed theories. A clause of the form $\theta \cup (X \setminus \{a\})$ is said to be an E-*consistent consequence *of W if there is a rule $\theta \leftarrow \tau \in P$ with $X \in W$, $a \in X$, $a \models \tau$, and $(\theta \cup (X \setminus \{a\})) : E$.*

DEFINITION 6.3 (Syntactic extensions). *Let T and E be logically closed theories. Define $\Gamma_P(T, E)$ to be the smallest logically closed theory W such that (i) $T \subseteq W$ and (ii) every E-consistent consequence of W is in W. If E is a consistent theory, and $\Gamma_P(T, E) = E$, we say that E is a* (syntactic) extension *of T.*

EXAMPLE 6.2. Let T be the theory generated by $\{p, q\}$ (the official version of $p \vee q$) and consider the program P of the previous example. If E is generated by $\{pr, qr\}$ then $\Gamma_P(T, E) = E$. In classical notation, E is generated by the formula $p \wedge (q \vee r)$. The clause $\{r\}$ is a logical consequence of the clause $\{pr, qr\}$, and so is in E. It is a nonmonotonic consequence of $\{p, q\}$.

We are going to tie this notion of an extension to the semantic version of extensions found in [ZR97a] and in [RZ97]. We first translate default (aka logic) programs to the compact saturated setting, obtaining a semantic version of the notion of E-consistent consequence.

DEFINITION 6.4. *Let P be a program, L, N be compact saturated, and K compact open. A compact open set of the form $\uparrow(\mu\theta \cup (\mu K - \{a\}))$ is an N-consistent semantic consequence of L if there is a rule $\theta \leftarrow \tau \in P$ with $K \supseteq L$, $a \in \mu K$, $a \models \tau$, and $\uparrow(\mu\theta \cup (\mu K - \{a\})) \cap N \neq \emptyset$.*

DEFINITION 6.5 (Semantic extensions.). *Fix a program P, and let M and N be compact saturated. The set $\eta_P(M,N)$ is the largest compact saturated set L (i) contained in M, and (ii) such that every N-consistent semantic consequence of L contains L. We say that the nonempty c.s.s. N is a* semantic extension *of M if $\eta_P(M,N) = N$.*

We can compare these two notions of extension easily, using duality:

THEOREM 6.6. *Fix a program P. Then for any logically closed theories T and E, we have that E is a syntactic extension of T if and only if $[\![E]\!]$ is a semantic extension of $[\![T]\!]$, with respect to P.*

Proof. We simply show that $[\![\Gamma_P(T,E)]\!] = \eta_P([\![T]\!],[\![E]\!])$. First we prove that $[\![\Gamma_P(T,E)]\!] \subseteq \eta_P([\![T]\!],[\![E]\!])$ by showing that (i) $[\![\Gamma_P(T,E)]\!] \subseteq [\![T]\!]$, and (ii) $[\![\Gamma_P(T,E)]\!]$ is closed under $[\![E]\!]$-consistent semantic consequence. Part (i) follows from duality since $\Gamma_P(T,E) \supseteq T$. For part (ii), let K be compact open, $K \supseteq [\![\Gamma_P(T,E)]\!]$, $a \in \mu K$, and $\theta \leftarrow \tau$ be a rule of P with $a \models \tau$ and $\uparrow(\mu\theta \cup (\mu K - \{a\})) \cap [\![E]\!] \neq \emptyset$. Take the clause X to be μK. Then $(\theta \cup (X \setminus \{a\})) : E$ and so it is in $\Gamma_P(T,E)$. Thus $\uparrow\theta \cup (X \setminus \{a\}) \supseteq [\![\Gamma_P(T,E)]\!]$, as we wanted.

Conversely, we can show that $Th(\eta_P([\![T]\!],[\![E]\!]))$ is a theory (i) containing T, and (ii) closed under E-consistent syntactic consequence. This is sufficient to be able to apply duality, and to finish the proof. The details are straightforward. ∎

The definition of extensions in [RZ97] is slightly different from that presented above. The rest of this section is devoted to showing how the definitions are related. A convenient way to do this is to present the general definition of extensions in a Scott domain. This definition applies to our case, because the Smyth powerdomain of a coherent algebraic dcpo is indeed a Scott domain. We start with a general presentation of defaults. In the following definition we intend to specialize A to be the Smyth powerdomain of a coherent algebraic dcpo D.

DEFINITION 6.7. *Let $(A, \sqsubseteq)$ be a Scott domain. A default set is a subset Δ of $\mathsf{K}A \times \mathsf{K}A$. We call a pair $(a,b) \in \Delta$ a (normal) default.*

The motivation for this definition is from Reiter's default logic, where the Scott domain in question is the collection of logically closed classical propositional theories, and where a and b are propositional theories generated by single formulas. A pair (a,b) is thought of as a normal default rule

$$\frac{a : b}{b}$$

in Reiter's notation, with the intuitive meaning that if a is a formula in a theory, and b is consistent with a (guessed) theory, then b is in the theory too.

Our notions of syntactic and semantic defaults can be presented using the definition of default set. Here we take the Scott domain A to be either the space of consistent theories over D, or the (isomorphic) space of nonempty compact saturated sets over D. Fix a program P. Then the syntactic default set determined by P is the set

$$\{(Cn(X), Cn(Y)) \mid \text{ for some } (\theta \leftarrow \tau) \in P,\\ a \models \tau, \text{ and } a \in \mu X, \text{ we have } Y = \theta \cup (X \setminus \{a\})\}.$$

Similarly, P determines the semantic default set

$$\{(K, L) \mid K \in \mathsf{K}\Omega D \text{ and for some } (\theta \leftarrow \tau) \in P,\\ a \models \tau, \text{ and } a \in \mu K, \text{ we have } L = \uparrow(\mu\theta \cup (\mu K \setminus \{a\}))\}.$$

These two default sets are isomorphic under duality.

EXAMPLE 6.3. There is a big difference between defaults in the standard sense of default logic, and our clausal syntactic defaults, which are normal defaults in the domain of clausal theories, generated by the resolution rule. Consider, for example, the defaults generated by the program P with two rules $r \leftarrow p$ and $r \leftarrow q$. Using standard default logic, this would be translated to the two default rules

$$\frac{p : r}{r} \text{ and } \frac{q : r}{r}. \tag{1}$$

But using the unary resolution operator U_P, we put $(\{p, q\}, \{r, q\})$ and $(\{p, q\}, \{p, r\})$ into the syntactic default set, among (infinitely) many other such pairs. In Reiter's notation, we would generate the clausal defaults

$$\frac{(p \vee q) : (r \vee q)}{(r \vee q)} \text{ and } \frac{(p \vee q) : (p \vee r)}{(p \vee r)}. \tag{2}$$

It is this behavior which enables us do do default "reasoning by cases". In the present example, notice that if we start with the clause $\{p, q\}$, representing the formula $p \vee q$, the rules in (1) do not apply. However, the rules in (2) do apply, allowing the adjunction of the formulas $r \vee q$ and $p \vee r$. The formula r is a logical consequence of these two formulas together with $p \vee q$, so that we may infer r as a "nonmonotonic consequence" of $p \vee q$. Notice, though, that there is nothing non-monotonic about this situation. If T is a given theory, and P is a program, we may consider the subdomain $\uparrow T$ of the domain of theories, and compute the least fixpoint of T_P, or equivalently U_P, with respect to this domain. The result will be the set of consequences of P with respect to the starting theory T. The rules exemplified by (1) are thus not really the right rules with which to do deductions starting with disjunctive information. We should be using clausal inference rules instead. (See below for more on the property of reasoning by cases – Theorem 6.12.)

The notion of default set enables the definition of a general extension operator in a Scott domain. Again, A is intended to be the Smyth powerdomain of an underlying domain D. Here $A^\top$ is the Scott domain A augmented with an "inconsistent" top element. Also, by $x : y$ we mean that x and y have an upper bound in A.

DEFINITION 6.8. *The function $\xi(x, e)$ is the least element y of $A^\top$ such that (i) $x \sqsubseteq y$; and (ii) for any default $(a, b) \in \Delta$, if $a \sqsubseteq y$ and $b : e$, then $b \sqsubseteq y$. We paraphrase (ii) by saying that y is closed under e-consistent default application. We call $e \in A$ an* extension *of $x \in P$ if $\xi(x, e) = e$.*

It should be clear that the abstract ξ operator specializes to the syntactic Γ and the semantic extension operator η using the domain of theories and the domain of compact saturated sets respectively.

We still need to consider the form of defaults used in [RZ97]. This is presented using the notion of an *update*.

DEFINITION 6.9. *Let K be a compact open subset of he coherent algebraic dcpo D. If θ is a clause, and $a \in \mu K$, we say that*

$$K[a \leftarrow a \cap \theta] = K \cap \uparrow(\mu\theta \cup (\mu K - \{a\})).$$

is an update *of K. This update is said to be based on a rule $\theta \leftarrow \tau$ of P if $a \models \tau$ and $a \in \mu K$.*

This definition says that an update of the set K of models of a clause consists of intersecting K with the set of models of a succedent of (the unary version of) the hyper-resolution rule.

As above, we can form a default set based on updates:

$$\{(K, L) \mid K \in \mathsf{K}\Omega D \text{ and } L \text{ is an update of } K \text{ based on a rule } \theta \leftarrow \tau \text{ in } P\}.$$

The update default set gives rise to the same semantic extensions as does the semantic default set determined by P. This is a consequence of an observation about general default sets in a Scott domain, found in [ZR97b, Proposition 2.3]:

PROPOSITION 6.10. *Let Δ be a default set in a Scott domain. Form the set*

$$\Delta' = \{(a, a \sqcup b) \mid (a, b) \in \Delta \wedge a : b\}.$$

Then e is an extension of x with respect to Δ iff it is an extension of x with respect to Δ'.

This result shows that using updates gives us the same notion of extension as we get from our original definition of a semantic extension, because in the domain of compact saturated sets, $K \cap L$ is the least upper bound of the consistent compact open sets K and L.

The definitions in this section allow us to generalize two theorems in [RZ97] from Scott domains to coherent algebraic dcpos. For the first theorem, we define (relative to a program P over the coherent domain D) an element $e \in D$ to be *safe* for a compact saturated set M if it is both in M and is a model of P. An extension N of M is safe if it has at least one safe element as a member.

THEOREM 6.11 (Dichotomy Theorem). *Fix a default program. If there exists an element e which is safe for M, then there will be a unique safe semantic extension N of M. If M does not contain a safe element, then the (multiple) extensions of M will all be of the form $\uparrow g$, for certain $g \in M$.*

THEOREM 6.12 (Extension Splitting Theorem). *Fix a default program. Consider two nonempty compact saturated sets M_1 and M_2. Then any semantic extension N of $M_1 \cup M_2$ can be written as a union $N = N_1 \cup N_2$, where N_1 and N_2 are either empty or extensions of M_1 and M_2, respectively.*

It is not hard to extend the proofs of these theorems given in [RZ97] to coherent algebraic dcpos. One needs only the Interpolation Theorem (Theorem 5.2) and some general results about extensions in a Scott domain.

Theorems 6.11 and 6.12 provide a syntactic version of the result that the problem of deciding non-monotonic consequences of a standard formula ϕ in Kleene's three-valued logic is a problem complete for co-NP(3), the third level of the Boolean hierarchy (see [ZR97a]). Using the Extension Splitting Theorem, we can obtain the principle of *reasoning by cases*: if X is a non-monotonic consequence of Y and also of Z then it is a consequence of $Y \cup Z$. (This law fails for standard default logic – see Example 6.3.)

We close the section by applying Theorem 6.11 to relate monotonic and non-monotonic inference.

THEOREM 6.13. *Let P be a consistent logic program. Then the set fixT_P of logical consequences of P is the unique extension of the (logical closure of) the theory $\{\{\bot\}\}$ with respect to P, considered now as a set of default rules.*

Proof. If P has a model, then the set $D = \uparrow \{\bot\}$ is safe; so by the Dichotomy Theorem it will have a unique (semantic) extension. By duality, the theory $\{\{\bot\}\}$ will have a unique syntactic extension, say E. Using the definition of the syntactic extension operator, we see that any application of the unary rule will automatically be consistent with E, so applying this rule will b the same as applying the ordinary rule U_P. But applying U_P repeatedly to $\{\{\bot\}\}$, interleaved with clausal inference steps, generates exactly the clauses in $\mathsf{fix}\mathsf{T}_P = \overline{P}$. ▮

7. CONCLUSION

Although we have proposed a general framework for logic programming, it remains to be seen whether or not significant progress can be made towards implementable logic programming languages based on the framework. We are currently working towards at least a first-order instantiation of our system. A first step has been made to build the clausal logic approach in the concrete setting of information systems [ZR99]. Whether or not this can be extended to higher order logics is debatable – one needs to marry an explicit syntax for higher-order logic to the abstract model theory. Partial logic, however, may be one key to progress – higher-order logic has traditionally not been partial. Our work suggests that *clausal* partial logic may be a way to proceed. Further evidence for this can be found in the recent work of Coquand and Zhang which shows that clausal logic and the clausal resolution rule arise naturally in the spectral theory of coherent locales [CZ00]. One may also wish to work within the collection of Scott domains. There the least upper bound operation in the underlying domain corresponds to unification. Finally, it would be desirable to integrate a theory of types into our framework.

Our version of default reasoning has been shown in [ZR97a] to have better complexity properties than does standard default logic, for the propositional calculus. We have implemented this system in [KRZ98]. The results show that default inference in our system is very close to standard propositional inference, when one considers random instances. It also competes well with inference engines based on more standard approaches, for specific benchmark problems. However, the overall utility of our approach still needs to be compared with the method of *stable models* [BG94]. There are also reasonably large-scale applications of non-monotonic methods in the business domain [Mor98] and in natural language semantics [LBAC96]; it remains to be seen whether or not a stable model theory for general domains would provide a competitive framework for these application areas.

REFERENCES

Abr91. S. Abramsky, *Domain theory in logical form*, Annals of Pure and Applied Logic **51** (1991).

AJ94. S. Abramsky and A. Jung, *Domain theory*, Handbook of Logic in Computer Science (S. Abramsky, D. Gabbay, and T. S. E. Maibaum, eds.), vol. 3, Clarendon Press, 1994.

Apt90. K. R. Apt, *Logic programming*, Handbook of Theoretical Computer Science, vol. B, MIT Press/Elsevier, 1990, pp. 495–571.

BG94. C. Baral and M. Gelfond, *Logic programming and knowledge representation*, Journal of Logic Programming (1994), 73–148.

CZ00. T. Coquand and G. Q. Zhang, *Sequents, frames, and completeness*, Computer Science Logic 2000 (P. Clote and H. Schwichtenberg, eds.), Lecture Notes in Computer Science, vol. 1862, 2000, pp. 277–291.

HM81. K. Hofmann and M. W. Mislove, *Local compactness and continuous lattices*, Lecture Notes in Mathematics, vol. 871, 1981, pp. 209–240.

KP94. Klaus Keimel and Jan Paseka, *A direct proof of the Hofmann-Mislove theorem*, Proc. Amer. Math. Soc. **120** (1994), no. 1, 301–303.

KRZ98. Eric Klavins, W. Rounds, and Guo-Qiang Zhang, *Experimenting with power default reasoning.*, Proceedings of AAAI National Conference on Artificial Intelligence, 1998.

LBAC96. A. Lascarides, E. J. Briscoe, N. Asher, and A. Copestake, *Order independent persistent typed default unification*, Linguistics and Philosophy **19.1** (1996), 1–89.

LMR92. J. Lobo, J. Minker, and A. Rajasekar, *Foundations of disjunctive logic programming.*, MIT Press series in logic programming, Cambridge, MA, 1992.

Mor98. L. Morgenstern, *Inheritance comes of age: applying nonmonotonic techniques to problems in industry*, Artificial Intelligence **103** (1998), 237–271.

NM98. G. Nadathur and D. Miller, *Higher-order logic programming.*, Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 5, Oxford Univ. Press, New York, 1998, pp. 499–590.

Rei80. Raymond Reiter, *A logic for default reasoning*, Artificial Intelligence **13** (1980), 81–132.

RZ97. W. Rounds and Guo-Qiang Zhang, *Power defaults*, Proceedings of 4th International Conference on Logic Programming and Non-monotonic Reasoning, July 1997, Springer Lecture Notes in AI, number 1265.

Sco82. Dana S. Scott, *Domains for denotational semantics*, Lecture Notes in Computer Science 140, 1982.

Smy78. M. B. Smyth, *Powerdomains*, Journal of Computer and System Science **16** (1978), no. 1, 23–36.

Smy83. M. B. Smyth, *Powerdomains and predicate transformers: a topological view*, Lecture Notes in Computer Science, vol. 298, 1983, pp. 236–253.

Vic89. S. Vickers, *Topology via logic*, Cambridge University Press, 1989.

Win83. G. Winskel, *A note on powerdomains and modality*, Lecture Notes in Computer Science 158, 1983.

Zha91. Guo-Qiang Zhang, *Logic of domains*, Birkhauser, Boston, 1991.

ZR97a. G. Q. Zhang and W. C. Rounds, *Complexity of power default reasoning*, Proceedings of 113th IEEE Symposium on Logic in Computer Science, 1997, pp. 152–169.

ZR97b. Guo-Qiang Zhang and W. Rounds, *Defaults in domain theory*, Theoretical Computer Science **177** (1997), no. 1, 155–182.

ZR99. G. Q. Zhang and W.C. Rounds, *Semantics of logic programs and representation of smyth powerdomain*, Proceedings of the 1st International Symposium on Domain Theory, 1999, (to appear).