Power Defaults (preliminary report)

Guo-Qiang Zhang[1] and William C. Rounds[2]

[1] Department of Computer Science
University of Georgia, Athens, Georgia 30602, U.S.A.
[2] Artificial Intelligence Laboratory
University of Michigan, Ann Arbor, Michigan 48109, U.S.A.

1 Introduction

1.1 Synopsis

This paper is concerned with using Scott's *domain theory* to generalize and improve the kind of default reasoning introduced by Reiter [11]. We can define a new class of default logics, each of which is totally faithful to Reiter's ideas, and also has much better semantic properties. The key is to integrate defaults with the *model theory* of the logic, instead of keeping them as pseudo-deductive rules. To do this adequately, we have to keep track of disjunctions model-theoretically. This entails using the *Smyth powerdomain* to define defaults, whence the title of our paper. Specifically, in this paper we indicate a spectrum of default logics which obey the law of "reasoning by cases", something which fails in Reiter's original system. We consider Reiter's *extensions*, which are fixed points of a certain non-monotonic operator on an *arbitrary* Scott domain D induced by a normal default set over that domain. We specialize this definition to the Smyth powerdomain, which consists of all nonempty compact saturated sets in the Scott topology over D, ordered by reverse subset inclusion.

We have two main results. Both of them concern the structure of extensions in the Smyth powerdomain, given that the set of "power defaults" is obtained from "semantic default constraints" via "updating", an idea due to Libkin [7].

The first main result, called the *Dichotomy Theorem*, shows that extensions in the Smyth powerdomain fall into two classes: either there is a unique *safe* extension of a given compact saturated set, or else the (multiple) extensions are each generated by a singleton element of the underlying domain. (A safe extension is one which contains some element of the domain which is invariant under the action of default rules.)

The second result is called the *Extension Splitting Theorem.* It says that any extension E of a union of compact saturated sets U and V can be written as a union $E = E_U \cup E_V$, where E_U and E_V are either empty or extensions of U and V respectively. We give a proof of this theorem as a consequence of the Dichotomy Theorem. Both of these results hold in arbitrary Scott domains. The Splitting Theorem implies that default logics based on power default semantics will allow reasoning by cases.

We illustrate our methods using propositional logic. Since the results are for arbitrary domains, though, we can define a spectrum of logics each of which can be made non-monotonic. For example, one can easily give a domain-theoretic partial model theory for first-order logic. When we add power defaults to this partial model theory, we seem to get a semantically reasonable account of *open defaults*. (See our other work [12].) We mention also that in the propositional case, we have what seems to be a "fast" algorithm for deciding power skeptical non-monotonic entailment. The problem is co-NP(3)-complete, which means that we can express the problem as a certain Boolean combination of three NP languages, and no fewer will do. By contrast, standard propositional default logic entailment is complete for the second (Π) level of the polynomial hierarchy [5]. So our complexity is very much simpler. These results appear elsewhere [14]. However, we briefly sketch in this paper how, in the case of propositional logic, the Dichotomy Theorem implies the complexity results.

1.2 General introductory remarks

Much work in the area of logic programming has shown that appropriate semantics for such systems really must reduce eventually to the semantics of non-deterministic computation, as (roughly) exemplified by disjunctive logic programs. Traditionally, though, the semantics has had a distinct syntactic flavor, beginning with the traditional Herbrand models. When one introduces negation and non-monotonicity, this trend has continued, with many connections having been made between Reiter's default logic and logic programs with negation [2].

This paper concentrates on the problem of disjunctive default reasoning, but we work with disjunctive default information in a model-theoretic setting, not in a syntactic way. Our setting is a general framework for modelling the semantic values associated with structured programs of various kinds, called *denotational semantics*, where the values themselves live in spaces called *complete partial orders* or *domains*. We think of defaults as semantic values residing in such spaces. In an actual application, though, there will be a syntax for defaults which gives rise to the semantic defaults.

It will help in what follows to keep the example of propositional logic in mind. When we have a specific syntax to work with, we can start with a syntactic way of writing normal default constraints. Our notation is $\tau \leadsto \theta$, where τ and θ are ordinary formulas of our given logic. For very good reasons, we interpret these formulas in a three-valued setting. In the propositional case, this amounts to using Kleene's strong three-valued truth tables. We regard a set Σ of syntactic constraints as a program. The reader may temporarily think of $\tau \leadsto \theta$ as part of a disjunctive logic program where for each constraint, θ is the head, and where $\tau \wedge not(\neg\theta)$ is the body. The big difference comes in how we assign meanings to these programs.

We are interested in a starting formula ϕ; in logic programming, this would be a description of the "extensional" data. The meaning of ϕ is given by the set of its 3-valued partial models. In domain-theoretic terms, this would be a compact open set in the Scott topology of the domain D of partial truth assignments to the set of propositional variables, ordered pointwise. It is also an element of the

Smyth powerdomain of D. The meaning of the set of constraints Σ will be a very large set Δ of normal *semantic* defaults

$$\frac{K : L}{L}$$

where K and L are *also* compact open sets, constructed using the syntactic constraints in Σ. This construction is where we use the idea of updates; L will be obtained from K by updating minimal elements of K using syntactic constraints in a natural way.

The generality of our approach allows us to define default extensions using normal default sets where the constituents of defaults are arbitrary (compact) elements of any Scott domain. The Smyth powerdomain of any Scott domain is again such a domain. So we can use Reiter's definitions (originally for the domain of closed first order theories) word for word to build extensions *semantically* of the set of models of ϕ. To see if ψ is entailed, we just check that every semantic extension of $Mod(\phi)$ is less than or equal to $Mod(\psi)$ in the Smyth ordering. This is well-known to be reverse subset inclusion, so we want that every extension E is a subset of the set of models of ψ.

Another way to think of this semantics is by thinking of the set of default constraints Σ as inducing a non-monotonic and non-deterministic state transformer. Then non-monotonic entailment is just like Hoare logic, with ϕ being the pre-condition, and ψ the post-condition. The Smyth powerdomain was invented to explicate this idea in the case where the non-deterministic program induced a monotonic (continuous) function. Here we have neither monotonicity nor continuity, but the Smyth powerdomain together with Reiter's defaults gives us a robust semantics anyway.

Our two main results are the Dichotomy Theorem and the Extension Splitting Theorem; we give their proofs in Section 3. The Extension Splitting Theorem entails the Or law of Kraus, Lehmann, and Magidor [6]; and it does so for general Scott domains, not just the domains we have mentioned. The Or law says that if ψ is "default-entailed" as above from ϕ, and also from γ, then it is entailed from $\phi \vee \gamma$. This law fails in standard default logic, so our results provide a definite improvement. In the case of propositional logic, the Dichotomy Theorem also entails the law of Cautious Monotony. However, this law does not always hold in a general domain.

1.3 Previous work

Buneman and his students [3] pioneered the use of domain theory in database applications. Libkin's thesis [7] contains a very good overview. The Buneman school considers many more powerdomain constructions besides the Smyth powerdomain.

Marek, Nerode, and Remmel showed how the extension operator could be considered very generally, and represented all sorts of mathematical model constructions thereby [8]. Our paper follows their lead, but introduces domain theory as an additional crucial tool.

In artificial intelligence and natural language Pollard and Moshier [10] considered a number of ways to use powerdomains to model grammatical inference, though the first use of partial-order techniques in a setting related to artificial intelligence, was by Pereira and Shieber [9] in providing a semantics for disjunctive feature structures in computational linguistics. Our semantics was anticipated as well by Ait-Kaci [1], but (for his ϵ-terms) he did not use domain theory explicitly. Another implicit use of our semantics is in a paper by Gelfond *et al.* [4], who introduce a notion of *disjunctive defaults* in order to overcome certain problematic examples involving disjunctive reasoning in default logic.

The present paper is in fact a continuation of [13], in which we apply powerdomain theory to give a non-monotonic semantics for feature or attribute-value logic. One purpose of this paper is to show that the methods introduced in [13] are not as special as they might have seemed there. The Extension Splitting Theorem is a solution to the question on reasoning by cases left open in that paper.

The question of the general relationship of our model to the differing semantics for standard logic programs (disjunctive or not; with or without normal defaults; stationary, stable, etc., semantics) seems to need much more investigation. Our analysis indicates that there is no obvious direct connection. The reason seems to be that we take the radical step of incorporating defaults as semantic entities. This enables us to use the powerful modelling tool of denotational semantics – specifically the Smyth powerdonain for nondeterminism – as a tool for modelling non-monotonicity.

1.4 Plan of the paper

The next subsection is purely expository, illustrating the Smyth powerdomain in a very simple setting. We also illustrate the construction of normal defaults from syntactic constraints, using updating. Our logic in the subsection is 3-valued propositional logic.

We continue in the next full section with general concepts from domain theory. We define and characterize the Smyth powerdomain. We also introduce defaults in domain theory. We give the general definitions, and then turn to the way defaults will be defined over the Smyth powerdomain in general (not just in the case of finite domains.)

Section 3 sketches the proof of the Extension Splitting Theorem in the Smyth powerdomain. The final section contains concluding remarks, with references to some of our previous work.

1.5 Example from three-valued propositional logic

In three-valued logic, one uses *partial truth assignments* as models of propositional formulas. These truth assignments can be considered as partial functions from the set of propositional variables to $\{0, 1\}$. They can be partially ordered pointwise in the standard ordering of partial functions, and form a Scott domain thereby. We use $\sqsubseteq$ as the name of this order. We represent assignments by writing them as strings of literals; for example, if we have three variables $\{p, q, r\}$, then $p\neg r$ is the partial assignment of 1 to p and 0 to r. A fundamental fact about Kleene's strong connectives is that if an assignment d satisfies a formula, then any h with $d \sqsubseteq h$ will satisfy it also.

We continue with an example showing the failure of reasoning by cases in standard default logic. We show how to interpret the example instead in the power setting. The example has already been discussed by Gelfond *et al.*, using disjunctive defaults [4]; our treatment just goes a few steps further.

Reasoning by cases. Consider the standard default theory (D, W) where W is the logical closure of $(p \vee q)$, and we have the two default rules

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

It seems as though r should be a default consequence of this theory, but of course neither of the two defaults apply, since their preconditions are not in W. We show how to code this example in our setting.

Start with the formula $p \vee q$, and represent it by the set of its models in three-valued logic. The variable set in question here is $\{p, q, r\}$, and the two $\sqsubseteq$-minimal models of the formula are the assignments p and q. The set of models of $p \vee q$ is then just the upward closure (in the $\sqsubseteq$ order) of $\{p, q\}$, written $\uparrow\{p, q\}$. (In general $\uparrow X$ is the set of all d such that $x \sqsubseteq d$ for some $x \in X$; similarly for $\downarrow X$.)

The two defaults above could be expressed syntactically as $p \rightsquigarrow r$ and $q \rightsquigarrow r$, the first of these reads as "normally, if p then r". However, this relation is not going to be the same as the Kraus, Lehmann notion $\mid\!\sim$. This latter symbol will be used for skeptical default consequence. In the example, we will derive $(p \vee q) \mid\!\sim r$. The way that this happens is to let the defaults act semantically on the generators p and q of the set $\uparrow\{p, q\}$. For example, the default $p \rightsquigarrow r$ will be able to improve the generator p to pr. This results in the shrunken set $\uparrow\{pr, q\}$. Similarly, we could have "updated" the other generator q, resulting in the set $\uparrow\{p, qr\}$. We then intersect these two sets, and obtain $\uparrow\{pr, qr\}$. This set turns out to be the unique semantic extension of $\{p, q\}$, and it is contained in the set $\uparrow\{r\}$, so r is a default consequence. The next subsection gives full details.

1.6 Semantic constraints and updates

We are working in the domain of upper sets of truth assignments, so normal defaults will be objects of the form $\dfrac{K : L}{L}$, where K and L are upward-closed sets of partial truth assignments. We will insist, however, that these defaults (which might as well be written as pairs (K, L)) be derived from our syntactic default constraints. The difficulty to keep in mind is that these defaults should permit reasoning by cases.

We have a two-stage process. The intermediate stage is a construct called a *semantic constraint.* In the example (2), we would create two pairs

$$(p, \uparrow\{r\}) \text{ and } (q, \uparrow\{r\})$$

In such a constraint, the set on the right is a Smyth powerdomain element, and the element on the left is just a truth assignment. If we were working with a more general constraint $\tau \rightsquigarrow \theta$, we would create several semantic constraints of the form (d, θ) where d is a minimal satisfier of τ, and θ is used ambiguously to denote the upward closure of the set of minimal models of the syntactic formula θ.

The final stage in the creation of normal power defaults is to lift semantic constraints so that the first element of a pair is a Smyth powerdomain element. We do this as follows. If (d, θ) is a semantic constraint, then we add a large number of defaults to our system. Take any K such that d subsumes a generator b of K. Consider the upward closure of all of the generators except b, and add in the set $(\uparrow\{b\}) \cap \theta$. This is called *updating* K by replacing b by $b \cap \theta$, and is notated $K[b \leftarrow b \cap \theta]$. Then our defaults are all pairs $(K, K[b \leftarrow b \cap \theta])$, provided that the second element of the pair is not actually empty.

This "update semantics" says that if we have a disjunctive set K at some stage, and some generator b of K is subsumed by a definite description d, then we can have all the disjunctive possibilities we had before, except that the set $\uparrow b$ can be improved by intersecting it with the set θ, where (d, θ) is a semantic constraint derived from our syntactic presentation.

In our example, suppose that we have the set $K = \uparrow\{p, q\}$. Consider the syntactic default $p \rightsquigarrow r$. This gives rise to a semantic constraint $(p, \uparrow\{r\})$. Since p subsumes itself, we can update it in K. Thus $K[p \leftarrow p \cap r] = \uparrow\{pr, q\}$. Thus we derive the default pair $(\uparrow\{p, q\}, \uparrow\{pr, q\})$. Another default pair might be $(\uparrow\{p, q, r\}, \uparrow\{p, qr, r\})$, where we have updated q using the semantic constraint $(q, \{r\})$. Notice that $\uparrow\{p, qr, r\}$ is the same as $\uparrow\{p, r\}$, since $r \sqsubseteq qr$.

This completes the construction of semantic defaults. In the next section we give a complete formal treatment.

2 Domain Theory

We now begin an exposition of background material, which will be familiar to denotational semanticists, and can be skipped by them.

2.1 Basic definitions

A *partial order* is a set D with a binary relation on D which is reflexive ($x \sqsubseteq x$), transitive ($x \sqsubseteq y \,\&\, y \sqsubseteq z \Rightarrow x \sqsubseteq z$), and antisymmetric ($x \sqsubseteq y \,\&\, y \sqsubseteq x \Rightarrow x = y$). We read $x \sqsubseteq y$ as "x subsumes y."

A *directed* set of a partial order $(D, \sqsubseteq)$ is a non-empty subset $S \subseteq D$ such that $\forall s, t \in S \exists u \in S. s \sqsubseteq u \,\&\, t \sqsubseteq u$. A *complete partial order* (cpo) is a partial order $(D, \sqsubseteq)$ which has a least element $\perp$ and all least upper bounds for directed subsets. A *compact* element of a cpo $(D, \sqsubseteq)$ is an element $x \in D$ such that for any directed subset $S \subseteq D$ if $x \sqsubseteq \bigsqcup S$, then there is an $s \in S$ such that $x \sqsubseteq s$. The set of compact elements of a domain D is denoted $\mathsf{K}(D)$. Note that for finite cpos, all elements are compact. A cpo $(D, \sqsubseteq)$ is *algebraic* if for all $x \in D$, $\{ e \sqsubseteq x \mid e \in \mathsf{K}(D) \}$ is directed and $x = \bigsqcup\{e \sqsubseteq x \mid e \in \mathsf{K}(D)\}$. When $(D, \sqsubseteq)$ is algebraic and $\mathsf{K}(D)$ is countable, D is said to be ω-algebraic. A set X is *consistent* if it has an upper bound. When the set $\{ a, b \}$ is consistent, we write $a \uparrow b$. A cpo is a *Scott domain* if it is ω-algebraic and *consistently complete*, *i.e.*, every consistent subset X of D has a least upper bound $\bigsqcup X$.

If $\top$ is a new symbol, we define the domain $D^\top$ by putting $\top$ above all the elements of D. Note that in domain theory, $\bot$ denotes "no information", and $\top$ denotes "inconsistency".

Example 1. **The Scott domain of closed first-order theories:** Let a proof system for first-order logic be given. Take D to be the collection of all deductively closed consistent sets T of sentences, ordered by inclusion. The bottom element of this domain is the set of logically valid sentences; the finite elements are the theories generated by finite sets (in fact singleton sets) of formulas. This is the domain over which standard default logic is defined.

The notion of compact element plays a crucal role in domain theory. As can be seen from Example 1, such elements are an abstraction of the idea of a finite amount of accessible information. Requiring any element to be the least upper bound of finite elements means in some sense that every element is abstractly computable. In fact the notion of computable function itself has an abstract formulation as the notion of a *continuous* function. This is just a function from one domain to another which preserves least upper bounds. Such a function (say from a domain D to itself) can be written as follows. For any $d \in D$,

$$f(d) = \bigsqcup\{b \in \mathsf{K}(D) \mid (\exists a \in \mathsf{K}(D))(a \sqsubseteq d \text{ and } b \sqsubseteq f(a)\}.$$

This observation is a motivating factor in thinking of a set of semantic defaults as a set of ordered pairs of the form (a, b) which "try" to "support" the graph of a continuous function.

We next present some topological facts. Every domain (in fact algebraic cpo) has a "natural" positive logic associated with it. This logic is given by the *Scott topology* of the domain.

Definition 1. Let $(D, \sqsubseteq)$ be a cpo. A set $U \subseteq D$ is said to be **Scott open** if for any directed $L \subseteq D$, we have $\bigsqcup L \in U$ iff $U \cap L \neq \emptyset$.

One checks readily that under this definition of "open", that the collection of open subsets of a cpo forms a topological space. Such a space must contain $\emptyset$ and D, and be closed under finite intersections and arbitrary unions. More importantly, though, we think of sets in a topology like this as "observable" predicates of domain elements. By "observable", we mean that if an element has the property given by the set U, then in a finite amount of time we can verify that it indeed has the property. In the above definition, the directed set L is usually a sequence of compact elements "approaching" its least upper bound, so that having a finite element of L with the property U means that we have verified membership in U in a finite time. Clearly this definition is an abstract version of recursive enumerability.

We also have occasion to use the topological definition of compactness: every covering of K by open sets has a finite subcovering.

It is straightforward to prove the following in any algebraic cpo D.

Theorem 2. Compact open sets in the Scott topology

1. *For each compact element $f \in D$, the principal filter $\uparrow f = \{u \mid f \sqsubseteq u\}$ is open.*
2. *Every compact open set K is a finite union of such compact principal filters.*

$\square$

Example 2. Let D be the set of all partial truth assignments from a countable set V of variables to $\{0, 1\}$. Technically this set can be expressed as the set of all functions from X to $\{0, 1, \perp\}$, where $\perp \sqsubseteq 0$ and also $\perp \sqsubseteq 1$. Such functions can be ordered pointwise in the obvious way. The compact elements of D are the truth assignments which are non-$\perp$ on a finite set of variables. If ϕ is a propositional formula, then the set of satisfiers of ϕ, using Kleene's strong truth tables, is a compact open set in the Scott topology over D.

Our purpose in introducing the Smyth powerdomain is to have an explicit representation of the disjunction operation between elements of a Scott domain; this corresponds to having a "nondeterministic" model of computation, but one which has a definite logical flavor. We summarize the construction. For the present we return to the general setting of Scott domains. As a preliminary to the next definition, we say that a subset S of a domain D is *saturated* if it is the intersection of all open sets which contain it. (Every compact open set is therefore trivially saturated.)

Definition 3. The Smyth powerdomain $P^\sharp(D)$ of a Scott domain $(D, \sqsubseteq)$ is the domain consisting of all nonempty Scott-compact saturated subsets of D, ordered by reverse subset inclusion. The basis elements are the nonempty compact open subsets of D. Moreover, the domain $P^\sharp(D)^\top$ is obtained by adjoining the empty set as an extra compact set.

This is not the original definition of the Smyth powerdomain, but it is equivalent to it, and it is what we use in our proof of the Splitting Theorem. Since we often with finite cpos, it is useful to keep in mind that for a finite Scott domain D, the Smyth powerdomain $P^\sharp(D)$ simply consists of nonempty, upward-closed sets ordered by reverse inclusion. (A subset X is upward-closed if $X = \uparrow X$.)

2.2 Default domain theory

First, recall that if $(D, \sqsubseteq)$ is a Scott domain, we define a new domain $D^\top$ by adjoining a new element $\top$ to D, and put $D \sqsubseteq \top$ for $x \in D$.

Definition 4. Let $(D, \sqsubseteq)$ be a Scott domain. A default set is a subset Δ of $\mathsf{K}(D) \times \mathsf{K}(D)$. We call a pair $(a, b) \in \Delta$ a (normal) default and think of it as a rule $\dfrac{a : b}{b}$.

Example 3. Consider Example 1, the domain of deductively closed sets of sentences. Then a default consists of a pair $(\overline{\varphi}, \overline{\psi})$, where φ and ψ are sentences, and $\overline{\theta}$ is the deductive closure of the singleton set $\{\theta\}$. This corresponds exactly to the normal default rule

$$\frac{\varphi : \psi}{\psi}$$

in the sense of Reiter.

We remind the reader at this point that we also want to think of a default set as giving us the graph of a continuous function. Then the definition of the extension operator, to be presented next, can be seen as trying to compute the least fixed point of this function. We thus present the general version of Reiter's operator (which he called Γ), used to build extensions of closed theories. In his case, x would be a theory W, e would be a theory E, and $\phi(D, E)$ would be the smallest theory containing W and closed under E-consistent application of user-specified (normal) default rules in a set Δ.

Definition 5. The function $\eta(x, e)$ is the least element y of $D^{\top}$ such that (i) $x \sqsubseteq y$; and (ii) for any default $(a, b) \in \Delta$, if $a \sqsubseteq y$ and $b \uparrow e$ (in D), then $b \sqsubseteq y$. We paraphrase (ii) by saying that y is invariant under e-consistent default application.

Reiter provides a way to construct the function η iteratively. This can be imitated in general Scott domains as well.

Lemma 6. *Let $(D, \sqsubseteq)$ be a Scott domain. Let Δ be a default set in D. We have that $\eta(x, e) = \bigsqcup_i \eta(x, y, i)$, where $\eta(x, e, 0) = x$, and*

$$\eta(x, e, i + 1) = \eta(x, e, i) \sqcup \bigsqcup \{b \mid (a, b) \in \Delta \;\&\; a \sqsubseteq \eta(x, e, i) \;\&\; b \uparrow e\}$$

for all $i \geq 0$.

Before passing to the definition of extension, we mention two properties of the η operator which appear in the folklore of default theories, but hold in general.

Lemma 7. *The function $\eta(x, e)$ is monotonic in x for fixed e.*

Lemma 8. *The function $\eta(x, e)$ is antimonotonic in e for fixed x.*

These lemmas can be easily proved using the iterative definition of η. We return to our main business.

Definition 9. We call $e \in D$ an extension of $x \in D$ if $\eta(x, e) = e$.

Clearly extensions over the domain of first-order theories are exactly Reiter's extensions.

The next theorem states some properties of extensions in general Scott domains. In each case the proof is obtained by imitating the proof for first-order theories.

Theorem 10. *Given a Scott domain D and a (normal) default set Δ, we have:*

1. *Extensions always exist.*
2. *If e is an extension of x then $e \sqsupseteq x$.*
3. *If e is an extension of x and f is an extension of e then $f = e$.*
4. *If e and f are distinct extensions of x then e and f have no upper bound in D.*
5. *If e is an extension of x and $y \sqsubseteq e$, then e is an extension of $x \sqcup y$.*

Finally, we can state a general notion of (skeptical) non-monotonic consequence over an arbitrary Scott domain, equipped with a default set Δ.

Definition 11. Let d be a finite element and x an arbitrary element of a Scott domain $(D, \sqsubseteq)$. We say that x is a non-monotonic consequence of d, and write $d \mid\!\sim x$, if $x \sqsubseteq e$ for every Δ-extension e of d.

Technically we should subscript $\mid\!\sim$ with the default set Δ, but the default set will always be clear from context.

2.3 Constraints and power defaults

Definition 12. A default constraint over the Smyth powerdomain $P^{\sharp}(D)$ of a Scott domain $(D, \sqsubseteq)$ is a pair (d, θ), where $d \in D$ and θ is a nonempty compact open subset of X.

For the next definition, if A is a compact open set, we let $\mu(A)$ be the finite set of generators of A. Thus $A = \uparrow\mu(A)$.

Definition 13. Let A be a compact open subset of D. If θ is a compact open set, and $a \in \mu(A)$, we say that

$$A[a \leftarrow a \cap \theta] = \uparrow(\mu(A) \setminus \{a\}) \cup (\uparrow a \cap \theta)$$

is an **update** of A, provided it is not empty.

This definition is due to Libkin [7]. It says that an elementary way to improve information in a compact open set is to improve one of its generators. Notice that "improving" a generator is represented by intersection here, corresponding to the Smyth ordering.

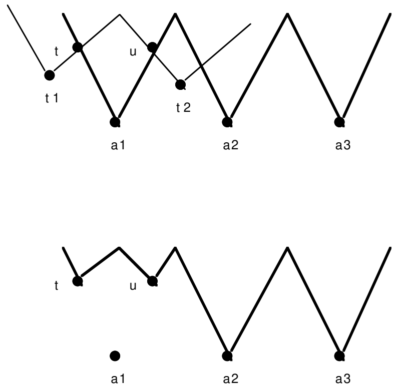

Fig. 1. Updating $a1 \in A$ by the set θ.

Example 4. A general picture of updating is presented in Figure 1. In this picture, the set A is the upward closure of the three element set $\{a1, a2, a3\}$. The set θ is indicated by lighter lines in the top half of the figure. It is generated by the set $\{t1, t2\}$. To update $a1$, we remove this element from the set of generators of A, and take the upward closure of $\{a1, a2\}$. This leaves the other two dark "cones" intact. Then we add back in the set $(\uparrow a1) \cap \theta$. This results in the two new generators t and u shown above $a1$ in the figure.

The next definition provides the actual lifting of default constraints to the powerdomain setting. Recall that a normal default set in any domain has to be a relation whose field is the set of finite elements of that domain.

Definition 14. Let Γ be a set of default constraints over $P^\sharp(D)$. The default set Δ_Γ determined by Γ is the set of pairs

$$\{(A, A[a \leftarrow a \cap \theta]) \mid A \in \mathsf{K}(P^\sharp(D));\ (d, \theta) \in \Gamma,\ \text{and } d \sqsubseteq a\}.$$

When clear from context, we omit the subscript Γ.

In general, for arbitrary Scott domains, we simply provide Γ as a set of semantic constraints. But when we have a syntax, we can derive the semantic constraints from syntactic ones as in the case of 3-valued logic. This is done by considering each pair $\tau \rightsquigarrow \theta$ of formulas, and adding all semantic constraints of the form (d, θ) where d is a minimal satisfier of τ, and θ is (by abuse of notation) the set of minimal satisfiers of the syntactic formula θ.

3 Main results

Our first objective is the Dichotomy Theorem. It requires the following basic definition.

Definition 15. Let D be a Scott domain, Γ be a set of semantic default constraints, and M a nonempty compact open subset of D. An element $e \in D$ is said to be **safe** if (i) $e \in M$, and (ii) for any semantic constraint $(d, \theta) \in \Gamma$, if $d \sqsubseteq e$ then $\uparrow e \cap \theta = \uparrow e$. Further, an extension E of M is said to be safe if there is a safe element $e \in E$.

The idea of the definition is that a safe element e is one to which none of the defaults can add any new information. Of course, the idea of safety is with respect to starting information; this is why we have the initial information M.

Remark. In the example of three-valued propositional logic, it is easy to see that if M is the set of minimal satisfiers of a formula ϕ, and Σ is a set of syntactic default constraints of the form (τ, θ), then an partial truth assignment e is safe if and only if (i) $e \models \phi$, and (ii) for each $\tau \rightsquigarrow \theta \in \Sigma$, if $e \models \tau$, then $e \models \theta$.

.

Our first main result characterizes extensions using the concept of safe elements.

Theorem 16 (Dichotomy Theorem). *Let Γ be a set of semantic constraints, and let Δ be the associated set of Smyth power defaults. If the compact saturated set M contains a safe element, then there will be a unique safe Δ-extension E 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 $g \in M$.*

The proof of the Dichotomy Theorem rests on several lemmas. The first of these is just a restatement for the Smyth powerdomain of the extension operator (Definition 5).

Lemma 17. *Let M and W be compact saturated sets. The set $\eta(M, W)$ is the inclusion-largest compact saturated set Y such that (i) $M \supseteq Y$; and (ii) whenever $(K, K[a \leftarrow a \cap \theta])$ is a default with $K \supseteq Y$ and $K[a \leftarrow a \cap \theta] \cap W \neq \emptyset$, then $Y = K[a \leftarrow a \cap \theta] \cap Y$.*

Now we turn to a lemma giving us information about general compact saturated sets. Recall that μA is the set of minimal elements of A.

Lemma 18. *Suppose A is a compact, saturated subset of a Scott domain D. Then A is minimal generated: $\uparrow \mu A = A$.*

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

For any element $x \in A$, we want to show that there exists a minimal element $y \in A$ such that $x \sqsupseteq y$.

Fix an arbitrary element $x \in A$, consider a chain C of $\downarrow x \cap A$. The greatest lower bound of C, $\sqcap C$, exists in D, since the set of all lower bounds of C is nonempty and directed. We show that $\sqcap C$ is a member of $\downarrow x \cap A$, as well.

Let O be any compact open set containing A. Clearly, $C \subseteq O$. Since O is compact, μO is a finite set, and $O = \uparrow \mu O$. This implies that there exists some $a \in \mu O$ such that $C \subseteq \uparrow a$, and hence $\sqcap C \in \uparrow a$, which implies $\sqcap C \in O$. Therefore, $\sqcap C$ is a member of every compact open set containing A.

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

The next lemma gives an "approximation property" for compact saturated sets in any Scott domain. It will be needed to deal with general extensions, because these are "limits" of compact opens, not compact opens themselves. We call it the Interpolation Theorem. We state this first, and then use it to prove a strengthened version called the Augmented Interpolation Theorem, which we apply finally to get the Dichotomy Theorem.

Lemma 19 (Interpolation theorem). *Let E be a compact saturated set of a Scott domain, and g a generator of E. For any compact open set X such that $X \supseteq E$, and generator f of X with $f \sqsubseteq g$, there exists a compact open set Y such that $X \supseteq Y \supseteq E$, f is still in Y, and f is the unique generator of Y subsuming g.*

Proof. Let $R = \{x \in \mathsf{K}(X) \mid x \not\sqsubseteq g\}$. Consider the collection C of all $\uparrow x$ for $x \in R$ together with $\uparrow f$. 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 there is a compact element $k \in D$ such that $k \sqsubseteq e$ but k does not subsume g. Since $e \in E$, then there is a $b \in \mu X$ with $b \sqsubseteq e$. Then $b \sqcup k \in \mathsf{K}(X)$ 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. $\square$

Corollary 20 (Augmented Interpolation Theorem). *Let E be a compact saturated set of a Scott domain, g be a generator of E, and d be a compact element subsuming g. For any compact open set X such that $X \supseteq E$, there exists a compact open set Y, such that*

1. *$X \supseteq Y \supseteq E$,*
2. *the generator y of Y subsuming g is unique, and*
3. *$d \sqsubseteq y$.*

Proof. Let $X \supseteq E$ be given. Choose a generator f of X with $f \sqsubseteq g$. Apply the Interpolation Theorem to X and f. Let Y' be the result. Then we can let $Y = Y'[f \leftarrow \uparrow(f \sqcup d)]$, with $y = f \sqcup d$. $\square$

This completes the lemmas which hold in general; now we consider extensions explicitly. In the next lemma, we use the "iterative" characterization (Lemma 6) of the function η, restated for the Smyth powerdomain. We use the following abbreviation: for sets U and W, put

$$\Delta(U, W) = \bigcap\{K[b \leftarrow b \cap \theta] : (K, K[b \leftarrow b \cap \theta]) \in \Delta \,\&\, K \supseteq U \,\&\, K[b \leftarrow b \cap \theta] \cap W \neq \emptyset\}.$$

Now we give the iterative definition of the function η as follows:

$$\begin{aligned}\eta(M,W,0) &= M;\\ \eta(M,W,n+1) &= \eta(M,W,n)\cap\Delta(\eta(M,W,n),W);\\ \eta(M,W) &= \bigcap_n \eta(M,W,n).\end{aligned}$$

Lemma 21. *If there is a safe element e under a system of constraints, then for any powerdomain element W, we have $e \in \eta(M,W)$.*

Proof. We show $e \in \eta(M,W,n)$ for all n. If $n=0$, then this is just (i) of Definition 15. Assume $e \in \eta(M,W,n)$. Let $(K, K[a \leftarrow a\cap\theta])$ be a default with $K \supseteq \eta(M,W,n)$ and $W \cap K[a \leftarrow a\cap\theta] \neq \emptyset$. If a does not subsume e, then e remains in the updated K, and so e must be in $\eta(M,W,n+1)$. Otherwise $a \sqsubseteq e$, so that $\uparrow a \supseteq \uparrow e$. But then $\uparrow a \cap \theta \supseteq \uparrow e \cap \theta = \uparrow e$. So e is still in the next approximation $\eta(M,W,n+1)$. □

Lemma 22. *If E is a safe extension of M with respect to Γ, then*

$$E = \uparrow\{e \in M \mid e \text{ is a safe element}\}.$$

In other words, if there exists a safe element, then the extension is just the upward-closure of all safe elements.

Proof. Given Lemma 18 and Lemma 21, all we need to show is that every generator of a safe extension is a safe element. Let g be a generator of the safe extension E. If g is not safe, then there is a semantic constraint (d,θ) such that $d \sqsubseteq g$ but $\uparrow g \cap \theta \subset \uparrow g$. We specify a compact open set K including E, where d subsumes one of its generators y, such that $E \cap K[y \leftarrow y\cap\theta]$ (i) is not empty, and (ii) is not equal to E. This violates Lemma 17, as $E = \eta(M,E)$.

We define K as follows. Apply the Augmented Interpolation Theorem using $X = M$ and the elements d and g in the previous paragraph. Let K be the obtained set Y, and $y \in K$ be the unique generator of K subsuming g. Since $d \sqsubseteq y$, and since the updated set $K[y \leftarrow y\cap\theta]$ still contains a safe element $e \in E$, the default $(K, K[y \leftarrow y\cap\theta])$ is applicable. But since $g \notin \theta$, we have our desired contradiction. □

Proof of the Dichotomy Theorem. Suppose M has a safe extension E. By Lemma 22, E is the unique extension of M, being the upward closure of the set of safe elements in M.

To show that unsafe extensions are singleton generated, let g_1 and g_2 be distinct non-safe generators of E. Since g_1 is unsafe, there is a semantic constraint (d,θ) such that $d \sqsubseteq g_1$ but $g_1 \notin \theta$. By the algebraic property of D, there is a compact element d' such that $d \sqsubseteq d' \sqsubseteq g_1$, but $g_2 \notin \uparrow d'$. Now apply the Augmented Interpolation Theorem and let K be a compact open superset of E which has a unique generator g subsuming g_2, and, moreover, $d' \sqsubseteq g$. The update of K at g with respect to (d,θ) is consistent with E, since g_2 is a common member; however, the update is no longer a superset of E because it does not contain g_1. □

We now turn to the next main result:

Theorem 23 (Extension Splitting Theorem). *Let Δ be a set of normal defaults determined as in Definition 14. Consider two nonempty compact saturated sets W_1 and W_2. Then any extension E of $W_1 \cup W_2$, using Δ as the default set, can be written as a union $E = E_1 \cup E_2$, where E_1 and E_2 are either empty or extensions of W_1 and W_2 respectively.*

Before beginning the proof, let us consider the example of three-valued logic to show how this theorem sanctions reasoning by cases. Let Σ be a set of default constraints $\tau \rightsquigarrow \theta$. Let ϕ, χ, and ψ be three-valued propositional formulas. Suppose that $\phi \mathrel{|\!\!\sim}_\Sigma \psi$. This means that every extension of $Mod(\phi)$ is contained in $Mod(\psi)$, where $Mod(\phi)$ is just the set of all partial truth assignments satisfying ϕ. Similarly suppose $\chi \mathrel{|\!\!\sim}_\Sigma \psi$.

We want to show that $\phi \vee \chi \mathrel{|\!\!\sim}_\Sigma \psi$; by definition, this means that every extension of $Mod(\phi \vee \chi)$ is a subset of $Mod(\psi)$. Since

$$Mod(\phi \vee \chi) = Mod(\phi) \cup Mod(\chi),$$

the result follows immediately from the Splitting Theorem.

Proof of the Splitting Theorem. Apply the Dichotomy Theorem. Suppose first that E is a safe extension of $W_1 \cup W_2$. All safe elements of $W_1 \cup W_2$ are in E, and a safe element of E is either a safe element of W_1 or a safe element of W_2. So, if $W_1 \cap \mu E$ is non-empty then it is the set of all minimal safe elements of W_1. This means that it must be the set of generators of the unique safe extension E_1 of W_1. Similar remarks apply to W_2, giving us the set E_2.

Now suppose that there are no safe elements in $W_1 \cup W_2$. Then by Dichotomy, $E = \uparrow g$ for some g in $W_1 \cup W_2$. Without loss of generality, $g \in W_1$. Then we have

$$(W_1 \cup W_2) \supseteq W_1 \supseteq \uparrow g,$$

so that by Theorem 10, item(5), $\uparrow g$ is an extension of W_1. If $g \in W_2$ as well, then similar remarks show that $\uparrow g$ is an extension of W_2. □

4 Observations and consequences

4.1 A constructive proof of the Splitting Theorem

The above proof of the Splitting Theorem is nonconstructive (note the use of Zorn's Lemma in Lemma 18). Our original proof was much more constructive, but gave no hint of the general picture of Smyth power extensions. Here is a quick overview of that proof, which does make use of new properties of the extension operator η in any Scott domain.

The first lemma is specific to the Smyth powerdomain:

Lemma 24. *Let E be an extension of $W_1 \cup W_2$. Then*

$$\eta(W_1 \cup W_2, E) = \eta(W_1, E) \cup \eta(W_2, E).$$

This lemma can be proved by induction. But the proof is not easy, and depends on establishing essentially the structural lemma needed for the induction step by assuming that the sets W_1 and W_2 are compact open, that is, finite elements of the Smyth powerdomain. We have to overcome the fact, though, that the lemma is false unless E is an extension of $W_1 \cup W_2$. This is handled using a special combinatorial argument. We then argue using continuity considerations that the needed structural lemma holds for general compact saturated sets. In the midst of all of this we have to change the set of defaults.

Once we have attained Lemma 24, we still need to show how it entails the Splitting Theorem. This comes from a new and general fact about extensions in any Scott domain. We therefore include the proof.

Theorem 25. *Let Δ be a default set in a Scott domain, and suppose $x \sqsubseteq y$, and that e is an extension of x. If $\eta(y, e)$ is not $\top$, then it is an extension of y.*

Proof. Suppose $x \sqsubseteq y$. By Lemma 7, we have $\eta(x, e) \sqsubseteq \eta(y, e)$. But $\eta(x, e) = e$, so $e \sqsubseteq \eta(y, e)$. By Lemma 8, $\eta(y, \eta(y, e)) \sqsubseteq \eta(y, e)$. The lemma will now follow if we can show that $\eta(y, e) \sqsubseteq \eta(y, \eta(y, e))$. We do this by induction, showing that for each i

$$\eta(y, e, i) \sqsubseteq \eta(y, \eta(y, e), i).$$

The basis is clear. Assume the result for i. By definition

$$\eta(y, e, i + 1) = \eta(y, e, i) \sqcup \bigsqcup\{b \mid (a, b) \in \Delta \wedge a \sqsubseteq \eta(y, e, i) \wedge b \uparrow e\}.$$

Now notice that any b's used on the right actually subsume $\eta(y, e)$, because $\eta(y, e)$ is the least upper bound of all the $\eta(y, e, i)$ as i goes to infinity. Therefore the expression on the right subsumes

$$\eta(y, \eta(y, e), i) \sqcup \bigsqcup\{b \mid (a, b) \in \Delta \wedge a \sqsubseteq \eta(y, \eta(y, e), i) \wedge b \uparrow \eta(y, e)\}$$

by inductive hypothesis and our observation on the b's. But the last line is just $\eta(y, \eta(y, e), i + 1)$. $\square$

The Splitting Theorem follows if we take $e = E$, $x = W_1 \cup W_2$, and successively $y = W_1$, then $y = W_2$ in Theorem 25 (recall that the ordering in the Smyth powerdomain is reverse inclusion.)

4.2 Complexity properties

Using the Dichotomy Theorem, it is possible to understand why one might obtain the co-NP(3)-completeness result mentioned in the introductory remarks. Suppose we are given a set of syntactic defaults Σ, and two formulas ϕ and ψ. We wish to decide if $\phi \mid\!\sim_{\Sigma} \psi$. In ordinary default logic, one has problems with doing repeated consistency checks to test the applicability of the default rules. But in power default logic, the only really necessary extra consistency check is for the existence of a safe element. If one finds such an element, then another NP test can be used (because of the unique safe extension) to determine non-monotonic entailment. In the non-safe case, though, one can prove that the singleton generator g of a non-safe extension must be a total truth assignment (this is actually not straightforward.) For the non-safe case, therefore, non-monotonic entailment reduces to *monotonic* entailment $\phi \vdash \psi$ – the defaults disappear entirely from the picture! The upshot is that only three NP "oracle" calls are ever used to decide $\mid\!\sim$. See [14] for details of this result.

5 Concluding remarks

We have not yet indicated how to get a spectrum of logics based on default domains. We first remark that because the Scott topology gives us a notion of open set, we can think of these sets as defining predicates over domains. Smyth himself notes in his powerdomain paper [15] that we get in this way an abstract logic. The problem is to make such a logic concrete, and here there are several options. We can take propositional logic, and derive semantic constraints from syntactic constraints as indicated in the introductory sections. This is also possible for first-order logic; the full version of the paper supplies details for this. Briefly, the way to generate semantic defaults from syntax is to use *open normal defaults* as in Reiter's work. We have explicated the idea in [12], though that paper does not treat disjunctive or power defaults. Finally, it is also possible to do the same for *feature logic* [13]. For each of the three logics mentioned, we use a Scott domain of partial models, and pass to its Smyth powerdomain.

In the propositional case, we have mentioned complexity results; work is underway on an implementation of non-monotonic reasoning using satisfiability testing rather than non-monotonic theorem proving. We will report on this project in a future paper.

Finally, the Smyth powerdomain seems like a promising tool for further investigation of the semantics of disjunctive logic programming. One would need to consider unification issues in partial first-order logic, because one needs first-order terms to keep track of structured data.

References

1. H. Aït-Kaci. *A Lattice-Theoretic Approach to Computation based on a Calculus of Partially Ordered Type Structures.* PhD thesis, University of Pennsylvania, 1984.
2. K. R. Apt and R. N. Bol. Logic programming and negation: a survey. Technical Report CS-R9402, CWI, Amsterdam, 1994.
3. P. Buneman. Functional programming and databases. In D. Turner, editor, *Research Topics in Functional Programming*, pages 155–169. Addison-Wesley, 1990.
4. M. Gelfond, V. Lifschitz, H. Przymuzińska, and M. Truszczyński. Disjunctive defaults. In *Proceedings of Second Annual Conference on Knowledge Representation*, pages 230–237. Morgan Kaufmann, 1991.
5. G. Gottlob. Complexity results for non-monotonic logics. In *Notes on Fourth Int. Workshop on Non-monotonic Reasoning*, 1992.
6. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models, and cumulative logics. *Artificial Intelligence*, 44:167–207, 1990.

7. L. Libkin. *Aspects of Partial Information in Databases.* PhD thesis, University of Pennsylvania, 1994.
8. W. Marek, A. Nerode, and J. Remmel. A theory of nonmonotonic rule systems. In *Proceedings of 5th IEEE Symposium on Logic in Computer Science*, pages 79–94, 1990.
9. F. Pereira and S. Shieber. The semantics of grammar formalisms seen as computer languages. In *Proceedings of 10th International Conference on Computational Linguistics: COLING 84*, 1984.
10. C. Pollard and M. A. Moshier. Unifying partial descriptions of sets. In P. Hansen, editor, *Vancouver Studies in Cognitive Science: vol. I.* University of British Columbia Press, 1990.
11. Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.
12. W. Rounds and G. Q. Zhang. Logical considerations on default semantics. To appear,*Annals of Mathematics and Artificial Intelligence*, 1997.
13. W. Rounds and G. Q. Zhang. Suggestions for a non-monotonic feature logic. Technical Report CS-R9551, CWI, Amsterdam, 1995.
14. W. Rounds and G. Q. Zhang. Complexity of power default reasoning. In *Proceedings of LICS 97*, July 1997.
15. M. B. Smyth. Powerdomains. *Journal of Computer and System Science*, 16(1):23–36, February 1978.