Optimality Theory as a Family of Cumulative Logics

Ph. Besnard, G. Fanselow, T. Schaub[0]
Universität Potsdam, D–14439 Potsdam, Germany

January 31, 2003

Abstract. We investigate two formalizations of Optimality Theory, a successful paradigm in linguistics. We first give an order-theoretic counterpart for the data and process involved in candidate evaluation. Basically, we represent each constraint as a function that assigns every candidate a degree of violation. As for the second formalization, we define (after Samek-Lodovici and Prince) constraints as operations that select the best candidates out of a set of candidates. We prove that these two formalizations are equivalent (accordingly, there is no loss of generality with using violation marks and dispensing with them is only apparent).

Importantly, we show that the second formalization is equivalent with a class of operations over sets of formulas in a given logical language. As a result, we prove that Optimality Theory can be characterized by certain cumulative logics. So, applying Optimality Theory is shown to be reasoning by the rules of cumulative logics.

Keywords: Cumulative logics, non-monotonic logics, Optimality Theory.

1. Introduction

Optimality Theory is a grammatical architecture that was invented in phonology [Prince & Smolensky 1993] but managed to spread into the other subdisciplines of linguistics quite successfully. In its standard version, Optimality Theory (cf [Kager 1999] for instance) is a representational rather than a derivational account of grammatical facts: It comprises of a set of grammatical constraints that evaluate the quality of candidate structures (i.e., representations), but it does not care how these candidate structures are generated. (To be to the point, candidate evaluation is representational but a derivational theory could underly the generation of candidates.)

In this respect, Optimality Theory only needs a component that decides which structures are compared with each other. The grammatical description of Optimality Theory is thus anchored with an input component. Inputs could be strings of sounds (in phonology), sets of morphemes (in morphology) or predicate-argument structures (in syntax). They are subjected to a GEN component that generates the candidate set on the basis of the input by very general grammat-

[0] Corresponding author: Torsten Schaub, Institut für Informatik, Universität Potsdam, Postfach 900327, D–14439 Potsdam, Germany, torsten@cs.uni-potsdam.de (Phone: +49 331 977 3080, Fax: +49 331 977 3122)

ical processes. The candidate set is passed on to the EVAL component (EVAL stands for evaluation) that is in charge of selecting the optimal candidate according to the language at hand, using the grammatical constraints.

Optimality Theory assumes that the grammatical constraints are universal (all languages work with the same set of constraints): "syllables have an onset", "sentences have a subject" are examples of what could be a constraint in Optimality Theory.

A constraint is either categorical or graded, where categorical constraints are the ones that do not support multiple violations from a single candidate. E.g., "a declarative sentence has more than one word" would be categorical: A sentence could violate this *once*, if at all. In contrast, a graded constraint can be violated multiply by a candidate. In the sequel, we consider graded constraints (categorical constraints can clearly be viewed as a special case).

The grammatical constraints may imply incompatible requirements for certain structures. E.g., objects should follow the verb (compare *John loves Mary* with **John Mary loves*) but questions should begin with the question word (*how did she say this* vs. **she said this how*). For an object question, the two principles make different predictions (*what did she say* vs. **she said what*), and we see that the conflict between the two principles is resolved in favor of the question principle (as far as English is concerned, but not necessarily so with other languages).

Optimality Theory claims that the grammatical constraints are organized in a hierarchy. Whenever two options compete, the one with the better violation profile wins: A candidate structure S is grammatical if and only if there is no competitor S' such that the highest constraint on which S and S' differ incurs less violations from S' than from S.

Conflict resolution is thus lexicographic: The numbers of violations of a candidate with respect to each constraint form a vector (constraints being considered in decreasing order).

Here is an example. The highest constraint is "the question word occurs first" (categorical), the next highest constraint is "the verb group comes second" (categorical), and the lowest constraint is "any non-subject item occurs after the subject" (graded). Consider the candidates: (1) **where she is now?*, (2) **she is where now?*, (3) *where is she now?*, (4) **where is now she?* The first constraint rules out (2) (the only candidate to violate it), and then similarly for (1) with respect to the second constraint. The last constraint is violated thrice by (4) but only twice by (3) that is thus the best candidate (the fact that (2) does not violate the last constraint is irrelevant: (2) was already out).

In Optimality Theory, this is usually visualized in a two-dimensional table as follows. Rewriting the four candidates (1) to (4) as x_1 to x_4

and abbreviating the three constraints (from higher to lower) by c_1 to c_3, we obtain the configuration[1] depicted in Table I:

Table I. Constraint tableau in Optimality Theory (optimal candidate: x_3).

	c_1	c_2	c_3
x_1: **where she is now?*		$*$	
x_2: **she is where now?*	$*$		
x_3: *where is she now?*			$**$
x_4: **where is now she?*			$***$

The ranking among the contraints is reflected by their decreasing importance from left to right. With the exception of grey cells, Table I displays to what extent each candidate (dis)agrees with each constraint. E.g., the violation of c_1 by x_2 is indicated by $*$ while the triple violation of c_3 by x_4 is represented by $***$. Grey cells denote data that are not taken into account (for instance, the cell $x_2 \times c_3$ is grey to reflect the aforementioned fact that no matter how well (2) fares with respect to the last constraint it is irrelevant because (2) is out by virtue of the first constraint).

Summing up, constraints in Optimality Theory turn out to be rules with exceptions: They are universal but they are *not* universally valid (for instance, there are syllables in English that have no onset). Accordingly, Optimality Theory provides a methodology to apply rules with exceptions. However, an early attempt of defining a logic for Optimality Theory [Hammond 2000] amounts to specifying a first-order theory in classical logic although reasoning from rules with exceptions is known to fall under the umbrella of the so-called non-monotonic logics [Makinson 1994]. The present paper shows that the logic of Optimality Theory is no individual logic but a class of cumulative logics (which are non-monotonic logics).

In Section 2, we provide an order-theoretic formalization for the usual process of candidate evaluation. In effect, we introduce a formal representation for candidates and constraints where the relative merit of two candidates can be assessed (depending on the ranking of the constraints) from the amount of violations they cause to the constraints. In Section 3, we show that this formalization is equivalent to a more abstract one where each constraint is represented as a special operation that selects the best candidates from any set of candidates. The main

[1] Grey cells are cells whose contents are *not* given (because they are not to be used, an explanation for this follows Table I; in particular, a grey cell may have the candidate violating the constraint twice for instance).

purpose of this other formalization is to facilitate the transition to logics as we show in Section 4 that candidate evaluation in Optimality Theory amounts to a class of cumulative logics.

2. Basic features

Outline. *In this section, we give an order-theoretic counterpart for the data and process involved in candidate evaluation such as exemplified by Table I. In effect, we represent each constraint as a function that assigns every candidate a degree of violation.*

Let $\{v_c \mid c \in \mathcal{C}\}$ be a family of total functions from $\mathcal{U}$ to M such that the index set $\mathcal{C}$ is finite whereas M is any[2] well-ordered set.

Using the terminology from Optimality Theory, $\mathcal{C}$ consists of constraints and $\mathcal{U}$ is the universe of all candidates (a non-empty set which is possibly infinite). Intuitively, $v_c(x)$ gives the degree of violation of the constraint c by the candidate x (for all x and c).

As M (the co-domain of v_c) comes with a well-ordering $\leq$, candidates can be compared with respect to the violations they make a constraint to incur: $v_c(x) \leq v_c(y)$ means that y violates c at least as much as x does.

Of course, $a < b$ is defined as $a \leq b$ and $b \not\leq a$ for all a and b in M.

As an illustration, our above example has $v_{c_3}(x_3) \leq v_{c_3}(x_4)$ while $v_{c_3}(x_4) \not\leq v_{c_3}(x_3)$, hence $v_{c_3}(x_3) < v_{c_3}(x_4)$. Also, both $v_{c_2}(x_3) \leq v_{c_2}(x_4)$ and $v_{c_2}(x_4) \leq v_{c_2}(x_3)$ hold, yielding $v_{c_2}(x_3) = v_{c_2}(x_4)$.

A crucial ingredient in Optimality Theory is the ranking of constraints, which can be any linear ordering $\ll$ over the set of all constraints. As for notation, $c \ll c'$ means that violating c' is more serious than violating c.

Table II. Lexicographic ordering over violation profiles from greatest to least (best): p_2, p_1, p_4, p_3.

$p_2 :$	$\langle\{*\},$	$\{\},$	$\{\}\rangle$	(violation profile for x_2)
$p_1 :$	$\langle\ \{\},$	$\{*\},$	$\{*\}\rangle$	(violation profile for x_1)
$p_4 :$	$\langle\ \{\},$	$\{\},$	$\{***\}\rangle$	(violation profile for x_4)
$p_3 :$	$\langle\ \{\},$	$\{\},$	$\{**\}\rangle$	(violation profile for x_3)

[2] The set denoted by M (for *violation marks*) can be thought of as the set of natural numbers possibly extended with some infinite number(s) if one is careful enough not to use it to introduce explicit counting which Optimality Theory outlaws. Usually, one considers a countable M with a greatest element standing for ∞ but such a restriction need not be adopted in an abstract formalization as intended here.

The ranking of constraints $c_n \ll c_{n-1} \ll \ldots \ll c_2 \ll c_1$ induces a lexicographic order over the set of violation profiles of the candidates (where $\langle v_{c_1}(x), \ldots, v_{c_n}(x)\rangle$ is taken to be the violation profile of x, cf Table II). There are two ways to go.

– The lexicographic order induced by $\ll$ yields a *partial* order over the candidates:

$$x \prec y \text{ iff there exists } c \in \mathcal{C} \text{ s.t. } v_c(x) < v_c(y) \text{ and for all } c' \in \mathcal{C}, \text{ if } v_{c'}(y) < v_{c'}(x) \text{ then } c' \ll c$$

where $x \prec y$ expresses that the candidate y violates the constraints (as ranked) more severely than x does

Clearly, it then need not be the case that any two candidates can be ranked relative to each other (in our running example, this would happen for x_3 and x_4, if c_3 were omitted). If that is desired, another direction is possible as follows.

– The lexicographic preorder induced by $\ll$ yields a total *preorder* over the candidates:

$$x \preceq y \text{ iff there is no } c \in \mathcal{C} \text{ s.t. } v_c(y) < v_c(x) \text{ where for all } c' \in \mathcal{C}, \text{ if } c \ll c' \text{ then } v_{c'}(x) = v_{c'}(y)$$

where $x \preceq y$ expresses that the candidate y violates the constraints (as ranked) at least as much as x does

Notice that each of $\prec$ and $\preceq$ depends on $\mathcal{C}$ and $\ll$ so that the correct notation would rather be $\prec_{\mathcal{C}}$ and $\preceq_{\mathcal{C}}$ (where $\ll$ is implicit) but the subscript is omitted here because no confusion arises: $\mathcal{C}$ is fixed (and so is $\ll$).

Expectedly enough, $\prec$ as defined above turns out to be the strict ordering obtained from $\preceq$ in the usual way:

THEOREM 2.1. *$x \prec y$ iff $x \preceq y$ and $y \not\preceq x$*

COROLLARY 2.1. *$x \prec y$ iff $x \preceq y$ and $v_c(x) \neq v_c(y)$ for some $c \in \mathcal{C}$*

THEOREM 2.2. *$x \prec y$ iff there exists $c \in \mathcal{C}$ which satisfies both conditions below:*

- *$v_c(x) < v_c(y)$*
- *$v_{c'}(x) \leq v_{c'}(y)$ for all $c' \in \mathcal{C}$ s.t. $c \ll c'$*

The fundamental definition can now be given: A candidate is *optimal* iff it is a minimal element for $\preceq$ in $\mathcal{U}$.

THEOREM 2.3. *A candidate is optimal iff it is a minimal element for $\prec$ in $\mathcal{U}$.*

Underlying Theorem 2.2 is the above definition of $\prec$ which embodies the principle of *constraint demotion* [Tesar & Smolensky 1998] such that constraints satisfied by optimal candidates dominate (in the constraint hierarchy that $\ll$ governs) constraints violated by optimal candidates.

In the limiting case where $v_c(x) = v_c(y)$ for all x, y and c, all candidates are optimal (a degenerate situation). This happens exactly when all constraints are vacuous (c is *vacuous*[3] iff $v_c(x) = v_c(y)$ for all x and y).

There always exists at least one optimal candidate but there need *not* be a unique optimal candidate!

THEOREM 2.4. *A candidate $x \in \mathcal{U}$ is the unique optimal candidate wrt $\mathcal{C}$ ordered by $\ll$ iff for every other candidate $y \in \mathcal{U}$, there exists $c \in \mathcal{C}$ such that $v_c(x) < v_c(y)$ and $v_{c'}(x) = v_{c'}(y)$ for all $c' \in \mathcal{C}$ where $c \ll c'$.*

Generally, considering only some of the candidates and only some of the constraints (subject to the given ordering) is enough to discriminate candidates that may qualify as optimal: *current relative winners*. They can be defined, for all $\mathcal{K} \subseteq \mathcal{U}$ and for all $\mathcal{C}' \subseteq \mathcal{C}$, as follows:

$$W_{\ll}(\mathcal{K}, \mathcal{C}') = \begin{cases} \mathcal{K} & \text{if } \mathcal{C}' = \emptyset \\ W_{\ll}(\{x \in \mathcal{K} \mid \forall y \in \mathcal{K}, v_c(x) \leq v_c(y)\}, \mathcal{C}' \setminus \{c\}) & \text{otherwise} \end{cases}$$

where "otherwise" assumes $c \in \mathcal{C}'$ such that $c' \ll c$ for all $c' \in \mathcal{C}' \setminus \{c\}$

For an illustration, consider our running example.

$$\begin{aligned} W_{\ll}(\{x_1, x_2, x_3, x_4\}, \{c_1, c_2, c_3\}) &= W_{\ll}(\{x_1, x_3, x_4\}, \{c_2, c_3\}) \\ &= W_{\ll}(\{x_3, x_4\}, \{c_3\}) \\ &= W_{\ll}(\{x_3\}, \emptyset) \\ &= \{x_3\} \end{aligned}$$

THEOREM 2.5. *A candidate $x \in \mathcal{U}$ is an optimal candidate wrt $\mathcal{C}$ ordered by $\ll$ iff $x \in W_{\ll}(\mathcal{U}, \mathcal{C})$.*

This mimics[4] the most concrete manifestation of Optimality Theory at work. Yet, even abstract approaches can prove to shed light on some aspects of the theory as illustrated by the next two sections.

[3] This is a relative notion, depending on the set of candidates (a constraint c vacuous wrt $\mathcal{U}$ need not be vacuous wrt $\mathcal{U}' \neq \mathcal{U}$)

[4] In this respect, see also the formal construction by [Frank & Satta 1998].

3. Connecting with the formalization of Samek-Lodovici and Prince

Outline. *In this section, we no longer represent constraints as functions that assign each candidate a degree of violation. Instead, we give an equivalent formalization where constraints are operations that select the best candidates out of a set of candidates. That is, a constraint is an operation C that takes a set of candidates $\mathcal{X}$ and yields the winners $C(\mathcal{X})$ among these candidates:*

$$C : \mathcal{X} \subseteq \mathcal{U} \mapsto C(\mathcal{X}) \subseteq \mathcal{U}$$

$$\begin{array}{rl} I. & C(\mathcal{X}) \subseteq \mathcal{X} \\ II. & \forall \mathcal{X} \neq \emptyset,\ C(\mathcal{X}) \neq \emptyset \\ III. & \textit{if } C(\mathcal{Y} \cup \mathcal{Z}) \neq C(\mathcal{Y}) \textit{ and } C(\mathcal{Y} \cup \mathcal{Z}) \neq C(\mathcal{Z}) \\ & \qquad \textit{then } C(\mathcal{Y} \cup \mathcal{Z}) = C(\mathcal{Y}) \cup C(\mathcal{Z}) \end{array}$$

Principles I. and II. are choice and forced choice [Prince 2001][5]. Principle III. is simply divide and conquer (Section 3.2).

[Samek-Lodovici & Prince 1999] introduces a very insightful formalization of constraints as functions over the powerset of the universe of candidates. However, they depart from the more natural formulation of constraints and we therefore provide the missing link in Section 3.1.

3.1. Valuations over candidates vs unary operations over the powerset of candidates

Samek-Lodovici and Prince consider constraints as functions $C : \mathcal{P}(\mathcal{U}) \to \mathcal{P}(\mathcal{U})$ which can be captured here in the obvious way as:

$$C(\mathcal{K}) \stackrel{\text{def}}{=} \{x \in \mathcal{K} \mid v_c(x) \leq v_c(y) \text{ for all } y \in \mathcal{K}\}^6$$

It is shown in Section 3.2 that the move in the converse direction (from C to v_c) is possible, too.

[5] While this work was under submission, Alan Prince independently proposed an analogous formalization with III. replaced by what he calls contextual independence of choice:

$$\text{if } \mathcal{Y} \cap C(\mathcal{X}) \neq \emptyset \text{ then } C(\mathcal{Y} \cap \mathcal{X}) = \mathcal{Y} \cap C(\mathcal{X})$$

[6] Due to typographic ambiguity, it is worth giving the details: Consider a constraint in Optimality Theory. Formalizing it in the framework developed so far means considering some $c \in \mathcal{C}$ (and v_c is a total function from $\mathcal{U}$ to M). Then, the operation defined as $\{x \in \mathcal{K} \mid \forall y \in \mathcal{K}, v_c(x) \leq v_c(y)\}$ for all $\mathcal{K} \subseteq \mathcal{U}$ is an equivalent way of formalizing the same constraint (the new item is written with capital C to indicate that it is the same constraint as the original $c \in \mathcal{C}$).

Following Samek-Lodovici and Prince, the ranking of constraints $c_n \ll c_{n-1} \ll \ldots \ll c_2 \ll c_1$ is elegantly rendered by mere composition of the corresponding functions $C_n \circ C_{n-1} \circ \cdots \circ C_2 \circ C_1$ as follows: For all $\mathcal{K} \subseteq \mathcal{U}$, the best candidates in $\mathcal{K}$ according to $c_n \ll c_{n-1} \ll \ldots \ll c_2 \ll c_1$ are given as $C_n(C_{n-1}(\ldots C_1(\mathcal{K})\ldots))$.

We now show that the formalization of a constraint as done by Samek-Lodovici and Prince can easily be obtained as a byproduct of our own formalization:

THEOREM 3.1. *For all* $\mathcal{K} \subseteq \mathcal{U}$*,*

$$W_{\ll}(\mathcal{K}, c_n \ll c_{n-1} \ll \cdots \ll c_2 \ll c_1) = C_n(C_{n-1}(\ldots C_2(C_1(\mathcal{K}))\ldots))$$

When reformulated as functions over the powerset of candidates, constraints exhibit distinctive traits:

THEOREM 3.2. *Let* $c \in \mathcal{C}$*. For all* $\mathcal{X}$ *and* $\mathcal{Y}$ *in* $\mathcal{P}(\mathcal{U})$*, the following properties hold:*

- (i) $C(\mathcal{X}) \subseteq \mathcal{X}$
- (ii) *if* $\mathcal{X} \subseteq \mathcal{Y}$ *then* $\mathcal{X} \setminus C(\mathcal{X}) \subseteq \mathcal{Y} \setminus C(\mathcal{Y})$
- (iii) *if* $\mathcal{X} \cap C(\mathcal{X} \cup \mathcal{Y}) \neq \emptyset$ *then* $C(\mathcal{X}) \subseteq C(\mathcal{X} \cup \mathcal{Y})$

These are actually the most salient features of constraints, as it will appear in Section 3.2: Theorem 3.7 and Theorem 3.8 show that (ii) and (iii) are equivalent with the property which is meant to characterize constraints defined as unary operations over the powerset of candidates.

3.2. Constraints as unary operations over the powerset of candidates

Not all $C : \mathcal{P}(\mathcal{U}) \to \mathcal{P}(\mathcal{U})$ are constraints (even when $C(\mathcal{K}) \subseteq \mathcal{K}$ is implicitly assumed for all $\mathcal{K} \in \mathcal{P}(\mathcal{U})$) but Samek-Lodovici and Prince simply claim that "the ordering imposed by a constraint is any form of partial order on candidates in which every subset in a candidate set has a maximal element" (consequently, such an ordering induces strata within the set of candidates as observed by Samek-Lodovici and Prince). Basically, the ordering must be reconstructed in order to check whether the function at hand qualifies as a constraint. In their account, the formal definition of a constraint is accordingly clumsy because the function C is defined using the order $C\,\hat{}$ whereas the function should be primitive and the ordering should merely result from it.

A less sloppy approach is to consider unary operations C over the non-empty[7] subsets of $\mathcal{U}$ (but still requiring $C(\mathcal{K}) \subseteq \mathcal{K}$), imposing for all $\mathcal{K}$ and $\mathcal{K}'$ in $\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ the next condition:

(1) $$\text{if } C(\mathcal{K} \cup \mathcal{K}') \neq C(\mathcal{K}) \text{ and } C(\mathcal{K} \cup \mathcal{K}') \neq C(\mathcal{K}') \quad \text{then } C(\mathcal{K} \cup \mathcal{K}') = C(\mathcal{K}) \cup C(\mathcal{K}')$$

Condition (1) can be given an informal interpretation, arising from the divide-and-conquer process of applying Optimality Theory: Split the set of candidates into two parts. Possibly, the winners of one part take it all (i.e., they are the winners of the whole). Otherwise, it means that every winner of one part ties with every winner of the other part and therefore they all are the winners of the whole.

It can be shown that there is no loss of generality by introducing M (hence defining constraints as valuations) as done in Section 2. Here is why restriction to a pure order-based formulation referring to strata is illusory:

Define

$$v_c(z) \stackrel{\text{def}}{=} \bigcup_{\substack{\mathcal{K} \subseteq \mathcal{U} \\ z \in C(\mathcal{K})}} \mathcal{K}$$

That is, $v_c(x)$ is the set of all candidates that x is at least as good as.[8][9]

Clearly, all these comprise a set of violations marks M:

$$M \stackrel{\text{def}}{=} \left\{ \bigcup_{\substack{\mathcal{K} \subseteq \mathcal{U} \\ z \in C(\mathcal{K})}} \mathcal{K} \mid c \in \mathcal{C}, z \in \mathcal{U} \right\}$$

or, in short, $M = \{v_c(z) \mid c \in \mathcal{C}, z \in \mathcal{U}\}$.

[7] Generalization to the empty subset is trivial, hence extension to $C : \mathcal{P}(\mathcal{U}) \to \mathcal{P}(\mathcal{U})$ is taken for granted in the sequel.

[8] An equivalent formulation is

$$v_c(x) \stackrel{\text{def}}{=} \{y \in \mathcal{U} \mid x \in C(\mathcal{K} \cup \{y\}) \text{ for some } \mathcal{K} \subseteq \mathcal{U}\}$$

[9] Actually, the simplest formulation is

$$v_c(x) \stackrel{\text{def}}{=} \{y \in \mathcal{U} \mid x \in C(\{x, y\})\}$$

Moreover, condition (1) is essential in passing from C to v_c because it ensures coherence[10] (precluding the case $C(\{x, y, z\}) = \{x, y\}$ when $C(\{x, y\}) = \{x\}$ for instance). In particular, two distinct operations C and C' are of course meant to represent two distinct constraints but this is only guaranteed if both operations enjoy condition (1) in which case C and C' define two distinct valuations (in symbols, $v_c \neq v_{c'}$).

Also, define

$$(*) \qquad v_c(x) \leq v_c(y) \text{ iff } v_c(y) \subseteq v_c(x)$$

The ordering just defined by $(*)$ is linear over the image of all subsets of $\mathcal{U}$ by v_c as condition (1) guarantees:

THEOREM 3.3. *Let $C : \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\} \to \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ be such that $C(\mathcal{K}) \subseteq \mathcal{K}$ for all $\mathcal{K} \in \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$. If C satisfies condition (1) then either $v_c(x) \leq v_c(y)$ or $v_c(y) \leq v_c(x)$ for all x and y in $\mathcal{U}$.*

Furthermore, condition (1) ensures that $\leq$ is a well-ordering for the set of violation marks induced from a single constraint:

THEOREM 3.4. *Let $C_i : \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\} \to \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ be a finite family such that $C_i(\mathcal{K}) \subseteq \mathcal{K}$ for all $\mathcal{K} \in \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$. If each C_i satisfies condition (1) then $\leq$ defined by $(*)$ is well-founded on $M = \bigcup_i \{v_{c_i}(z) \mid z \in \mathcal{U}\}$ and is a well-ordering for each $M_i = \{v_{c_i}(z) \mid z \in \mathcal{U}\}$.*

Samek-Lodovici and Prince notice that the class of functions from $\mathcal{P}(\mathcal{U})$ to $\mathcal{P}(\mathcal{U})$ is evidently preserved by composition but they cannot state that constraints form a subclass which is also preserved by composition. By means of condition (1), we can take care of this:

THEOREM 3.5. *Let all $f : \mathcal{P}(\mathcal{U}) \to \mathcal{P}(\mathcal{U})$ satisfying condition (1) and such that $f(\mathcal{K}) \subseteq \mathcal{K}$ for all $\mathcal{K} \subseteq \mathcal{U}$ be called an abstract constraint. If C and C' are abstract constraints then $C \circ C'$ is also an abstract constraint.*

We can also verify that repeating a constraint is harmless:

THEOREM 3.6. *Let $C : \mathcal{P}(\mathcal{U}) \to \mathcal{P}(\mathcal{U})$ be such that $C(\mathcal{K}) \subseteq \mathcal{K}$ for all $\mathcal{K} \subseteq \mathcal{U}$. If C satisfies condition (1) then $C(C(\mathcal{K})) = C(\mathcal{K})$ for all $\mathcal{K} \subseteq \mathcal{U}$.*

COROLLARY 3.1. *Let $C_i : \mathcal{P}(\mathcal{U}) \to \mathcal{P}(\mathcal{U})$ be a family (for all i in some I) such that $C_i(\mathcal{K}) \subseteq \mathcal{K}$ for all $\mathcal{K} \subseteq \mathcal{U}$. If each C_i satisfies condition (1) then $C_j(C_l(\ldots C_j(\mathcal{K}) \ldots)) = C_l(\ldots C_j(\mathcal{K}) \ldots)$ for all $\mathcal{K} \subseteq \mathcal{U}$ and j, l in I.*

[10] Samek-Lodovici and Prince have coherence implied by the statement cited above, another reason why their account is somehow shaky.

Although they do not exhibit such obvious significance, there are other principles of interest as given in the next theorem.

THEOREM 3.7. *Let $C : \mathcal{P}(\mathcal{U}) \to \mathcal{P}(\mathcal{U})$ be such that $C(\mathcal{K}) \subseteq \mathcal{K}$ for all $\mathcal{K} \subseteq \mathcal{U}$. If C satisfies condition (1) then it also satisfies the principles below:*

(2) $C(\mathcal{X} \cup \mathcal{Y}) \subseteq C(\mathcal{X}) \cup C(\mathcal{Y})$

(3) $C(\mathcal{X}) \subseteq C(\mathcal{X} \cup \mathcal{Y})$ *or* $C(\mathcal{Y}) \subseteq C(\mathcal{X} \cup \mathcal{Y})$

(4) *if* $\mathcal{X} \subseteq C(\mathcal{Y})$ *then* $\mathcal{X} = C(\mathcal{X})$

(5) *if* $\mathcal{X} \subseteq \mathcal{Y}$ *then* $\mathcal{X} \setminus C(\mathcal{X}) \subseteq \mathcal{Y} \setminus C(\mathcal{Y})$

(6) *if* $\mathcal{X} \cap C(\mathcal{X} \cup \mathcal{Y}) \neq \emptyset$ *then* $C(\mathcal{X}) \subseteq C(\mathcal{X} \cup \mathcal{Y})$

(7) *if* $\mathcal{X} \subseteq \mathcal{Y} \setminus C(\mathcal{Y})$ *then* $C(\mathcal{Y} \setminus \mathcal{X}) = C(\mathcal{Y})$ *unless* $C(\mathcal{X}) = C(\mathcal{Y}) = \emptyset$

None of (2)–(7) is equivalent with (1).

Intuitively, (2) means that the winners for a set of candidates must be winners for some (but not any!) subset of these candidates. Conversely, (3) indicates that the winners for one of any two complementary subsets of the candidates are winners for all the candidates (but they need not be the only winners for all the candidates). As for (4), it expresses that any collection of winners for some set of candidates is its own set of winners. Next, (5) is the obvious fact that losers remain losers no matter what additional candidates may enter the picture. Roughly, (6) states that the members in a group of winners for a set of candidates all keep or lose the status of a winner together. Lastly, (7) is another obvious fact to the effect that disregarding any bunch of losers leaves the set of winners unchanged.

Now, the question arises: Are there constraints beyond the scope of condition (1)? In view of Theorem 3.2, it can be seen that condition (1) actually captures all possible constraints when expressed in the form of functions from sets of candidates to sets of candidates as shown now:

THEOREM 3.8. *Let $C : \mathcal{P}(\mathcal{U})\setminus\{\emptyset\} \to \mathcal{P}(\mathcal{U})\setminus\{\emptyset\}$. If all three properties*

(*i*) $C(\mathcal{X}) \subseteq \mathcal{X}$

(*ii*) *if* $\mathcal{X} \subseteq \mathcal{Y}$ *then* $\mathcal{X} \setminus C(\mathcal{X}) \subseteq \mathcal{Y} \setminus C(\mathcal{Y})$

(*iii*) *if* $\mathcal{X} \cap C(\mathcal{X} \cup \mathcal{Y}) \neq \emptyset$ *then* $C(\mathcal{X}) \subseteq C(\mathcal{X} \cup \mathcal{Y})$

are satisfied for every $\mathcal{X}$ and $\mathcal{Y}$ in $\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ then condition (1) holds.

Accordingly, every property implied by condition (1) is enjoyed by all constraints in Optimality Theory.

THEOREM 3.9. *$\mathcal{U}$ being non-empty, v_c is a surjective total function with domain $\mathcal{U}$ and well-ordered range iff $C : \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\} \to \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ satisfies condition (1) as well as $C(\mathcal{K}) \subseteq \mathcal{K}$ for all $\mathcal{K} \in \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$. Moreover, $v_c(x) \leq v_c(y)$ iff*[11] *$x \in C(\{x, y\})$ for all x and y in $\mathcal{U}$.*

Theorem 3.9 means[12] that condition (1) characterizes constraints: c is a constraint[13] if and only if C satisfies[14] condition (1).

We are now in the position to substantiate, in the next section, our claim at the end of the introduction that Optimality Theory can be formalized through cumulative logics: We only need to show that cumulativity holds for all (logical) operations satisfying condition (1).

4. Constraints as cumulative operations

Outline. *In this section, we still consider the formalization of constraints as operations that select the best candidates. We show that this is equivalent with a class of operations over sets of formulas in a given logical language. As a result, we prove that candidate evaluation in Optimality Theory can be characterized by certain cumulative logics. Actually, we relate each operation C (Section 3) with a cumulative logic L such that $L(\overline{\mathcal{X}})$ consists of the formulas concluded from the complete set of formulas standing for "z is not a winner" for all z not in $\mathcal{X}$. We establish the following equivalence:*

$C(\mathcal{X})$ consists of the best candidates out of $\mathcal{X}$
if and only if
$L(\overline{\mathcal{X}})$ consists of all formulas standing for "z is not a winner" for all candidates z not in $C(\mathcal{X})$

Optimality Theory is about handling conflicts, in a specific sense. Importantly, it deals with conflicting rules (they may have exceptions

[11] Due to the conditions at hand, $x \in C(\{x, y\})$ is equivalent with the following: Whenever $\{x\} \subseteq \mathcal{K} \subseteq \mathcal{U}$, if $y \in C(\mathcal{K})$ then $x \in C(\mathcal{K})$.

[12] Theorem 3.9 does *not* mean that the correspondence between c (or v_c) and C is one-one. For example, consider $\mathcal{U} = \{x, y\}$. Let $v_{c_1}(x) = [\,]$ and $v_{c_1}(y) = [*]$. Also, let $v_{c_2}(x) = [**]$ and $v_{c_2}(y) = [***]$. That is, $v_{c_1}(y) \neq v_{c_2}(y)$. However, the abstract violation degrees induced by C_1 and C_2 share the same value, $\{x, y\}$, over y. On the other hand, let $v_{c_1}(x) = []$ and $v_{c_1}(y) = [*]$. Also, let $v_{c_2}(x) = [*]$ and $v_{c_2}(y) = []$. That is, $v_{c_1}(x) = v_{c_2}(y)$ whereas the abstract violation degrees induced by C_1 over x and by C_2 over y have distinct values (i.e., $\{x\}$ and $\{y\}$).

[13] Formally, c is an element of some $\mathcal{C}$ such that v_c is as indicated in Section 2 (obviously, requiring surjectivity amounts to disregarding unused violation degrees but all that results in no loss of generality).

[14] As we only consider operations such that $C(\mathcal{K}) \subseteq \mathcal{K}$ for all $\mathcal{K} \in \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$.

as noted at the end of the introduction). Now, reasoning with such rules[15] points to the so-called non-monotonic logics [Makinson 1994].

These logics aim at capturing tentative conclusions, like in the following example. Assume that you enter Mr. Johnson's office (you do not know what he looks like) and see a single desk in the room, with a man sitting behind it. Then, you expect him to be Mr. Johnson. In other words, you conclude that he is Mr. Johnson unless told otherwise. That is, you would withdraw that conclusion in the face of evidence to the contrary (e.g., the man tells you he is Mr. Johnson's associate). This is where monotonicity breaks down: The set of your conclusions may decrease while the set of statements you take for granted increases.

Of interest for the abstract analysis of non-monotonic logics is cumulativity as introduced by Makinson in the late eighties. Intuitively, cumulativity expresses that whenever something expected turns out to be true, then whatever else was expected is still expected (in the example, assume that you expected Mr. Johnson to be an accountant, then, after you find out that the man is actually Mr. Johnson, you do not stop expecting him to be an accountant).

More formally, cumulativity states that the set of conclusions (including the tentative ones) remains exactly the same when the premises of this set of conclusions are extended with some of these conclusions. Technically, cumulativity happens to be a property weaker than monotonicity. This is why it is of interest in the area of non-monotonic logics: Cumulative logics, even when non-monotonic, retain some of the nice behaviour of monotonic logics.

Let us now introduce the formal definition of cumulativity for arbitrary operations (not just logics).

A *cumulative operation*[16] over a poset $\langle D, \leq \rangle$ is a total function $f : D \to D$ which satisfies the conditions $(\alpha) - (\beta)$ for all a and b in D:

$$(\alpha) \quad a \leq f(a)$$
$$(\beta) \quad a \leq b \leq f(a) \Rightarrow f(a) = f(b)$$

Taking the poset to be the powerset of $\mathcal{U}$ ordered by the superset relation, constraints in Optimality Theory are cumulative operations:

THEOREM 4.1. *Let $C : \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\} \to \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ be such that $C(\mathcal{K}) \subseteq \mathcal{K}$ for all non-empty $\mathcal{K} \subseteq \mathcal{U}$. If C satisfies condition (1) then C is a cumulative operation (over $\langle \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}, \supseteq \rangle$).*

[15] The example to come does not necessarily involve conflicting rules. However, they can enter the picture as follows:

– You expect people to tell you their true surname, not a fake one.

– You expect people to work in their own office, not somebody's else.

[16] Comparing with a well-known notion, closure operations are a special case of cumulative operations.

COROLLARY 4.1. *Any constraint in Optimality Theory is a cumulative operation.*

All this suggests that every constraint (in Optimality Theory) embodies some form of cumulative reasoning from candidates.

The next step is that any constraint hierarchy in Optimality Theory is a combination of cumulative logics.

However, it first must be determined in what sense discarding suboptimal candidates amounts to applying a cumulative logic.

We define (rather restrictively) a logic as a function $L : \mathcal{P}(\mathcal{F}) \rightarrow \mathcal{P}(\mathcal{F})$ where $\mathcal{F}$ is the set of all formulas in a given logical language. We say that L is a cumulative logic if it is also a cumulative operation over $\langle \mathcal{P}(\mathcal{F}), \subseteq \rangle$.

On the other hand, it has just been shown above that a constraint is a cumulative operation over $\langle \mathcal{P}(\mathcal{U}), \supseteq \rangle$. This means that a constraint almost, but not quite, qualifies as a cumulative logic: Although in both cases the carrier is some powerset, the orderings at hand are the converse of one another. As a powerset ordered by set inclusion is a complemented (distributive) lattice, it is unproblematic to define a cumulative logic out of a constraint as follows. First, the logical language is fixed just by taking the set of all formulas in the language to be the set of all candidates.[17] Doing so does *not* mean that a candidate is to be represented by itself as a formula, it only means:

$$\mathcal{F} \stackrel{\text{def}}{=} \mathcal{U}$$

Second, let $h : \mathcal{P}(\mathcal{U}) \rightarrow \mathcal{P}(\mathcal{F})$ (or equivalently, $h : \mathcal{P}(\mathcal{U}) \rightarrow \mathcal{P}(\mathcal{U})$ because $\mathcal{F}$ is $\mathcal{U}$) be defined such that every set of candidates is mapped to a set of formulas by means of the following equality:

$$h(\mathcal{X}) \stackrel{\text{def}}{=} \overline{\mathcal{X}}$$

where $\overline{}$ stands for set complementation in $\mathcal{U}$ (i.e., $\overline{\mathcal{X}} = \mathcal{U} \setminus \mathcal{X}$).

These two steps are straightforward: Since the ordering over $\mathcal{P}(\mathcal{U})$ is to be reversed, a natural solution is to consider set complementation. The third step finally consists in mirroring C in the upside down lattice, through $L : \mathcal{P}(\mathcal{U}) \rightarrow \mathcal{P}(\mathcal{U})$, by making C and L to be *duals* as follows:

$$L(\mathcal{X}) \stackrel{\text{def}}{=} \overline{C(\overline{\mathcal{X}})}$$

[17] The reader may object that a logical language requires symbols, whereas $\mathcal{U}$ need not have anything to do with symbols. Clearly, this is well-taken but it is always possible to introduce a set $\mathcal{U}'$ of symbols which is in a bijective correspondence with $\mathcal{U}$ and the problem vanishes (but the formal properties become less readable because there will be an explicit symbol for the correspondence and it will occur repeatedly).

Of course, the underlying idea is that selecting the best candidates according to a constraint (abstracted by C) collapses to deduction in a cumulative logic (abstracted by L). The condition is that h be a homomorphism, in symbols,

$$h(C(\mathcal{X})) = L(h(\mathcal{X}))$$

and also be bijective (so that h is an isomorphism, actually an automorphism). The next theorem involves the fact that, given C and the above definitions, the condition is met.

THEOREM 4.2. *A constraint over a set of candidates in Optimality Theory is the dual of a cumulative logic.*

According to the interpretation (formally, the automorphism h), applying a constraint in Optimality Theory is merely reasoning about losers. Consider some set of candidates $\mathcal{X}$, it amounts to taking as premises that no candidate outside $\mathcal{X}$ is a current winner (obviously!). Further conclusions will be that other candidates, within $\mathcal{X}$, also fail to be current winners. That is, all statements (whether premises or conclusions) involved here are of the form "the candidate z is *not* a current winner". Clearly, the corresponding interpretation of any formula φ in $\mathcal{F}$ is thus that "x is not a current winner" for a particular[18] candidate x. Accordingly, cumulativity holds as this explains why, having concluded φ, using φ as an additional premise does not change the set of conclusions: Basically, if something has led us to conclude that "x is not a current winner" then that something is enough to have us reaching any conclusion "y is not a current winner" which we can draw when taking "x is not a current winner" as an additional premise (keep in mind that that "something" can only be a collection of statements of the form "the candidate z is not a current winner").

COROLLARY 4.2. *A hierarchy of constraints over a set of candidates in Optimality Theory is a composition of duals of cumulative logics.*

Such results greatly differ from previous work [Hammond 2000] [Besnard, Mercer, Schaub 2000] that provides some way to encode Optimality Theory in first-order logic or some non-classical logic such as default logic: Encoding is merely descriptive and need not give any insight about the nature of the theory being modelled.

So, applying constraints according to Optimality Theory is cumulative reasoning. Strictly speaking, not all cumulative logics are nonmonotonic (as was briefly mentioned, closure operations —they are

[18] Remember: $\mathcal{F} = \mathcal{U}$. For all $x \in \mathcal{U}$, it follows that $x \in \mathcal{F}$ is a formula standing for "the candidate x is not a current winner".

monotonic— are a special case of cumulative operations). However, it is easy to illustrate that applying constraints according to Optimality Theory is non-monotonic reasoning. Here is an example. Consider $\mathcal{U} = \{x, y, z\}$. Let C be such that $C(\{x, y, z\}) = \{x\}$. Stated otherwise, C picks x as the best candidate. Of course, $C(\{x, y\}) = \{x\}$ and $C(\{y\}) = \{y\}$. Hence, $C(\{y\}) \not\subseteq C(\{x, y\})$. That is,

$$C(\overline{\{x, z\}}) \not\subseteq C(\overline{\{z\}})$$

Equivalently,

$$\overline{C(\overline{\{z\}})} \not\subseteq \overline{C(\overline{\{x, z\}})}$$

Then, $L(\{z\}) \not\subseteq L(\{x, z\})$ by the definition of the automorphism h. In view of $\{z\} \subseteq \{x, z\}$, it follows that L is not monotonic (failing the monotonicity principle: if $X \subseteq Y$ then $L(X) \subseteq L(Y)$ for all X and Y). However, L is cumulative: if $X \subseteq Y \subseteq L(X)$ then $L(X) = L(Y)$ for all X and Y.

Intuitively, Optimality Theory is not monotonic: In the example just presented, "y is not a current winner" is concluded from the single premise "z is not a current winner" (meaning that z does not enter competition but x does). However, considering the additional premise "x is not a current winner" (meaning that x does not enter competition either, hence only y does) precludes "y is not a current winner" to hold because y being the only candidate evaluated must be the current winner. In other words, a case where premises extend the premises for another case lead to fewer conclusions: The set of conclusions does not expand monotonically with respect to the premises.

THEOREM 4.3. *A non-vacuous constraint over a set of candidates in Optimality Theory is the dual of a non-monotonic cumulative logic.*

Notice that the converse of Theorem 4.2 (and similarly Theorem 4.3) is untrue: There are many (non-monotonic) cumulative logics with no premise-free consequences while identity is the only operation satisfying (1) such that all candidates qualify as winners.[19]

Rather, the class of cumulative logics corresponding to Optimality Theory is as follows:

[19] A simple example of a cumulative logic that fails to correspond to a case in Optimality Theory is: $\mathcal{U} = \{e_1, e_2, e_3\}$ and the pseudo-constraint f is such that no candidate is eliminated when *all* three candidates compete but whenever only two of these candidates e_i and e_j compete then e_i wins over e_j as $i < j$ (in symbols, $f(\{e_1, e_2, e_3\}) = \{e_1, e_2, e_3\}$ and $f(\{e_1, e_2\}) = \{e_1\}$, etc). The corresponding logic $L(X) = \overline{f(\overline{X})}$ is cumulative: it is identity except $L(\{e_1\}) = \{e_1, e_3\}$ and $L(\{e_2\}) = L(\{e_3\}) = \{e_2, e_3\}$.

THEOREM 4.4. *Given a fixed set* $\mathcal{S}$, *let* $C : \mathcal{P}(\mathcal{S}) \setminus \{\emptyset\} \to \mathcal{P}(\mathcal{S}) \setminus \{\emptyset\}$ *and* $L : \mathcal{P}(\mathcal{S}) \setminus \{\mathcal{S}\} \to \mathcal{P}(\mathcal{S}) \setminus \{\mathcal{S}\}$ *be duals (i.e.,* $C(\mathcal{S} \setminus \mathcal{X}) = \mathcal{S} \setminus L(\mathcal{X})$ *for all* $\mathcal{X} \in \mathcal{P}(\mathcal{S}) \setminus \{\mathcal{S}\}$ *while* $L(\mathcal{S} \setminus \mathcal{X}) = \mathcal{S} \setminus C(\mathcal{X})$ *for all* $\mathcal{X} \in \mathcal{P}(\mathcal{S}) \setminus \{\emptyset\}$*).* c *is an arbitrary (resp. non-vacuous) constraint iff* L *is an arbitrary (resp. non-monotonic) cumulative logic that obeys the property below:*

$$(\ddagger) \qquad \textit{if } L(X \cap X') \neq L(X) \textit{ and } L(X \cap X') \neq L(X') \textit{ then } L(X \cap X') = L(X) \cap L(X')$$

The interpretation arising from the automorphism h thus indicates that reasoning in Optimality Theory is essentially disjunction-free as the logics mentioned in Theorem 4.4 require $X \vee X'$ to be in $L(X \cap X')$ whereas, in general, $X \vee X'$ need not have anything to do with the formulas in $X \cap X'$: At least, the canonical inference of $\varphi \vee \psi$ from each of φ and ψ must fail when φ and ψ are distinct atomic formulas.

5. Conclusion

We have investigated two alternative ways to formalize Optimality Theory, one where constraints are functions (over candidates) delivering violation marks and one where constraints are operations (over sets of candidates) selecting the winners.

We have shown that both approaches are equivalent, but in a sense stronger than simply yielding the same set of optimal candidates: We have shown that the latter approach deals with violation marks as much as the former approach does. As a surprising consequence, all this means that there is no loss of generality with using violation marks explicitly and dispensing with violation marks can only be apparent, not actual.

By means of a simple condition, we have furthermore introduced a natural characterization of constraints when formulated as operations over sets of candidates

Settling the issue of “The Logic of Optimality Theory”, we have also shown that applying a hierarchy of constraints in Optimality Theory is reasoning according to non-monotonic, cumulative, logics.

Acknowledgements

The authors are grateful to the referees for many constructive remarks.

References

[Besnard Ph., Mercer R. E., Schaub T. 2000]
Optimality Theory via Default Logic. Unpublished.

[Besnard Ph., Schaub T. 2000]
What is a (non-constructive) non-monotone logical system? *Theoretical Computer Science* 238:489-494, 2000.

[Frank R., Satta G. 1998]
Optimality theory and the generative complexity of constraint violability. *Computational Linguistics* 24(1):307-315, 1998.

[Ginsberg M. L. 1994]
AI and Nonmonotonic Reasoning. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertainty Reasoning*, Gabbay D. M., Hogger C. J., Robinson J. A. (eds), pages 1–33. Oxford University Press, 1994.

[Hammond M. 2000]
The Logic of Optimality Theory. ROA (390), 2000.
`http://roa.rutgers.edu`

[Kager R. 1999]
Optimality Theory. Cambridge Textbooks in Linguistics. Cambridge University Press, 1999.

[Makinson D. 1994]
General Patterns in Non-Monotonic Reasoning. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3: Nonmonotonic Reasoning and Uncertainty Reasoning*, Gabbay D. M., Hogger C. J., Robinson J. A. (eds), pages 35–110. Oxford University Press, 1994.

[Prince A. 2001]
Entailed Ranking Arguments. ROA (500), 2001.
`http://roa.rutgers.edu`

[Prince A., Smolensky P. 1993]
Optimality Theory: Constraint Interaction in Generative Grammars. Technical Report, Rutgers University, New Brunswick (NJ), and Computer Science Department, University of Colorado, Boulder. 1993.

[Samek-Lodovici V., Prince A. 1999]
Optima. ROA (363), 1999.
`http://roa.rutgers.edu`

[Tesar B., Smolensky P. 1998]
Learnability in Optimality Theory. *Linguistic Inquiry* 29:229-268, 1998.

Appendix

Proof (Theorem 2.1) *Let us first prove that $x \prec y$ implies $x \preceq y$. Let us assume the contrary. That is,*

(¶) *there exists c such that*
$$v_c(x) < v_c(y) \text{ and for all } c', \text{ if } v_{c'}(y) < v_{c'}(x) \text{ then } c' \ll c$$

(‖) *there exists c'' such that*
$$v_{c''}(y) < v_{c''}(x) \text{ and for all } c', \text{ if } c'' \ll c' \text{ then } v_{c'}(x) = v_{c'}(y)$$

As $\ll$ is a total order, either $c'' \ll c$ or $c \ll c''$. First, consider $c'' \ll c$. Applying (‖) yields $v_c(x) = v_c(y)$, which is a contradiction. Second, consider $c \ll c''$. In view of $v_{c''}(y) < v_{c''}(x)$, applying (¶) yields $c'' \ll c$. Then, both $c'' \ll c$ and $c \ll c''$ hold. A contradiction then arises, even when assuming that $\ll$ is not strict because $c'' = c$ entails that both $v_c(y) < v_c(x)$ and $v_c(x) < v_c(y)$ must hold.
Let us now prove that $x \prec y$ implies $y \not\preceq x$. According to (¶), $c' \ll c$ for all c' such that $v_{c'}(y) < v_{c'}(x)$. Since $\ll$ is an ordering (hence an antisymmetric relation), this means that every c' distinct from c such that $c \ll c'$ satisfies $v_{c'}(x) \leq v_{c'}(y)$ (due to the fact that $\leq$ is a total ordering). Then, it is enough to consider the greatest (in the sense of $\ll$) such c' which satisfies $v_{c'}(x) < v_{c'}(y)$ (the existence of this constraint c' comes from the fact that c obeys $v_c(x) < v_c(y)$ and from the fact that $\ll$ is a total ordering over a finite domain). By construction, this constraint c' is such that if $c' \ll c''$ then $v_{c''}(y) = v_{c''}(x)$. Together with the fact that $v_{c'}(x) < v_{c'}(y)$, this means that $y \not\preceq x$ (the constraint c' just constructed plays the role of the undesirable constraint c in the definition of $y \preceq x$).
There only remains to prove the other half of the theorem: $x \preceq y$ together with $y \not\preceq x$ imply $x \prec y$. Instead, let us prove that $y \not\preceq x$ implies $x \prec y$ (a stronger but unsurprising result). The definition of $\prec$ makes it clear that either $v_{c'}(y) = v_{c'}(x)$ for all c' or $x \prec y$ or $y \prec x$. Assuming $y \not\preceq x$ means that

there exists c'' such that
$$v_{c''}(x) < v_{c''}(y) \text{ and for all } c', \text{ if } c'' \ll c' \text{ then } v_{c'}(x) = v_{c'}(y)$$

which makes $v_{c'}(y) = v_{c'}(x)$ for all c' to fail. Moreover, $y \not\preceq x$ implies $y \not\prec x$ (because the first part of this proof shows that $x \prec y$ implies $x \preceq y$). That is, assuming $y \not\preceq x$ only leaves the possibility $x \prec y$ and the proof is over.

Proof (Corollary 2.1) *It clearly follows from Theorem 2.1 that* $x \prec y$ *implies* $x \preceq y$ *and* $v_c(x) \neq v_c(y)$ *for some* c. *As for proving the converse, let us assume* $x \preceq y$ *and* $v_c(x) \neq v_c(y)$ *for some* c. *Consider now the greatest (in the sense of* $\ll$*)* c'' *which satisfies* $v_{c''}(x) \neq v_{c''}(y)$ *(the existence of this constraint* c'' *comes from the fact that* c *obeys* $v_c(x) \neq v_c(y)$ *and from the fact that* $\ll$ *is a total ordering over a finite domain). By construction,* $c'' \ll c'$ *implies* $v_{c'}(x) = v_{c'}(y)$ *for all* c'. *According to the definition of* $x \preceq y$, *it is then impossible that* $v_{c''}(y) < v_{c''}(x)$. *Therefore,* $v_{c''}(x) < v_{c''}(y)$ *because* $v_{c''}(x) \neq v_{c''}(y)$. *For* $x \prec y$ *to hold, there then only remains to show that for all* c', *if* $v_{c'}(y) < v_{c'}(x)$ *then* $c' \ll c''$. *This again results from the construction for* c''*: The fact that* $c'' \ll c'$ *implies* $v_{c'}(x) = v_{c'}(y)$ *for all* c' *also means that* $v_{c'}(y) \neq v_{c'}(x)$ *(hence* $v_{c'}(y) < v_{c'}(x)$ *a fortiori) is possible only if* $c' \ll c''$. *Overall,* $x \prec y$.

Proof (Theorem 2.2) *The result comes from the equivalence between the two conditions below:*

$$\textit{for all } c' \neq c, \textit{ if } c \ll c' \textit{ then } v_{c'}(x) \leq v_{c'}(y)$$

$$\textit{for all } c' \neq c, \textit{ if } v_{c'}(y) < v_{c'}(x) \textit{ then } c' \ll c$$

Indeed, observe first that $v_{c'}(x) \leq v_{c'}(y)$ *is the contrary to* $v_{c'}(y) < v_{c'}(x)$ *and vice versa. Second, the same holds for* $c \ll c'$ *and* $c' \ll c$ *because* $\ll$ *too is a total order. Lastly, the case that* $c' = c$ *is examined in view of* $v_c(x) < v_c(y)$ *and the outcome is that*

$$\textit{for } c' = c, \textit{ if } c \ll c' \textit{ then } v_{c'}(x) \leq v_{c'}(y)$$

and

$$\textit{for } c' = c, \textit{ if } v_{c'}(y) < v_{c'}(x) \textit{ then } c' \ll c$$

then are both true: in the latter, the antecedent is false; in the former, the succedent is true.

Proof (Theorem 2.3) *As usual (i.e. apply Theorem 2.1):* x *is minimal wrt* $\prec$ *in* $\mathcal{U}$ *iff* $\not\exists y \in \mathcal{U}$ *s.t.* $y \prec x$; x *is minimal wrt* $\preceq$ *in* $\mathcal{U}$ *iff for all* $y \in \mathcal{U}$, *if* $y \preceq x$ *then* $x \preceq y$.

Proof (Theorem 2.4) *Let us assume that* x *is the unique optimal candidate. Reasoning by reductio ad absurdum, let us assume further that another candidate* y *exists such that for no* $c \in \mathcal{C}$ *do we have both* $v_c(x) < v_c(y)$ *and* $v_{c'}(x) = v_{c'}(y)$ *for all* c' *where* $c \ll c'$. *That is, there is no* c *such that* $v_c(x) < v_c(y)$ *and for all* c', *if* $c \ll c'$ *then* $v_{c'}(x) = v_{c'}(y)$. *Now, this simply is* $y \preceq x$. *A contradiction then arises regardless of whether* $x \preceq y$ *or* $x \not\preceq y$ *is the case: Firstly, consider* $x \not\preceq$

y. According to Theorem 2.1, $y \prec x$ follows and this (using Theorem 2.3) contradicts the assumption that x is the unique optimal candidate. Secondly, consider $x \preceq y$. As $y \preceq x$ has just been proven, this means that the only possibility is $v_{c'}(x) = v_{c'}(y)$ for all c' (verification can be done by induction starting with the greatest $c \in \mathcal{C}$: it trivially is such that $v_{c'}(y) = v_{c'}(x)$ whenever $c \ll c'$; therefore, both $v_c(y) < v_c(x)$ and $v_c(x) < v_c(y)$ must fail in view of $x \preceq y$ and $y \preceq x$). However, having x and y identical wrt every constraint in $\mathcal{C}$ contradicts the assumption that x is a unique optimal candidate.
We now show the reverse direction. That is, we prove that x is the unique optimal candidate whenever the following condition is satisfied: For every candidate $y \in \mathcal{U} \setminus \{x\}$, there exists $c \in \mathcal{C}$ such that $v_c(x) < v_c(y)$ and $v_{c'}(x) = v_{c'}(y)$ for all $c' \in \mathcal{C}$ where $c \ll c'$. Observe that this condition implies the ones in Theorem 2.2, so that $x \prec y$ holds for all $y \in \mathcal{U} \setminus \{x\}$. Then, applying Theorem 2.3 shows that x is an optimal candidate. Finally, applying Theorem 2.1 shows that x is the unique optimal candidate.

Proof (Theorem 2.5) *We prove the result by induction on the cardinality of $\mathcal{C}$ (or equivalently $\mathcal{C}'$).*
Base.
Let us consider the case where the set of all constraints $\mathcal{C}'$ is a singleton set.

$$\begin{aligned} W_{\ll}(\mathcal{U}, \{c\}) &= W_{\ll}(\{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}, \emptyset) \\ &= \{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\} \end{aligned}$$

Clearly, $v_c(x) \leq v_c(y)$ holds for all $y \in \mathcal{U}$ iff there exists no z in $\mathcal{U}$ such that $v_c(z) < v_c(x)$, that is, $z \prec x$ is possible for no z in $\mathcal{U}$ (Theorem 2.2) which is equivalent to x being minimal for $\prec$ in $\mathcal{U}$, i.e., x being optimal (Theorem 2.3).
Step.
Let us assume that the result holds when the cardinality of $\mathcal{C}'$ is $n \geq 1$ and let us show that the result then holds as well for $\mathcal{C}'$ having cardinality $n+1$. As for proving the if part of the result, assume $x \in W_{\ll}(\mathcal{U}, \mathcal{C}')$. So, $x \in W_{\ll}(\{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}, \mathcal{C}' \setminus \{c\})$. Applying the induction hypothesis, x is then optimal in $\{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}$ wrt $\mathcal{C}' \setminus \{c\}$. I.e., there exists no z in $\{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}$ such that for some $c'' \in \mathcal{C}' \setminus \{c\}$, $v_{c''}(z) < v_{c''}(x)$ and $v_{c'}(z) \leq v_{c'}(x)$ for all $c' \in \mathcal{C}' \setminus \{c\}$ satisfying $c'' \ll c'$ (Theorem 2.2). That is,

$$\nexists z \in \mathcal{U} \text{ s.t.} \begin{cases} \bullet\ \forall y \in \mathcal{U}, v_c(z) \leq v_c(y) \\ \bullet\ \exists c'' \in \mathcal{C}' \setminus \{c\} \text{ s.t. } \begin{cases} \bullet\ v_{c''}(z) < v_{c''}(x) \\ \bullet\ \forall c' \in \mathcal{C}' \setminus \{c\}, c'' \ll c' \Rightarrow v_{c'}(z) \leq v_{c'}(x) \end{cases} \end{cases}$$

The last subcondition can be extended to all $c' \in \mathcal{C}'$ *in view of* $\forall y \in \mathcal{U}, v_c(z) \leq v_c(y)$. *The case of* $c'' \in \mathcal{C}' \setminus \{c\}$ *can be extended to all* $c'' \in \mathcal{C}'$ *for the following reason: Taking* c'' *to be* c*, it is impossible for* $v_{c''}(z) < v_{c''}(x)$ *to hold because* $x \in W_{\ll}(\{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}, \mathcal{C}' \setminus \{c\})$*, i.e.,* $x \in \{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}$*. Summarizing,*

$$\not\exists z \in \mathcal{U} \text{ s.t.} \begin{cases} \bullet\ \forall y \in \mathcal{U}, v_c(z) \leq v_c(y) \\ \bullet\ \exists c'' \in \mathcal{C}' \text{ s.t. } \begin{cases} \bullet\ v_{c''}(z) < v_{c''}(x) \\ \bullet\ \forall c' \in \mathcal{C}', c'' \ll c' \Rightarrow v_{c'}(z) \leq v_{c'}(x) \end{cases} \end{cases}$$

We now show that $\forall y \in \mathcal{U}, v_c(z) \leq v_c(y)$ *(the first condition) is actually implied by the second condition. Indeed, the second condition clearly entails* $v_c(z) \leq v_c(x)$ *because* $c'' \ll c$ *for all* $c'' \in \mathcal{C}' \setminus \{c\}$*. However,* $x \in \{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}$ *due to* $x \in W_{\ll}(\{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}, \mathcal{C}' \setminus \{c\})$*. Therefore,* $\forall y \in \mathcal{U}, v_c(z) \leq v_c(y)$*. Hence, the first condition can be omitted and we obtain*

$$\not\exists z \in \mathcal{U} \text{ s.t. } \exists c'' \in \mathcal{C}' \text{ s.t.} \begin{cases} \bullet\ v_{c''}(z) < v_{c''}(x) \\ \bullet\ \forall c' \in \mathcal{C}', c'' \ll c' \Rightarrow v_{c'}(z) \leq v_{c'}(x) \end{cases}$$

Applying Theorem 2.2 then shows that x *is a minimal element for* $\prec$ *in* $\mathcal{U}$*. By Theorem 2.3, it follows that* x *is optimal.*
Let us show the reverse direction for the step case of the induction. We start with assuming that x *is optimal in* $\mathcal{U}$*. That is,* $y \not\prec x$ *for all* y *in* $\mathcal{U}$ *(Theorem 2.3). Theorem 2.2 then implies that, for all* y *in* $\mathcal{U}$*, there exists no* $c \in \mathcal{C}'$ *such that* $v_c(y) < v_c(x)$ *and* $v_{c'}(y) \leq v_{c'}(x)$ *for all* $c' \in \mathcal{C}'$ *satisfying* $c \ll c'$*. Equivalently,*

$$\forall y \in \mathcal{U}, \forall c \in \mathcal{C}', v_c(y) < v_c(x) \Rightarrow \exists c' \in \mathcal{C}' \text{ s.t. } c \ll c' \ \&\ v_{c'}(x) < v_{c'}(y)$$

From now on, c *is taken to be the greatest element in* $\mathcal{C}'$ *wrt* $\ll$*, and it clearly is such that there can be no* c' *as indicated in the above property. So,* $v_c(x) \leq v_c(y)$ *for all* $y \in \mathcal{U}$*. Hence* $x \in \{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}$*. However,* x *is optimal in* $\mathcal{U}$ *only if* x *is optimal in* $\{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}$ *wrt* $\mathcal{C}' \setminus \{c\}$ *(otherwise, there would exist* w *and some* c' *in* $\mathcal{C}' \setminus \{c\}$ *such that* $v_{c''}(w) \leq v_{c''}(x)$ *whenever* $c' \ll c'' \in \mathcal{C}' \setminus \{c\}$ *while* $v_c(w) \leq v_c(y)$ *for each* $y \in \mathcal{U}$ *but all this would contradict Theorem 2.2 and Theorem 2.3 as applied to the optimality of* x *in* $\mathcal{U}$*). Then, we can apply the induction hypothesis to conclude* $x \in W_{\ll}(\{x \in \mathcal{U} \mid \forall y \in \mathcal{U}, v_c(x) \leq v_c(y)\}, \mathcal{C}' \setminus \{c\})$*. Therefore,* $x \in W_{\ll}(\mathcal{U}, \mathcal{C}')$*.*

Proof (Theorem 3.1) *Let* $\{c_1, \ldots, c_n\} \subseteq \mathcal{C}$ *where* $c_n \ll \cdots \ll c_1$*.*

$$\begin{aligned} & W_{\ll}(\mathcal{K}, c_n \ll \cdots \ll c_1) \\ & = W_{\ll}(\{x \in \mathcal{K} \mid \forall y \in \mathcal{K}, v_{c_1}(x) \leq v_{c_1}(y)\}, c_n \ll \cdots \ll c_2) \end{aligned}$$

$$= W_{\ll}(C_1(\mathcal{K}), c_n \ll \cdots \ll c_2)$$
$$\vdots$$
$$= W_{\ll}(C_n(\ldots C_1(\mathcal{K})\ldots), \emptyset)$$
$$= C_n(\ldots C_1(\mathcal{K})\ldots)$$

Proof (Theorem 3.2) *By the definition of $C(X)$, (i) is trivial and (ii) is clear because if $v_c(y) < v_c(x)$ for some $y \in K$ then $v_c(y) < v_c(x)$ for some $y \in K'$ whenever $K \subseteq K'$. As for (iii), assume $X \cap C(X \cup Y) \neq \emptyset$. I.e., there exists $x \in X$ such that $v_c(x) \leq v_c(y)$ for all $y \in X \cup Y$. Now, all $z \in C(X)$ satisfy $v_c(z) \leq v_c(x)$ because $x \in X$. By transitivity, all $z \in C(X)$ enjoy $v_c(x) \leq v_c(y)$ for all $y \in X \cup Y$.*

Proof (Theorem 3.3) *The case $x = y$ is trivial, hence we assume that x and y are distinct. Let G' be the (non-empty) set of all non-empty $K' \subseteq \mathcal{U}$ such that $x \in C(K')$ while $y \notin C(K')$. Let G'' be the (non-empty) set of all non-empty $K'' \subseteq \mathcal{U}$ such that $y \in C(K'')$ while $x \notin C(K'')$. There are two cases, depending on whether for each $K' \in G'$ there exists $K'' \in G''$ such that $C(K' \cup K'') \neq C(K')$. The first case is when each $K' \in G'$ is as just indicated. In view of condition (1), this yields $C(K'') \subseteq C(K' \cup K'')$. Therefore, $y \in C(K' \cup K'')$. So, there exists some K (namely, $K' \cup K''$) such that $K' \subseteq K$ and $y \in C(K)$. As this holds for each $K' \in G'$, it follows that*

$$\left(\bigcup_{K' \in G'} K'\right) \subseteq \left(\bigcup_{y \in C(K)} K\right)$$

holds and a trivial consequence is then

$$\left(\bigcup_{x \in C(K)} K\right) \subseteq \left(\bigcup_{y \in C(K)} K\right)$$

which is $v_c(y) \leq v_c(x)$ as defined by $(*)$. The second case is the contrary to the first case: There exists some $K' \in G'$ such that for all $K'' \in G''$, $C(K' \cup K'') = C(K')$. Hence, for each $K'' \in G''$ there exists $K' \in G'$ such that $C(K' \cup K'') \neq C(K'')$. Apply now a similar reasoning as in the first case, so that

$$\left(\bigcup_{y \in C(K)} K\right) \subseteq \left(\bigcup_{x \in C(K)} K\right)$$

holds which is $v_c(x) \leq v_c(y)$ as defined by $(*)$. Overall, $v_c(x)$ and $v_c(y)$ are comparable.

Proof (Theorem 3.4) *Each constraint C_i ($i = 1..n$ for some n because the set of all constraints is finite) is a unary operation over the non-empty subsets of $\mathcal{U}$ that satisfies condition (1) and such that $C_i(K) \subseteq K$ for every $K \subseteq \mathcal{U}$ (as was already indicated, we implicitly extend C_i with $C_i(\emptyset) = \emptyset$).*
A noticeable consequence is $C_i(\{z\}) = \{z\}$ for all $z \in \mathcal{U}$.
The corresponding valuations are obtained using an alternative definition for v_{c_i} as introduced in footnote 9:

$$v_{c_i}(x) \stackrel{\text{def}}{=} \{y \in \mathcal{U} \mid x \in C_i(\{x, y\})\}$$

Indeed, consider the definition

$$v_c(z) \stackrel{\text{def}}{=} \bigcup_{\substack{K' \subseteq \mathcal{U} \\ z \in C(K')}} K'$$

We show, for all $z \in \mathcal{U}$,

$$v_c(z) = \{w \in \mathcal{U} \mid z \in C(\{w, z\})\}$$

Clearly, $\{w \in \mathcal{U} \mid z \in C(\{w, z\})\} \subseteq v_c(z)$. As for the converse inclusion, let $y \in K'$ be such that $z \in C(K')$. We only consider the non-trivial case, i.e. $K' \neq \{y, z\}$. By $y \in K'$ and $z \in C(K') \subseteq K'$, condition (1) can apply for $C(K')$ to be either $C(K' \setminus \{y, z\}) \cup C(\{y, z\})$ or $C(\{y, z\})$ or $C(K' \setminus \{y, z\})$. Due to $C(K' \setminus \{y, z\}) \subseteq K' \setminus \{y, z\}$ and $z \in C(K')$, we obtain $z \in C(\{y, z\})$ so that the converse inclusion is also proven and the equality of definitions holds:

$$\{w \in \mathcal{U} \mid z \in C(\{w, z\})\} = \bigcup_{\substack{K' \subseteq \mathcal{U} \\ z \in C(K')}} K'$$

For any subset M' of $M = \bigcup_i \{v_{c_i}(z) \mid z \in \mathcal{U}\}$, let

$$K'_{c_i} \stackrel{\text{def}}{=} \{z \in \mathcal{U} \mid v_{c_i}(z) \in M'\}$$

and let

$$M'_{c_i} \stackrel{\text{def}}{=} \{v_{c_i}(z) \mid z \in K'_{c_i}\}$$

Clearly,

$$M' = M'_{c_1} \cup \cdots \cup M'_{c_n}$$

To start with, we prove $v_{c_i}(x) \subseteq v_{c_i}(z)$ for all $x \in K'_{c_i}$ and all $z \in C_i(K'_{c_i})$ in order to show that M'_{c_i} has a maximal element wrt set inclusion.

Let $x \in K'_{c_i}$. Let $y \in v_{c_i}(x)$. Let $z \in C_i(K'_{c_i})$. We need to show $y \in v_{c_i}(z)$.
First, assume $y \in K'_{c_i}$. Then, $C_i(K'_{c_i}) = C_i(K'_{c_i} \cup \{y\}) = C_i((K'_{c_i} \setminus \{z\}) \cup \{y,z\})$ which is either $C_i(K'_{c_i} \setminus \{z\})$ or $C_i(\{y,z\})$ or $C_i(K'_{c_i} \setminus \{z\}) \cup C_i(\{y,z\})$ by condition (1). Due to $C_i(K'_{c_i} \setminus \{z\}) \subseteq K'_{c_i} \setminus \{z\}$ and $z \in C_i(K'_{c_i})$, the first case is impossible and each of the other two cases implies that z is in $C_i(\{y,z\})$. Hence, $y \in v_{c_i}(z)$.
That is, we are done with $y \in K'_{c_i}$ and, from now on, we can then assume the opposite possibility $y \notin K'_{c_i}$ (together with its immediate consequence $x \neq y$).
Second, assume $y \notin C_i(K'_{c_i} \cup \{y\})$. In view of $C_i(\{y\}) = \{y\}$, it follows that $C_i(K'_{c_i} \cup \{y\}) \neq C_i(\{y\})$ and $C_i(K'_{c_i} \cup \{y\}) \neq C_i(K'_{c_i}) \cup C_i(\{y\})$. For condition (1) not to be violated, $C_i(K'_{c_i} \cup \{y\}) = C_i(K'_{c_i})$ must hold. So, $C_i((K'_{c_i} \setminus \{z\}) \cup \{y,z\}) = C_i(K'_{c_i})$. By condition (1), $C_i((K'_{c_i} \setminus \{z\}) \cup \{y,z\})$ is either $C_i(K'_{c_i} \setminus \{z\})$ or $C_i(\{y,z\})$ or $C_i(K'_{c_i} \setminus \{z\}) \cup C_i(\{y,z\})$. Due to $C_i(K'_{c_i} \setminus \{z\}) \subseteq K'_{c_i} \setminus \{z\}$ and $z \in C_i(K'_{c_i})$, the first case is impossible and each of the other two cases implies that z is in $C_i(\{y,z\})$. That is, $y \in v_{c_i}(z)$.
Third, assume $y \in C_i(K'_{c_i} \cup \{y\})$ (and $y \notin K'_{c_i}$). Applying condition (1), $C_i(K'_{c_i} \cup \{y\})$ is either $C_i(K'_{c_i})$ or $C_i(\{y\})$ or $C_i(K'_{c_i}) \cup C_i(\{y\})$. According to $C_i(K'_{c_i}) \subseteq K'_{c_i}$ and $y \notin K'_{c_i}$ while $y \in C_i(K'_{c_i} \cup \{y\})$, the first case is impossible. As for the second case, it simply means that $C_i(K'_{c_i} \cup \{y\}) = C_i(\{y\}) = \{y\}$. However, $C_i(K'_{c_i} \cup \{y\}) = C_i((K'_{c_i} \setminus \{x\}) \cup \{x,y\})$ which is either $C_i(K'_{c_i} \setminus \{x\})$ or $C_i(\{x,y\})$ or $C_i(K'_{c_i} \setminus \{x\}) \cup C_i(\{x,y\})$ due to condition (1). The first subcase is impossible because $C_i(K'_{c_i} \setminus \{x\}) \subseteq K'_{c_i} \setminus \{x\} \not\subseteq \{y\}$ (cf $y \notin K'_{c_i}$). Each of the other two subcases implies $C_i(\{x,y\}) \subseteq \{y\}$ which is impossible because $y \in v_{c_i}(x)$ and $x \neq y$ (that follows from $y \notin K'_{c_i}$ as mentioned above). As a consequence, the third case must hold. In symbols, $C_i(K'_{c_i} \cup \{y\}) = C_i(K'_{c_i}) \cup C_i(\{y\})$. Therefore, $C_i(K'_{c_i}) \cup C_i(\{y\}) = C_i((K'_{c_i} \setminus \{z\}) \cup \{y,z\})$ which is either $C_i(K'_{c_i} \setminus \{z\})$ or $C_i(\{y,z\})$ or $C_i(K'_{c_i} \setminus \{z\}) \cup C_i(\{y,z\})$ by condition (1). Due to $C_i(K'_{c_i} \setminus \{z\}) \subseteq K'_{c_i} \setminus \{z\}$ while $z \in C_i(K'_{c_i}) \subseteq C_i(K'_{c_i}) \cup C_i(\{y\})$, the first alternative is impossible and each of the other two implies that z is in $C_i(\{y,z\})$. That is, $y \in v_{c_i}(z)$.
Summing up, all cases (whether $y \in K'_{c_i}$ or $y \notin C_i(K'_{c_i} \cup \{y\})$ or $y \in C_i(K'_{c_i} \cup \{y\}) \setminus K'_{c_i}$ as just considered) entail $y \in v_{c_i}(z)$. That is, $v_{c_i}(x) \subseteq v_{c_i}(z)$ for all $x \in K'_{c_i}$ and all $z \in C_i(K'_{c_i})$.
As each element of M'_{c_i} is $v_{c_i}(x)$ for some $x \in K'_{c_i}$, it follows that each element of M'_{c_i} is a subset of any element S of M'_{c_i} which is $v_{c_i}(z)$ for some $z \in \mathcal{U}$ such that $z \in C_i(K'_{c_i})$ (there clearly exists at least one such S unless $K'_{c_i} = M'_{c_i} = \emptyset$). Since set inclusion is an ordering relation, all this means that S is unique and is the greatest element of M'_{c_i}.
So, the greatest element exists for each M'_{c_i} in each M'. This means that

no M'_{c_i} contain a chain which is infinitely ascending for set inclusion. As there are only finitely many M'_{c_i} from M', it follows that M (as well as any M') is well-founded for the superset relation.
Linearity proven in Theorem 3.3 allows us to conclude that each $M_i = \{v_{c_i}(z) \mid z \in \mathcal{U}\}$ is well-ordered by the superset relation.

Proof (Theorem 3.5) *Clearly, $C \circ C'$ is defined from $\mathcal{P}(\mathcal{U})$ to $\mathcal{P}(\mathcal{U})$ and $(C \circ C')(X) \subseteq X$ for all $X \subseteq \mathcal{U}$. It only remains to be shown that $C \circ C'$ satisfies condition (1). Consider the image of some $X \cup Y$ by $C \circ C'$, in symbols $C(C'(X \cup Y))$. As C' obeys condition (1), $C'(X \cup Y)$ is either $C'(X)$ or $C'(Y)$ or $C'(X) \cup C'(Y)$. The first two cases yield that $C(C'(X \cup Y))$ can be $C(C'(X))$ or $C(C'(Y))$. The last case is that $C(C'(X \cup Y))$ can be $C(C'(X) \cup C'(Y))$. As C satisfies condition (1), $C(C'(X) \cup C'(Y))$ is either $C(C'(X))$ or $C(C'(Y))$ or $C(C'(X)) \cup C(C'(Y))$. All five possibilities (the two cases above and the three subcases of the third case) amount to the formulation of condition (1) for $C \circ C'$ in the form: $C(C'(X \cup Y))$ is either $C(C'(X))$ or $C(C'(Y))$ or $C(C'(X)) \cup C(C'(Y))$.*

Proof (Theorem 3.6) *The case $C(X) = \emptyset$ is trivial. The case $C(X) = X$ is also trivial. So, we only need considering a non-empty X such that $X \setminus C(X) \neq \emptyset$. Assume $C(X \setminus C(X)) \subseteq C(C(X) \cup (X \setminus C(X)))$. Then, $C(X \setminus C(X)) \subseteq C(X)$ because $C(X) \cup (X \setminus C(X))$ is X. This contradicts $C(X \setminus C(X)) \subseteq X \setminus C(X)$ and it follows that the assumption must fail: $C(X \setminus C(X)) \not\subseteq C(C(X) \cup (X \setminus C(X)))$. As a consequence, $C(C(X) \cup (X \setminus C(X))) \neq C(X \setminus C(X))$ and $C(C(X) \cup (X \setminus C(X))) \neq C(C(X)) \cup C(X \setminus C(X))$. Taking K to be $C(X)$ and K' to be $X \setminus C(X)$, condition (1) is then contradicted unless $C(C(X) \cup (X \setminus C(X))) = C(C(X))$. That is, $C(X) = C(C(X))$.*

Proof (Corollary 3.1) *$C_l(\ldots C_j(X) \ldots)$ is taken to be non-empty because the case $C_l(\ldots C_j(X) \ldots) = \emptyset$ is trivial.*
Observe that $C_l(\ldots C_j(X) \ldots) \subseteq C_j(X)$ and $C_j(X \setminus C_l(\ldots C_j(X) \ldots)) \subseteq X \setminus C_l(\ldots C_j(X) \ldots)$. Assuming $C_j(X) \subseteq C_j(X \setminus C_l(\ldots C_j(X) \ldots))$ would then imply $C_l(\ldots C_j(X) \ldots) \subseteq X \setminus C_l(\ldots C_j(X) \ldots)$ which is impossible because $C_l(\ldots C_j(X) \ldots) \neq \emptyset$. So, the assumption is false. Therefore, $C_j(X) \not\subseteq C_j(X \setminus C_l(\ldots C_j(X) \ldots))$. Hence, $C_j(X) \neq C_j(X \setminus C_l(\ldots C_j(X) \ldots))$. As X is $C_l(\ldots C_j(X) \ldots)$ (indeed, $C_l(\ldots C_j(X) \ldots)$ is clearly a subset of X) unioned with $X \setminus C_l(\ldots C_j(X) \ldots)$, applying condition (1) would then yield a contradiction unless either

$$C_j(X) = C_j(C_l(\ldots C_j(X) \ldots))$$

or

$$C_j(X) = C_j(C_l(\ldots C_j(X) \ldots)) \cup C_j(X \setminus C_l(\ldots C_j(X) \ldots))$$

In view of $C_l(\ldots C_j(X)\ldots) \subseteq C_j(X)$, *it follows that* $C_l(\ldots C_j(X)\ldots) \subseteq C_j(C_l(\ldots C_j(X)\ldots)) \cup C_j(X \setminus C_l(\ldots C_j(X)\ldots))$. *Due to* $C_j(X \setminus C_l(\ldots C_j(X)\ldots)) \subseteq X \setminus C_l(\ldots C_j(X)\ldots)$, *this then yields* $C_l(\ldots C_j(X)\ldots) \subseteq C_j(C_l(\ldots C_j(X)\ldots))$. *The converse inclusion is obvious:* $C_j(C_l(\ldots C_j(X)\ldots)) \subseteq C_l(\ldots C_j(X)\ldots)$.

Proof (Theorem 3.7) *Let us prove (2) (and (3) likewise). It is enough to observe that (1) can be rewritten as* $C(X \cup Y) = C(X)$ *or* $C(X \cup Y) = C(Y)$ *or* $C(X \cup Y) = C(X) \cup C(Y)$.
Let us prove (4). Assume $X \subseteq C(Y)$. *Hence* $X \subseteq Y$ *because* $C(Y) \subseteq Y$. *Observe that* $C(Y \setminus X) \subseteq Y \setminus X$. *So, the assumption makes it impossible that* $C(X \cup (Y \setminus X)) = C(Y \setminus X)$. *For (1) not to be contradicted as applied to* $X \cup (Y \setminus X)$ *it then must be the case that either* $C(X \cup (Y \setminus X)) = C(X)$ *or* $C(X \cup (Y \setminus X)) = C(X) \cup C(Y \setminus X)$. *That is,* $C(Y) = C(X)$ *or* $C(Y) = C(X) \cup C(Y \setminus X)$. *Then the assumption now entails* $X \subseteq C(X) \cup C(Y \setminus X)$ *hence* $X \subseteq C(X)$ *in view of* $C(Y \setminus X) \subseteq Y \setminus X$.
Let us prove (5). Assume $X \subseteq Y$. *Consider* $z \in X \setminus C(X)$. *Therefore,* $z \in Y$. *Then, there only remains to show that* $z \notin C(Y)$. *Clearly,* $z \in X$ *hence* $z \notin C(Y \setminus X)$ *because* $C(Y \setminus X) \subseteq Y \setminus X$. *Now,* $z \notin C(X)$ *and* $z \notin C(Y \setminus X)$ *show that* $z \notin C(X \cup (Y \setminus X))$ *because (1), as applied to* $X \cup (Y \setminus X)$, *requires* $C(X \cup (Y \setminus X))$ *to consist at most of* $C(X)$ *and* $C(Y \setminus X)$. *So,* $z \notin C(Y)$ *immediately follows because* $X \cup (Y \setminus X)$ *is* Y.
Let us prove (6). To simplify notation, we write Y' *for* $Y \setminus (X \cap Y)$ *(i.e.,* $X \cup Y' = X \cup Y$ *but* $X \cap Y' = \emptyset$*). Assume* $X \cap C(X \cup Y) \neq \emptyset$. *The case* $C(X) = C(X \cup Y')$ *is trivial. So we are left with* $C(X \cup Y') \neq C(X)$. *Observe that* $C(X \cup Y') \neq C(Y')$ *due to the assumption and the fact that* $C(Y') \subseteq Y'$. *Then (1) entails* $C(X \cup Y') = C(X) \cup C(Y')$. *So,* $C(X) \subseteq C(X \cup Y)$.
Let us prove (7). The assumption $X \subseteq Y \setminus C(Y)$ *implies* $X \subseteq Y$, *so* $C(X \cup (Y \setminus X)) = C(Y)$. *The first case is* $C(Y) = \emptyset$ *and* $C(X) \neq \emptyset$. *The second case is* $C(X) = \emptyset$ *and* $C(Y) \neq \emptyset$. *In either case,* $C(X \cup (Y \setminus X)) \neq C(X)$ *and assuming* $C(X \cup (Y \setminus X)) \neq C(Y \setminus X)$ *would then let (1) to apply, yielding a contradiction. There only remains the case that* $C(X)$ *and* $C(Y)$ *are non-empty. In view of* $C(X) \subseteq X$, *assuming* $X \subseteq Y \setminus C(Y)$ *requires* $C(X \cup (Y \setminus X))$ *to be distinct from* $C(X)$ *and* $C(X) \cup C(Y \setminus X)$. *By (1),* $C(X \cup (Y \setminus X)) = C(Y \setminus X)$.

Proof (Theorem 3.8) *In order to prove (1), let us assume* $C(X \cup Y) \neq C(X)$ *and* $C(X \cup Y) \neq C(Y)$.
Observe that (ii) has the following two instances. First, if $z \in X$ *but* $z \notin C(X)$ *then* $z \notin C(X \cup Y)$. *Second, if* $z \in Y$ *but* $z \notin C(Y)$ *then* $z \notin C(X \cup Y)$. *The contrapositives are as follows. First,* $z \in C(X \cup Y)$ *implies that if* $z \in X$ *then* $z \in C(X)$. *Second,* $z \in C(X \cup Y)$ *implies that if* $z \in Y$ *then* $z \in C(Y)$. *So,* $z \in C(X \cup Y)$ *implies that if* $z \in X$ *or* $z \in Y$ *then* $z \in C(X)$ *or* $z \in C(Y)$. *By (i), if* $z \in C(X \cup Y)$ *then* $z \in X$

*or $z \in Y$. Therefore, $z \in C(X \cup Y)$ implies $z \in C(X)$ or $z \in C(Y)$.
Hence, assuming $Y \cap C(X \cup Y) = \emptyset$ yields that $z \in C(X \cup Y)$ implies $z \in C(X)$ in view of (i). In symbols, $C(X \cup Y) \subseteq C(X)$. Due to $Y \cap C(X \cup Y) = \emptyset$, (i) and $C(X \cup Y) \neq \emptyset$ entail $X \cap C(X \cup Y) \neq \emptyset$. By (iii), $C(X) \subseteq C(X \cup Y)$. As we have already proved $C(X \cup Y) \subseteq C(X)$, the assumption $C(X \cup Y) \neq C(X)$ is now contradicted. So, we can dismiss the extra assumption $Y \cap C(X \cup Y) = \emptyset$. Of course, we can dismiss $X \cap C(X \cup Y) = \emptyset$ in a similar way.
Then, $X \cap C(X \cup Y) \neq \emptyset$ and $Y \cap C(X \cup Y) \neq \emptyset$. Applying (iii), $z \in C(X)$ or $z \in C(Y)$ implies $z \in C(X \cup Y)$. Overall, $z \in C(X \cup Y)$ iff $z \in C(X)$ or $z \in C(Y)$.*

Proof (Theorem 3.9) *For the if direction, apply Theorem 3.4 with the definition of v_c and $(*)$ that precede it (considering a family with exactly one C_i). For the only if direction, apply Theorem 3.2 with the definition of C preceding it and apply Theorem 3.8.
So, it is possible to map a valuation to a powerset operation obeying condition (1) and it is possible to map a powerset operation obeying condition (1) to a valuation. Verification that these two transformations are essentially inverses of each other is as follows.
As for the first direction, we have to show that mapping a valuation v_c to a powerset operation C and mapping back C to a valuation results in a valuation which is isomorphic to v_c. Here is how we prove this: Let $c \in \mathcal{C}$. Define*

$$C(K) \stackrel{\text{def}}{=} \{x \in K \mid v_c(x) \leq v_c(y) \text{ for all } y \in K\}$$

and

$$v_{c'}(z) \stackrel{\text{def}}{=} \bigcup_{\substack{K' \subseteq \mathcal{U} \\ z \in C(K')}} K'$$

Now, $K' \subseteq \mathcal{U}$ is such that $z \in C(K')$ iff $z \in \{u \in K' \mid v_c(u) \leq v_c(y)$ for all $y \in K'\}$. This happens iff $v_c(z)$ is the least element in the image of K' by v_c (remember that $\leq$ is a well-ordering for the image of $\mathcal{U}$, as well as any of its subsets, by v_c). This is equivalent to K' being a subset of $\{u \in \mathcal{U} \mid v_c(z) \leq v_c(u)\}$. Therefore,

$$\bigcup_{\substack{K' \subseteq \mathcal{U} \\ z \in C(K')}} K' = \{u \in \mathcal{U} \mid v_c(z) \leq v_c(u)\}$$

So, $v_{c'}(z) = \{u \in \mathcal{U} \mid v_c(z) \leq v_c(u)\}$. Clearly, $v_{c'}(x) \subseteq v_{c'}(w)$ iff $v_c(w) \leq v_c(x)$ (because $\leq$ is a well-ordering, hence a total order, for *the image of* $\mathcal{U}$ *by* v_c*). Applying* $(*)$*,* $v_{c'}(w) \leq v_{c'}(x)$ *iff* $v_c(w) \leq v_c(x)$ *for all* w *and* x *in* $\mathcal{U}$*.*

Also, $v_c(x) \leq v_c(y)$ *iff* $x \in C(\{x, y\})$ *(cf the way* $C(K)$ *is defined).*

As for the other direction, we must show that mapping a powerset operation C *to a valuation* v_c *and mapping back* v_c *to a powerset operation results in* C*. We prove this as follows.*

Consider $C : \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\} \to \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ *that satisfies condition (1) and is such that* $C(K) \subseteq K$ *for all non-empty subsets* K *of* $\mathcal{U}$*. Define*

$$v_c(z) \stackrel{\text{def}}{=} \bigcup_{\substack{K' \subseteq \mathcal{U} \\ z \in C(K')}} K'$$

We first show, for all $z \in \mathcal{U}$*,*

$$v_c(z) = \{w \in \mathcal{U} \mid z \in C(\{w, z\})\}$$

Clearly, $\{w \in \mathcal{U} \mid z \in C(\{w, z\})\} \subseteq v_c(z)$*. As for the converse inclusion, let* $y \in K'$ *such that* $z \in C(K')$*. The non-trivial case is* $K' \neq \{y, z\}$*. By* $y \in K'$ *and* $z \in C(K') \subseteq K'$*, condition (1) can apply for* $C(K')$ *to be either* $C(K' \setminus \{y, z\}) \cup C(\{y, z\})$ *or* $C(\{y, z\})$ *or* $C(K' \setminus \{y, z\})$*. Due to* $C(K' \setminus \{y, z\}) \subseteq K' \setminus \{y, z\}$ *and* $z \in C(K')$*, we obtain* $z \in C(\{y, z\})$ *so that the converse inclusion is also proven.*

Next, define

$$C'(K) \stackrel{\text{def}}{=} \{x \in K \mid v_c(x) \leq v_c(y) \textit{ for all } y \in K\}$$

Then, $C'(K) = \{x \in K \mid v_c(y) \subseteq v_c(x) \textit{ for all } y \in K\}$ *by* $(*)$ *and*

$$C'(K) = \Big\{x \in K \mid \ \forall y \in K,\ \{w \in \mathcal{U} \mid y \in C(\{w, y\})\} \subseteq \{w \in \mathcal{U} \mid x \in C(\{w, x\})\}\Big\}$$

That is, $x \in C'(K)$ *iff* $x \in K$ *has the following property*

$$\forall y \in K \quad \forall w \in \mathcal{U} \qquad y \in C(\{w, y\}) \Rightarrow x \in C(\{w, x\})$$

We proceed to show $C' = C$ *by proving that* x *obeys the property iff* x *is in* $C(K)$*.*

We first show that $x \in C(K)$ *implies the property. So, we assume* $x \in C(K)$*. Let* $w \in \mathcal{U}$ *and* $y \in K$ *such that* $y \in C(\{w, y\})$*. Let* K_w *abbreviate* $K \cup \{w\}$ *(which is just* K *for* $w \in K$*). Observe that* $x \in C(\{w, x\})$ *(what we must show) holds if* $x \in C(K_w)$*. Indeed,* $C(K_w)$ *is either* $C(K \setminus \{x\}) \cup C(\{w, x\})$ *or* $C(K \setminus \{x\})$ *or* $C(\{w, x\})$*. In view of* $x \in C(K_w)$ *and* $C(K \setminus \{x\}) \subseteq K \setminus \{x\}$*, the second case is impossible and either of the remaining two cases implies* $x \in C(\{w, x\})$*. Hence, we are left with proving* $x \in C(K_w)$*. We consider two possibilities. (i)*

Assume $w \in C(K_w)$. *Should* w *be in* K, $C(K_w) = C(K)$ *hence the conclusion follows:* $x \in C(K_w)$. *So, we can assume* $w \notin K$. *By condition (1),* $C(K_w)$ *is either* $C(K \setminus \{w,y\}) \cup C(\{w,y\})$ *or* $C(K \setminus \{w,y\})$ *or* $C(\{w,y\})$. *The second case is impossible, due to the assumption* $w \in C(K_w)$ *and* $C(K \setminus \{w,y\}) \subseteq K \setminus \{w,y\}$. *Therefore,* $y \in C(K_w)$. *Due to condition (1),* $C(K_w)$ *is either* $C(K) \cup C(\{w\})$ *or* $C(K)$ *or* $C(\{w\})$. *The third case is impossible because* $y \in C(K_w)$ *(clearly,* $w \neq y$ *due to* $w \notin K$ *and* $y \in K$*). So,* $C(K_w) = C(K) \cup C(\{w\})$. *In view of* $x \in C(K)$, *it follows that* $x \in C(K_w)$. *(ii) Assume alternatively* $w \notin C(K_w)$. *This implies* $C(\{w\}) \not\subseteq C(K_w)$ *hence* $C(K_w) \neq C(\{w\})$ *and* $C(K_w) \neq C(K) \cup C(\{w\})$ *so that* $C(K_w) = C(K)$ *by condition (1). Therefore,* $x \in C(K_w)$.
We now show that the property implies $x \in C(K)$. *As* $C(K)$ *is non-empty, there exists* $w \in C(K) \subseteq K$ *so that identifying* w *with* y *in the property yields* $C(\{w,x\})$. *Applying condition (1),* $C(K)$ *is either* $C(K \setminus \{w,x\}) \cup C(\{w,x\})$ *or* $C(\{w,x\})$ *or* $C(K \setminus \{w,x\})$. *Due to* $w \in C(K)$, *the third case is impossible and it follows that* $C(\{w,x\}) \subseteq C(K)$, *hence* $x \in C(K)$.
Also, it is obvious that $v_c(x) \leq v_c(y)$ *iff* $x \in C'(\{x,y\}) = C(\{x,y\})$.

Proof (Theorem 4.1) *Extending* C *so as to have* $C(\emptyset) = \emptyset$, *we show that* C *is a cumulative operation over* $\langle \mathcal{P}(\mathcal{U}), \supseteq \rangle$. *That is, we show that for all subsets* X *and* Y *of* $\mathcal{U}$,

$$X \supseteq Y \supseteq C(X) \Rightarrow C(X) = C(Y)$$

Or, equivalently:

$$C(X \cup Y) \subseteq Y \Rightarrow C(X \cup Y) = C(Y)$$

We only consider the non-trivial case $X \not\subseteq Y$. *Assume* $C(X \cup Y) \subseteq Y$ *so that we must prove* $C(X \cup Y) = C(Y)$. *First of all,* $C(X \setminus (X \cap Y)) \subseteq X \setminus (X \cap Y)$. *As* $X \setminus (X \cap Y)$ *and* Y *are clearly disjoint, the assumption implies* $C(X \setminus (X \cap Y)) \not\subseteq C(X \cup Y)$ *(indeed,* $C(X \setminus (X \cap Y))$ *is non-empty because* $X \setminus (X \cap Y) \neq \emptyset$ *in view of* $X \not\subseteq Y$*). It then follows that* $C((X \setminus (X \cap Y)) \cup Y) \neq C(X \setminus (X \cap Y))$ *and* $C((X \setminus (X \cap Y)) \cup Y) \neq C(X \setminus (X \cap Y)) \cup C(Y)$ *(due to* $C(X \cup Y) = C((X \setminus (X \cap Y)) \cup Y)$ *which is a trivial fact). For condition (1) not to be contradicted, it then must be the case that* $C((X \setminus (X \cap Y)) \cup Y) = C(Y)$. *That is,* $C(X \cup Y) = C(Y)$.

Proof (Theorem 4.2) *By Theorem 3.9 and the proof of Theorem 4.1 (where it is shown that* C *is a cumulative operation over* $\langle \mathcal{P}(\mathcal{U}), \supseteq \rangle$*) together with the fact that* h *is bijective, it is enough to prove that* h *is a homomorphism:*

$$h(C(X)) = \mathcal{U} \setminus C(X) = \overline{C(X)} = \overline{C(\overline{\overline{X}})} = L(\overline{X}) = L(\mathcal{U} \setminus X) = L(h(X))$$

Indeed, it follows that L is a cumulative logic (i.e., a cumulative operation over $\langle \mathcal{P}(\mathcal{U}), \subseteq \rangle$) because

$$\overline{X} \supseteq \overline{Y} \supseteq C(\overline{X}) \Rightarrow C(\overline{X}) = C(\overline{Y})$$

is then equivalent to

$$X \subseteq Y \subseteq L(X) \Rightarrow L(X) = L(Y)$$

Proof (Theorem 4.3) *In view of Theorem 4.2, consider the case that there exists $x \in \mathcal{U} \setminus C(\mathcal{U})$. As C is an operation over $\mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$ such that $C(K) \subseteq K$ for all $K \in \mathcal{P}(\mathcal{U}) \setminus \{\emptyset\}$, it follows that $y \in C(\mathcal{U})$ for some $y \in \mathcal{U}$. So, a consequence of applying condition (1) to $\mathcal{U} \setminus \{x,y\}$ unioned with $\{x,y\}$, is that $C(\{x,y\}) \subseteq C(\mathcal{U})$. Then, it is easy to show that $C(\{x,y\}) = \{y\}$. Hence, $C(\{x\}) \not\subseteq C(\{x,y\})$. Using h and reversing $\not\subseteq$ by set complementation, $L(\mathcal{U} \setminus \{x,y\}) \not\subseteq L(\mathcal{U} \setminus \{x\})$.*

Proof (Theorem 4.4) *By Theorem 3.9, we only need to show the equivalence between C and L. To start with, $C(\mathcal{S} \setminus X) \subseteq \mathcal{S} \setminus X$ for all $X \in \mathcal{P}(\mathcal{S}) \setminus \{\mathcal{S}\}$. Hence, $X \subseteq \mathcal{S} \setminus C(\mathcal{S} \setminus X)$. That is, $X \subseteq L(X)$. Similarly, C enjoys condition (1) which implies that if $X \supseteq Y \supseteq C(X)$ then $C(X) = C(Y)$ (see the proof for Theorem 4.2). As a special case, if $\mathcal{S} \setminus X \supseteq \mathcal{S} \setminus Y \supseteq C(\mathcal{S} \setminus X)$ then $C(\mathcal{S} \setminus X) = C(\mathcal{S} \setminus Y)$. Consequently, if $X \subseteq Y \subseteq \mathcal{S} \setminus C(\mathcal{S} \setminus X)$ then $\mathcal{S} \setminus C(\mathcal{S} \setminus X) = \mathcal{S} \setminus C(\mathcal{S} \setminus Y)$. Therefore, if $X \subseteq Y \subseteq L(X)$ then $L(X) = L(Y)$. Hence, L is a cumulative logic. Yet, L also satisfies (‡) which can be shown from condition (1) below:*

$$\textit{if } C(\overline{X} \cup \overline{Y}) \neq C(\overline{X}) \textit{ and } C(\overline{X} \cup \overline{Y}) \neq C(\overline{Y}) \qquad \textit{then } C(\overline{X} \cup \overline{Y}) = C(\overline{X}) \cup C(\overline{Y})$$

by substituting $L(\mathcal{S} \setminus X)$ for $\mathcal{S} \setminus C(X)$ as follows:

$$\textit{if } \overline{C(\overline{X \cap Y})} \neq \overline{C(\overline{X})} \textit{ and } \overline{C(\overline{X \cap Y})} \neq \overline{C(\overline{Y})} \qquad \textit{then } \overline{C(\overline{X \cap Y})} = \overline{C(\overline{X})} \cap \overline{C(\overline{Y})}$$

As a trivial consequence, if $L(X \cap Y) \neq L(X)$ and $L(X \cap Y) \neq L(Y)$ then $L(X \cap Y) = L(X) \cap L(Y)$. That is, (‡) is verified.
As for the other direction of the equivalence, $\mathcal{S} \setminus X \subseteq L(\mathcal{S} \setminus X)$ for all $X \in \mathcal{P}(\mathcal{S}) \setminus \{\emptyset\}$ because L is cumulative. Hence, $\mathcal{S} \setminus L(\mathcal{S} \setminus X) \subseteq X$. That is, $C(X) \subseteq X$. Also, C satisfies condition (1) which can be shown from (‡) below:

$$\textit{if } L(\overline{X} \cap \overline{Y}) \neq L(\overline{X}) \textit{ and } L(\overline{X} \cap \overline{Y}) \neq L(\overline{Y}) \qquad \textit{then } L(\overline{X} \cap \overline{Y}) = L(\overline{X}) \cap L(\overline{Y})$$

by substituting $C(\mathcal{S} \setminus X)$ *for* $\mathcal{S} \setminus L(X)$ *as follows:*

$$\text{if } \overline{L(\overline{X \cup Y})} \neq \overline{L(\overline{X})} \text{ and } \overline{L(\overline{X \cup Y})} \neq \overline{L(\overline{Y})}$$
$$\text{then } \overline{L(\overline{X \cup Y})} = \overline{L(\overline{X})} \cup \overline{L(\overline{Y})}$$

As a trivial consequence, if $C(X \cup Y) \neq C(X)$ *and* $C(X \cup Y) \neq C(Y)$ *then* $C(X \cup Y) = C(X) \cup C(Y)$*. That is, condition (1) holds.*
In view of Theorem 4.3, the restriction to non-vacuous constraints and non-monotonic cumulative logics also holds because the remaining case is unproblematic: If L is non-monotonic then C *cannot be vacuous.*
That is, there only remains to prove that L being non-monotonic makes C *to be non-vacuous. Since L is non-monotonic, there exist* $S' \subset \mathcal{S}$ *and* $S'' \subset \mathcal{S}$ *such that* $S' \subset S''$ *while* $L(S') \not\subseteq L(S'')$*. Clearly,* $\mathcal{S} \setminus L(S'') \not\subseteq \mathcal{S} \setminus L(S')$*. Hence,* $C(\mathcal{S} \setminus S'') \not\subseteq C(\mathcal{S} \setminus S')$*. As* $\mathcal{S} \setminus S'$ *is* $(\mathcal{S} \setminus S'') \cup (S'' \setminus S')$ *due to* $S' \subset S'' \subset \mathcal{S}$*, we get* $C(\mathcal{S} \setminus S'') \not\subseteq C((\mathcal{S} \setminus S'') \cup (S'' \setminus S'))$*. So,* $C((\mathcal{S} \setminus S'') \cup (S'' \setminus S')) \neq C(\mathcal{S} \setminus S'')$ *and* $C((\mathcal{S} \setminus S'') \cup (S'' \setminus S')) \neq C(\mathcal{S} \setminus S'') \cup C(S'' \setminus S')$*. Then, condition (1) implies* $C((\mathcal{S} \setminus S'') \cup (S'' \setminus S')) = C(S'' \setminus S')$ *to hold. That is,* $C(\mathcal{S} \setminus S') = C(S'' \setminus S')$*.*
Let us assume for the rest of the proof that C *is vacuous:* $C(\mathcal{S}) = \mathcal{S}$*. As* $\mathcal{S}$ *is of course* $S' \cup (\mathcal{S} \setminus S')$*, this entails* $C(S' \cup (\mathcal{S} \setminus S')) = \mathcal{S}$*. By* $C(S') \subseteq S'$ *(while* $S' \neq \mathcal{S}$*) and* $C(\mathcal{S} \setminus S') \subseteq \mathcal{S} \setminus S'$ *(while* $\mathcal{S} \neq \mathcal{S} \setminus S'$*, otherwise* S' *being empty and* C *being vacuous would yield* $C(\mathcal{S} \setminus S') = \mathcal{S}$ *hence* $\mathcal{S} = \mathcal{S} \setminus L(S')$ *by duality so that* $L(S') = \emptyset$ *and* $L(S') \subseteq L(S'')$ *would hold), all this leads to* $C(S' \cup (\mathcal{S} \setminus S')) \neq C(S')$ *and* $C(S' \cup (\mathcal{S} \setminus S')) \neq C(S \setminus S')$*. For condition (1) not to be contradicted,* $C(S' \cup (\mathcal{S} \setminus S')) = C(S') \cup C(\mathcal{S} \setminus S')$ *must then hold. That is,* $C(\mathcal{S}) = C(S') \cup C(\mathcal{S} \setminus S')$*. Due to* $C(\mathcal{S} \setminus S') = C(S'' \setminus S')$ *as was previously obtained, it then follows that* $C(\mathcal{S}) = C(S') \cup C(S'' \setminus S')$*. Therefore,* $C(\mathcal{S}) \subseteq S' \cup (S'' \setminus S')$*. Consequently,* $C(\mathcal{S}) \subseteq S''$ *and* $C(\mathcal{S}) \neq \mathcal{S}$ *is proved so that a contradiction arises.*