Optimality Theory through Default Logic

Ph. Besnard
Institut für Informatik
Universität Potsdam
Postfach 900327
D-14439 Potsdam
Germany
torsten@cs.uni-potsdam.de

R. E. Mercer
Computer Science Department
Middlesex College
University of Western Ontario
London Ont. N6A 5B7
Canada
mercer@csd.uwo.ca

T. Schaub
Institut für Informatik
Universität Potsdam
Postfach 900327
D-14439 Potsdam
Germany
torsten@cs.uni-potsdam.de

Abstract

Optimality Theory is an approach to linguistic problems which is based on rules with exceptions, resorting to a ranking among rules to resolve conflicts arising from competing rules.

In such a way, dealing with linguistic problems amounts to applying rules with exceptions: That is reasoning. A related issue is then about a formalization of the logic at work. An immediate candidate is Default Logic which is dedicated to reasoning from rules with exceptions. Moreover, there are versions of default logic with priorities.

We show that Default Logic is well-suited as a specification language capturing Optimality Theory and suggests that implementations of default logic can be applied to run experiments with grammatical interaction in the sense of Optimality Theory and beyond.

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 (or, say, representations), but it does not care how these candidate structures are generated.

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 grammatical 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 simple and 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.

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.

In Optimality Theory, the grammatical constraints are organized in a hierarchy. When 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 are considered in decreasing order).

Here is an example. The highest constraint is "the question word occurs first", the next highest constraint is "the verb group comes second", and the lowest constraint is "any non-subject item occurs after the subject". 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) that is 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 of the competition).

In Optimality Theory, this is often visualized in a two-dimensional table as follows. Rewriting the 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 depicted in Table 1:

Table 1: A 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 1 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).

Constraints in Optimality Theory turn out to be rules with exceptions: They are universal but they are *not* universally valid (e.g., there are syllables in English that have no onset). Indeed, Optimality Theory provides a methodology to apply rules with exceptions. As Default Logic was motivated by the need to deal with such rules, it seems natural to investigate whether Optimality Theory conforms with Default Logic.

Actually, Optimality Theory appears as reasoning non-monotonically not only about optimal candidates but also about candidates that fail to be optimal. Indeed, applying constraints in Optimality Theory leads to identify candidates as being suboptimal.

Unexpectedly enough, focusing on suboptimal candidates turns out to yield a direct characterization using defaults with priorities. This is interesting because the intuitive approach in Optimality Theory is more about the so-called winners than the so-called losers. Yet, losers and winners can easily be related in the obvious way using Default Logic.

2 Interaction in a linear hierarchy of constraints

Consider the case where the candidates involve no calculation: The list of candidates in final form $a, b, \ldots$ is available and the status (whether[1] absolute or relative to any other candidate) of each candidate with respect to every constraint is decided.

2.1 Harmonic parallel approach

Logic relations and defaults

The status of candidates with respect to constraints is to be encoded by means of $c_i defeats(a,b)$ relations over candidates and suboptimality (i.e., failure for a candidate to be a correct output) is encoded as $suboptimal(a)$.

The relation $c_i defeats(a,b)$ captures the case of non-binary constraints as well as the case of binary constraints (cf page 69 onwards in [Prince & Smolensky 1993]).

That suboptimality is determined from the status of the candidates is rendered through defaults:

$$(c_i) \quad \frac{c_i defeats(a,b) \ : \ \neg suboptimal(a)}{suboptimal(b)}$$

that should be read (after [Prince & Smolensky 1993] on page 74) as follows:

if b is less harmonic than a with respect to the constraint c_i then b is suboptimal unless a is itself suboptimal

While the set of all candidates is simply assumed to be finite, the set of constraints is assumed to be both finite and totally ordered: $C = \{c_1, c_2, \ldots\}$ where c_1 ranks highest then c_2 and so on. So, the above defaults for Optimality Theory are ordered accordingly:

$$c_i < c_j \quad \text{iff} \quad j \geq i \text{ and } i \neq j$$

Hence, let us consider default logic with priorities (adapted from [Brewka 1994]):

[1] Depending on each particular constraint.

Definition 1 *Let $(W, D, <)$ be a default theory where $<$ is a total order over D. Define*

- $E_0 = W$
- *for* $n \geq 0$,

$$E_{n+1} = \mathrm{Th}(E_n \cup \{cons(\delta)\})$$

where δ ranks highest among all defaults $\frac{\alpha:\beta}{\gamma}$ in D that satisfy the conditions[2] $\alpha \in E_n$ *and* $\neg\beta \notin E_n$ *while* $\gamma \notin \mathrm{Th}(E_n)$*; if no such default exist,* $\{cons(\delta)\} = \emptyset$.

Then, $E = \cup_{n \geq 0} E_n$ *is the extension of* $(W, D, <)$.

A correct output is then any candidate for which there exists an extension in which it is *not* proven suboptimal. [3]

Example 1 *The language is a dialect of Berber and the item to be parsed is /txznt/. Two constraints (the one denoted c_1 ranks higher than the one denoted c_2) are examined:*

(c_1) Ons *(Syllables must have onsets)*

(c_2) Hnuc *(High sonority nucleus is more harmonic)*

For this example, [Prince & Smolensky 1993] consider the following three candidates:

.txź.ńt.
.tx́.zńt.
.t́x.zńt.

With a and b thus ranging over the set of candidates {.txź.ńt., .tx́.zńt., .t́x.zńt.}, the defaults are:

$$(c_1) \quad \frac{c_1 defeats(a,b) \ : \ \neg suboptimal(a)}{suboptimal(b)}$$

(requiring $a \neq b$ yields six such defaults)[4]

and

$$(c_2) \quad \frac{c_2 defeats(a,b) \ : \ \neg suboptimal(a)}{suboptimal(b)}$$

(requiring $a \neq b$ yields six such defaults)[4]

Taking .txź.ńt. to be less behaved than the other two candidates regarding Ons,

$$c_1 defeats(.tx́.zńt., .txź.ńt.) \in W,$$

$$c_1 defeats(.t́x.zńt., .txź.ńt.) \in W,$$

and taking .t́x.zńt. to be less behaved than .tx́.zńt. regarding Hnuc,

$$c_2 defeats(.tx́.zńt., .t́x.zńt.) \in W,$$

the resulting default theory has exactly one extension:

$$\mathrm{Th}(W \cup \{suboptimal(.txź.ńt.), suboptimal(.t́x.zńt.)\})$$

The only candidate which is not proven suboptimal in the extension is .tx́.zńt., this is the unique optimal candidate (the correct output).

The outcome would be the same, should $c_1 defeats(.t́x.zńt., .txź.ńt.)$ not be in W. This is an instance of the property which ensures that those $c_i defeats(a, \cdot)$ items are enough to get the correct output(s), where a ranges over the optimal candidate(s) while c_i ranges over all constraints such that $i > j$ where j is the highest ranking constraint violated by the optimal candidate(s).

It would also make no difference if Hnuc *had .t́x.zńt. less behaved than .txź.ńt.*

Although the underlying idea is the same, using $c_i defeats$ relations is more general than the mark approach in [Prince & Smolensky 1993] (cf page 68 onwards) which can be captured (as $c_i defeats$ can be defined in virtually any way in W) as follows.

$$\forall a, b, m_a, m_b \quad Ons(a, m_a) \wedge Ons(b, m_b) \wedge m_a \succ_{Ons} m_b \rightarrow c_1 defeats(a, b)$$

$$\forall a, b, m_a, m_b \quad FM(m_a) \succ_{Ons} FM(m_b) \rightarrow m_a \succ_{Ons} m_b$$

$$\forall a, b, m_a, m_b \quad (FM(m_a) \approx_{Ons} FM(m_b) \wedge Rest(m_a) \succ_{Ons} Rest(m_b)) \rightarrow m_a \succ_{Ons} m_b$$

$Ons(c, m_c)$ means that Ons assigns a list m_c of violation marks to every candidate c. Then, $FM(m_a) \succ_{Ons}$

[2] It is alos possible to require for δ to be such that $\neg jus(\delta') \notin \mathrm{Th}(E_n \cup \{cons(\delta)\})$ for all defaults δ' selected prior to δ. (i.e., for some $m < n$, δ' ranking highest such that $\alpha' \in E_m$ and $\neg\beta' \notin E_m$ while $\gamma' \notin \mathrm{Th}(E_m)$).

[3] The reader should not worry about matters of uniqueness and non-uniqueness, which are unproblematic even though the existence of exactly one extension for $(W, D, <)$ does not preclude the existence of more than one optimal output. By contrast, it is an absolute requirement that the order of defaults be total and this means that *instances* of the *same* constraint c_i have to be ordered (or $c_i defeats(a, b)$ must be antisymmetric, as well as irreflexive of course).

[4] An arbitrary total order is assumed among these six defaults (cf the previous footnote).

$FM(m_b)$ always holds when m_a is empty while m_b is not. Also, $FM(m_a) \approx_{Ons} FM(m_b)$ always holds when m_a and m_b are both non-empty. Lastly, $Rest(m_c)$ denotes the list m_c deprived from its first mark.

$$\forall a, b, m_a, m_b \quad Hnuc(a, m_a) \wedge Hnuc(b, m_b) \wedge m_a \succ_{Hnuc} m_b \rightarrow c_2 defeats(a, b)$$

$$\forall a, b, m_a, m_b \quad FM(m_a) \succ_{Hnuc} FM(m_b) \rightarrow m_a \succ_{Hnuc} m_b$$

$$\forall a, b, m_a, m_b \quad (FM(m_a) \approx_{Hnuc} FM(m_b) \wedge Rest(m_a) \succ_{Hnuc} Rest(m_b)) \rightarrow m_a \succ_{Hnuc} m_b$$

Similarly to ONS, this assumes HNUC ([Prince & Smolensky 1993] page 72) to provide:

- a list of marks m_c sorted from most to least sonorous[5] for every candidate c (as is required in the first formula)
- an assessment for every pair of marks where one is more sonorous than the other (as is required in the second formula)
- an assessment for every pair of equally sonorous marks (as is required in the third formula)

All this assumes further that W contains the usual axioms for equality (for instance, $Rest(Rest(Rest(m_c)))$ must be provably equal with the list obtained by deleting the first three marks in m_c). There are other formulations, getting rid of FM and $Rest$, such that the axioms of equality can be dispensed with.

As to a different kind of an example, consider NONFINALITY in a version where it is a binary constraint that does not apply multiply ([Prince & Smolensky 1993] page 43)

$$\forall a, b \quad Nonfinality(a) \wedge \neg Nonfinality(b) \rightarrow c_i defeats(a, b)$$ [6]

where i is the rank of NONFINALITY and for every candidate c, $Nonfinality(c)$ holds when the head foot of the prosodic word is not final.

2.2 Harmonic sequential approach

Now, it is no longer the case that the candidates are available right from the start. They are to form step by step, some of them getting discarded even before they develop to final form (this is the difference with the parallel approach).

Logic relations and defaults

Generation of (partial) candidates is encoded by means of the $gen(a, b)$ relation indicative of a derivation step from a to b. That a representation in the derivation currently counts as a candidate is encoded by means of $current(a)$. Steps in generation are rendered through the default[7]

$$(Gen) \quad \frac{current(a) \wedge gen(a, b) \; : \; \neg suboptimal(b)}{current(b)}$$

that reads

if a counts as a candidate and there is a step in derivation turning a into b then b counts as a candidate unless b is suboptimal

Importantly, this default[8] ranks lower than all the ones representing constraints:

$$(c_i) \quad \frac{c_i defeats(a, b) \; : \; \neg suboptimal(a)}{suboptimal(b)}$$

These defaults are exactly as in the parallel approach[9] (thus making the sequential approach a generalization of it).

$$\forall x \;\; input(x) \rightarrow current(x) \in W$$

If that is desired, a more detailed formulation is possible where changed elements are explicited as it only takes including in W the following formulas for all relevant a, b, c:

$$change(a, e_a)$$

$$(gen(a, b) \wedge gen(a, c) \wedge change(b, e_b) \wedge change(c, e_c) \wedge c_i defeats(e_b, e_c)) \rightarrow c_i defeats(b, c)$$

The former formulas indicate that e_a is the changed element in a and the latter formulas state that defeat among candidates is ruled by defeat among changed elements.

---

[5] Harmonic, if we identify the mark with the nucleus it stands for (p. 72 [Prince & Smolensky 1993]).

[6] This formula makes $c_i defeats(a, b)$ to be antisymmetric. Also, if all candidates fail NONFINALITY then $c_i defeats(a, b)$ holds for no a and b.

[7] Singular is used here, although improperly, as the specific values of a and b are unimportant.

[8] Again, singular is improperly used. As a further motive, it can be pointed out that the highest ranking among all defaults of that form ranks lower than any default representing a constraint.

[9] Even though they need *not* obey the same order: see footnote 49 in [Prince & Smolensky 1993].

Digression. A subtlety is that antisymmetry is required for $c_i defeats$ if its arguments are changed elements (i.e., $c_i defeats(e_b, e_c)$ and $c_i defeats(e_c, e_b)$ are incompatible). Such a requirement is *not* needed for $c_i defeats$ if its arguments are partial candidates (that is, of concern here is symmetry through $c_i defeats(b, c)$ and $c_i defeats(c, b)$). How can this be? Well, b and c differ from each other by at least one changed element. Now, one of these must occur before the other and therefore *rules out* the other candidate.

Example 2 *In the same dialect of Berber, the item to be parsed is /ratlult/ so that*

$$input(ratlult) \in W$$

The following formulas are also in W:

$$gen(ratlult, \{rA\}tlult)$$
$$gen(\{rA\}tlult, \{rA\}t\{lu\}lt)$$
$$gen(\{rA\}tlult, \{rAt\}lult)$$

Distinctively from the parallel approach (cf [Prince & Smolensky 1993]), the ordering of constraints[10] *is:*

(c_1) $-$COD *(Syllables do not have codas)*

(c_2) HNUC *(Higher sonority nucleus is more harmonic)*

such that

$$c_2 defeats(\{rAt\}lult, \{rA\}t\{lu\}lt) \in W$$

$$c_1 defeats(\{rA\}t\{lu\}lt, \{rAt\}lult) \in W$$

The resulting default theory has exactly one extension:

$$\mathrm{Th}(W \cup X)$$

where X consists of the following formulas

$$current(\{rAt\}lult)$$
$$suboptimal(\{rAt\}lult)$$
$$current(\{rA\}t\{lu\}lt)$$

The above data are partial, they only describe an initial part of the process of selecting the correct output. Yet, they suffice to indicate that $\{rA\}t\{lu\}lt$ is the line to be developed but not $\{rAt\}lult$. Should more data be taken into account, the result goes further.

If the changed elements are to be explicited, the following formulas have to be in W:

$$gen(ratlult, \{rA\}tlult) \rightarrow change(\{rA\}tlult, \{rA\})$$

$$gen(\{rA\}tlult, \{rA\}t\{lu\}lt) \rightarrow change(\{rA\}t\{lu\}lt, \{lu\})$$

$$gen(\{rA\}tlult, \{rAt\}lult) \rightarrow change(\{rAt\}lult, \{rAt\})$$

$$c_2 defeats(\{rAt\}, \{lu\})$$

$$c_1 defeats(\{lu\}, \{rAt\})$$

$$c_2 defeats(\{rAt\}, \{lu\}) \rightarrow c_2 defeats(\{rAt\}lult, \{rA\}t\{lu\}lt)$$

$$c_1 defeats(\{lu\}, \{rAt\}) \rightarrow c_1 defeats(\{rA\}t\{lu\}lt, \{rAt\}lult)$$

Of course, the outcome is exactly the same.

3 Interaction in an arbitrary hierarchy of constraints

3.1 Harmonic parallel approach

Footnote 31 in [Prince & Smolensky 1993] acknowledges the possibility that a grammar should recognize nonranking pairs of constraints (although the authors make it clear that they found no evidence of crucial nonranking). Non-linear hierarchies of constraints may further seem invited from the view stated on page 88 in [Prince & Smolensky 1993], that a category of constraints dominated by others fixed in superordinate position may be relatively ranked in any dominance order in a particular language.[11]

Logic relations and defaults

Logic relations and defaults are still as described above. The set of constraints is still finite but need no longer to be totally ordered: Therefore, it is still the case that $c_i < c_j$ for some c_i and c_j in $C = \{c_1, c_2, \ldots\}$ but this no longer relates to $j \geq i$. Accordingly, let us now consider default logic with non-linear priorities (adapted from [Baader & Hollunder 1995]):

Definition 2 *Let $(W, D, <)$ be a default theory where $<$ is a partial order over D. Then, E is an extension for $(W, D, <)$ iff $E = \cup_{n \geq 0} E_n$ such that*

- $E_0 = W$

[10] Where $-$COD is introduced on page 34 in [Prince & Smolensky 1993].

[11] Not: ... are relatively ranked in some dominance order

- *for $n \geq 0$,*

$$E_{n+1} = \mathrm{Th}(E_n \cup Cons(\{\delta_1, \ldots, \delta_k\}))$$

where $\delta_1 \ldots \delta_k$ have priority[12] *among all defaults in D that are active*[13] *in E_n*

An example [Prince & Smolensky 1993] not requiring a linear hierarchy of constraints:

Example 3 *The language is Latin. Three constraints are considered:*

(c_1) FtBin *(Feet are binary at some level of analysis)*

(c_2) Lx≈Pr *(A member of a certain morphological category corresponds to a* PrWd*)*

(c_3) Nonfinality *(The head foot of the prosodic word is not final)*

Consider the following three candidates:

.a.qua.
(á) L
(á.qua)

With u and v ranging over the set of candidates {.a.qua., (á) L, (á.qua)}, defaults are:

$$(c_1) \quad \frac{c_1 defeats(u,v) \ : \ \neg suboptimal(u)}{suboptimal(v)}$$

$$(c_2) \quad \frac{c_2 defeats(u,v) \ : \ \neg suboptimal(u)}{suboptimal(v)}$$

$$(c_3) \quad \frac{c_3 defeats(u,v) \ : \ \neg suboptimal(u)}{suboptimal(v)}$$

c_3 ranks lowest of all but neither c_1 ranks higher than c_2 nor c_2 ranks higher than c_1. The other data are

$$\forall u \; c_1 defeats(u, (\acute{a})\, L) \in W$$

$$\forall u \; c_2 defeats(u, .a.qua.) \in W$$

$$\forall u \; c_3 defeats(u, (\acute{a}.qua)) \in W$$

The outcome is that the default theory at hand has a single extension

$$E = \mathrm{Th}(W \cup \{suboptimal(\acute{a})\, L), suboptimal(.a.qua.)\})$$

The optimal candidate is (á.qua), there was no need for a linear ordering of constraints.

The existence of a unique optimal candidate coincides with the existence of a greatest element in the induced[14] ordering.

[12]The precise definition depends on what variant of default logic is selected.

[13]Again, there is a choice of definitions here and $Cons(\{\delta_1, \ldots, \delta_k\}) = \emptyset$ by convention when no default is active.

[14]Induced by c_i's ordering.

4 Logic programming

As for harmonic parallelism, the case of a linear hierarchy of constraints is efficiently tackled using logic programming, in a guise e.g. [Delgrande, Schaub, & Tompits 2000] where preferences can be expressed so that preemption among constraints is resolved:

```
suboptimal(b) :-
              name(i(a,b)),
              c-i-defeats(a,b),
              not suboptimal(a).
```

For all $j < k$ (i.e., constraint j ranks higher than constraint k), one need the clauses:

```
(j(_,_) < k(_,_)).
```

Example 4 *(Back to the Berber example.) The predicate* `missing-ons(c,r)` *assigns the missing onsets in the candidate c to the variable r whereas* `s-nuclei(c,r)` *assigns the nuclei in the candidate c (sorted from most to least sonorous) to the variable r.*

```
c-1-defeats(a,b) :-
                 missing-ons(a,p),
                 missing-ons(b,q),
                 better-ons(p,q).
c-2-defeats(a,b) :-
                 s-nuclei(a,p),
                 s-nuclei(b,q),
                 better-hnuc(p,q).
better-ons([],[z|t]).
better-ons([x|r],[y|s]) :-
                        better-ons(r,s).
better-hnuc([x|r],[y|s]) :-
                         more-sonorous(x,y).
better-hnuc([x|r],[x|s]) :-
                         better-hnuc(r,s).
```

Entering the data,

```
missing-ons(".txZ.Nt.",[2nd-syll]).
missing-ons(".tX.zNt.",[]).
missing-ons(".Tx.zNt.",[]).
s-nuclei(".txZ.Nt.",["N","Z"]).
s-nuclei(".tX.zNt.",["N","X"]).
s-nuclei(".Tx.zNt.",["N","T"]).
more-sonorous("X","T").
```

the outcome is that the goal `suboptimal(".txZ.Nt.")` *succeeds as does the goal* `suboptimal(".Tx.zNt.")` *while the goal* `suboptimal(".tX.zNt.")` *fails, meaning that* `".tX.zNt."` *is optimal.*

5 Summary of method

The method introduced here can be described as follows:

1. *Include in W the status of candidates with respect to constraints:*

$$C_\lambda \rightarrow c_i defeats(\varphi, \sigma)$$
$$\vdots$$

where the conditions C_λ are basically tautological in the parallel approach[15]
—in which case the above formulas then simplify to $c_i defeats(\varphi, \sigma)$.

2. *Include in D the effect of constraints towards suboptimality:*

$$\frac{c_i defeats(\varphi, \sigma) \ : \ \neg suboptimal(\varphi)}{suboptimal(\sigma)}$$

3. *Specify the ordering $<$ between constraints.*

4. *In view of the resulting default theory $(W, D, <)$, look for any candidate c such that $suboptimal(c)$ is not in an extension. Such a candidate is a correct output.*

As we have seen, the above scheme can be extended when dealing with special cases such as the sequential approach and so on:

$$gen(\chi, \kappa)$$
$$\vdots$$
$$change(\kappa, \epsilon_\kappa)$$
$$\vdots$$
$$gen(\varphi, \sigma) \wedge gen(\varphi, \rho) \wedge change(\sigma, \epsilon_\sigma) \wedge change(\rho, \epsilon_\rho) \wedge c_i defeats(\epsilon_\sigma, \epsilon_\rho) \rightarrow c_i defeats(\sigma, \rho)$$
$$\vdots$$

Expressing Optimality Theory in default logic without priority is also possible, with defaults that are no more complex than above. However, modularity as well as independence of defaults from constraint ranking are lost.

Acknowledgements

This research has been partially supported by the German Science Foundation (DFG) under grant FOR 375/1-1, TP C.

[15] Reduplication of the mark approach is a good example showing that the conditions C_λ can take various forms, all of them amount to tautologies.

References

[Baader & Hollunder 1995]
F. Baader and B. Hollunder.
Priorities on defaults with prerequisites. *Journal of Automated Reasoning* 15:41–68, 1995.

[Brewka 1994] G. Brewka.
Adding priorities and specificity to default logic. In L. Pereira and D. Pearce, editors, *4th European Workshop on Logics in Artificial Intelligence (JELIA-94)*, pp. 247–260, 1994.

[Delgrande, Schaub, & Tompits 2000]
J. Delgrande, T. Schaub, and H. Tompits.
A compiler for logic programs with preferences. 2000.
http://www.cs.uni-potsdam.de/~torsten/plp

[Delgrande, Schaub, & Tompits 2000]
J. Delgrande, T. Schaub, and H. Tompits.
A compiler for ordered logic programs. *8th Workshop on Non-Monotonic Reasoning (NMR-2000)*, Breckenridge, CO, 2000.

[Delgrande, Schaub, & Tompits 2000]
J. Delgrande, T. Schaub, and H. Tompits.
Logic programs with compiled preferences. *14th European Conference on Artificial Intelligence (ECAI-00)*, Berlin, Germany, 2000.

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

[Prince & Smolensky 1993]
A. Prince and P. Smolensky.
Optimality Theory. Tech Report, Rutgers U., New Brunswick (NJ), and Computer Science Department, U. of Colorado, Boulder. 1993.