More than the sum of its parts: compact preference representation over combinatorial domains
Uckelman, J.D.

Citation for published version (APA):
Uckelman, J. D. (2009). More than the sum of its parts: compact preference representation over combinatorial domains Amsterdam: Institute for Logic, Language and Computation
In this dissertation we have presented a framework for compactly representing cardinal preferences over combinatorial domains and shown the feasibility of using this framework for auctions and voting.

Goalbase languages are formed by the parameters restricting the available formulas and weights. Due to their parametric nature, these languages are scattered all across the representational landscape. In order to make practical use of goalbases, we must first know the lay of the land. In Part I, we have explored the landscape of goalbase languages in three directions:

**Expressivity.** In Chapter 3, we considered this question: Given a goalbase language, what utility functions are representable in it? Many goalbase languages with natural definitions were revealed to correspond exactly to classes of utility functions having well-known properties. Furthermore, we showed that some goalbase languages have precisely one representation for any representable utility function, and provided methods for finding these representations. For a summary of these results, see Figures 3.1 and 3.2, and the accompanying explanatory text in Sections 3.4.4 and 3.5.4.

**Succinctness.** In Chapter 4, we pursued the problem of concision. Given two goalbase languages, are the smallest representations in one significantly smaller than equivalent smallest representations in the other? We systematically compared more than two hundred pairs of languages to determine which languages were more succinct. For a summary of these results, see Tables 4.1 and 4.2.

**Complexity.** In Chapter 5, we examined how the structure of a goalbase language affects the computational complexity of three decision problems: MAX-UTIL, MIN-UTIL, and MAX-CUF. More expressive languages tended to have NP-complete MAX-UTIL and MAX-CUF problems, and coNP-complete MIN-UTIL problems; for those which are solvable in polynomial time, we provided algorithms demonstrating
that. For a summary of the results for MAX-UTIL and MIN-UTIL, see Table 5.1, and for MAX-CUF see Table 5.2. Finally, we considered an alternative version of MAX-UTIL, which focuses on true atoms in optimal states instead of the existence of models which reach a given utility level.

In Part II of this dissertation, we considered two possible applications of goalbase languages:

**Auctions.** In Chapter 6, we introduced combinatorial auctions and the widely-studied XOR/OR family of bidding languages. Goalbase languages are sometimes more and sometimes less succinct than languages in the XOR/OR family, depending on the utility functions to be represented. This is as expected, and supplies some of our motivation for the investigations in Part I: An auction designer cannot choose the most appropriate bidding language without knowing the expressivity and succinctness characteristics of the languages on offer. We went on to describe the Winner Determination Problem, both formulating it as an integer program and defining branch-and-bound heuristics for solving it directly. Finally, we presented some experimental results showing the performance of our IP formulation and branch-and-bound solver, which demonstrated the feasibility of using goalbase languages for auctions of moderate size.

**Voting.** In Chapter 7, we considered the problem of insufficiently expressive voting methods, and suggested voting with goalbases as ballots as a possible remedy. Common single-winner voting methods do not extend well to multi-winner settings like committee elections due to interactions between candidates. We noted that finding winners when goalbases are used as ballots is similar to MAX-UTIL, and that in practice the complexity will tend to be manageable due to small numbers of candidates and seats. We suggested an extension to approval voting, where properties of the outcome are approved (or not) rather than particular outcomes as in standard approval voting. Finally, we discussed a number of directions for further investigation.

Together, these chapters provide a clear view of the power of goalbase languages for preference representation, and point to potential areas of application.

**Open Questions and Future Work.** Here we mention some open questions and directions for future work:

In Chapter 2, we have presented only a limited range of restrictions on formulas. It would be interesting to investigate other, less straightforward properties of formulas, such as being read-once, or representable by a Boolean circuit with certain features, to see what effect these might have on expressivity, succinctness, and complexity. Additionally, sum and max are only the most obvious aggregators; in principle, any function $F: \mathbb{N}^R \rightarrow \mathbb{R}$ could be used, so there are others (e.g., min) which might be of interest.
In Chapter 3, there were a few languages for which we were unable to ex-
actly characterize their expressivity. (See Figures 3.1 and 3.2.) In particular,
$L(k\text{-forms}, \mathbb{R}^+, \Sigma)$ represents some subset of the nonnegative $k$-additive utility
functions, but which subset or even whether the inclusion is proper is unknown.
For $L(state, \mathbb{R}^+, \Sigma)$, it is known to represent a proper subset of the nonnegative
utility functions, but here, again, exactly what other properties that subset has is
unknown.

In Chapter 4, some open succinctness questions can be seen in Table 4.1.
We do not expect any of the proofs which would resolve the remaining open
questions to be easy, as all involve pairs of languages where neither language has
unique representations. We would particularly like to know the relation between
$L(cubes, \mathbb{R}^+, \Sigma)$ and $L(state, \mathbb{R}^+, \Sigma)$, as well as between $L(state, \mathbb{R}^+, \Sigma)$ and
these two; but resolving any of the remaining open questions would be useful,
since the combination of Fact 4.2.3 and the numerous results we already have
mean that any new result can be leveraged to answer several open questions at
once.

In Chapter 5, we resolved for most languages whether $\text{max-util}$ and $\text{max-cuf}$
are polynomial or NP-complete. What this leaves open is where the boundary is:
For the languages which have polynomial decision problems, how much can we
loosen the restrictions on their structure and still remain polynomial? Or, from
the other direction: How little can we take away from the NP-complete languages
before they become polynomial? We have also not investigated how amenable
these decision problems are to approximation, nor whether for $\text{max-cuf}$ any of
the languages are strategyproof and if not, how hard manipulation is.

We believe that our WDP algorithms in Chapter 6 could be improved, either
through tighter heuristics or other combinatorial optimization techniques. In a
practical vein, it would be gratifying to see one of our languages used for real
combinatorial auctions.

The avenues for further work on voting in Chapter 7 are too numerous to
repeat here; for a full accounting of them, see Section 7.7. Of technical interest are
methods for finding a maximally sincere ballot when no ballot matches a voter’s
preferences and determining whether a voting rule advantages or disadvantages
voters with certain preferences. Of practical interest is how to design a goalbase
voting system which human voters would find usable.