Computational Social Choice: Spring 2007

Ulle Endriss
Institute for Logic, Language and Computation
University of Amsterdam

Bidding Languages for Combinatorial Auctions

- Bidding languages are languages for *representing valuations* over bundles of goods. They are called *bidding* languages, because they have been developed to (succinctly) encode bids in *combinatorial auctions*.

- Bidding languages are a particular class of languages for modelling preferences in combinatorial auctions. Their distinguishing feature is that they have been developed in the context of CAs, rather than than anything special about the languages themselves.

- We'll be discussing issues pertaining to *expressive power* and *comparative succinctness* issues, similarly to what we have done a few weeks ago.

Plan for Today

We are going to review several bidding languages and review their expressive power and succinctness:

- Basic OR and XOR languages
- Combinations of OR and XOR
- OR* with dummy items

This lecture will largely follow the (highly recommended!) review article by Nisan (2006), and many of the results are originally due to Nisan (2000).

N. Nisan. Bidding and Allocation in Combinatorial Auctions. Proc. EC-2000.

N. Nisan. Bidding Languages for Combinatorial Auctions. In P. Cramton *et al.* (eds.), *Combinatorial Auctions*, MIT Press, 2006.

Assumptions

Let $\mathcal{R}$ be a finite set of goods (resources).

We use bidding languages to encode valuations $v : 2^{\mathcal{R}} \rightarrow \mathbb{R}$.

Throughout this lecture, we shall assume that valuations are both *normalised* and *monotonic*.

- v is *normalised* iff $v(\{\,\}) = 0$
- v is *monotonic* iff $v(X) \leq v(Y)$ whenever $X \subseteq Y$

Observe that this entails that valuations are *non-negative*.

Atomic Bids

An *atomic bid* is a pair (B, p) where $X \subseteq \mathcal{R}$ is a bundle of goods and $p \in \mathbb{R}^+$ is a price. Intuitively, this means that the agent is prepared to pay p in return for receiving B.

Formally, the atomic bid (B, p) defines the valuation v:

$$v(X) \quad = \quad \begin{cases} p & \text{if } X \supseteq B \\ 0 & \text{otherwise} \end{cases}$$

Note how the assumption that all valuations are *monotonic* enters this definition (otherwise use $=$ in place of $\supseteq$).

Atomic bids alone cannot express very interesting valuations.

▶ How can we *combine* several atomic bids?

The OR Language

There are various options of how to combine several atomic bids to model a valuation function ... including the *OR language:*

The auctioneer may accept any combination of non-conflicting bids (bundles *don't overlap*) and charge the sum of the associated prices.

Formally, an OR-combination of two bids defining valuations v_1 and v_2 defines the following valuation:

$$(v_1 \text{ OR } v_2)(X) \quad = \quad \max_{X_1 \subseteq X} (v_1(X_1) + v_2(X \setminus X_1))$$

If there are k atomic bids defining valuations $v_1, \ldots, v_k$, then the overall bid defines the valuation $v_1 \text{ OR } (v_2 \text{ OR } \cdots (v_{k-1} \text{ OR } v_k)))$.

This is the standard bidding language. If an author doesn't say what language they are using, it's probably this one.

OR: Expressive Power

The OR language is not fully expressive. However:

Proposition 1 *The OR language can represent all supermodular valuations, and only those.*

Proof: Easy. ✓

Recall that a valuation v is supermodular iff we have
$v(X \cup Y) \geq v(X) + v(Y) - v(X \cap Y)$ for all $X, Y \subseteq \mathcal{R}$.

The XOR Language

Another possible interpretation of a set of atomic bids by the same bidder would be that the auctioneer can accept *at most one* of these bids. This is called the *XOR language*.

Formally, an XOR-combination of valuations v_1 and v_2 has got the following semantics:

$$(v_1 \text{ XOR } v_2)(X) \quad = \quad \max\{v_1(X), v_2(X)\}$$

If there are k bids defining valuations $v_1, \ldots, v_k$, then the overall bid defines the valuation $v_1 \text{ XOR } (v_2 \text{ XOR } \cdots (v_{k-1} \text{ XOR } v_k)))$.

XOR: Expressive Power

The XOR language is fully expressive:

Proposition 2 *The XOR language can represent all valuations.*

Proof: Easy. ✓

But keep in mind that this only applies to *monotonic* valuations. Valuations that are not monotonic cannot be expressed, unless we change the definition of the semantics of atomic bids.

For instance, if you submit the bid $(\{a\}, 7)$ XOR $(\{a, b\}, 5)$, then the auctioneer can allocate either $\{a\}$ or $\{a, b\}$ to you, and charge a price of 7 either way.

Comparative Succinctness

- The *size* of a bid is the number of atomic bids in it.
- Additive valuations require linear size in the OR language, but may require exponential size in the XOR language (why?).
- Hence, while the XOR language is more expressive than the OR language, it is not more *succinct* (indeed, it will be significantly less succinct for many natural valuations).

Combinations of OR and XOR

So far, we have assumed that each bidder submits a set of atomic bids and that the operator to be applied (OR or XOR) is implicit. If we write out operators explicitly, we can also allow arbitrary combinations of OR and XOR. Example:

$$(\{a,b\},6) \text{ OR } ((\{c\},4) \text{ XOR } (\{b,c\},8))$$

To interpret this, recall the semantics of the operators:

$$\begin{aligned} (v_1 \text{ OR } v_2)(X) &= \max_{X_1 \subseteq X}(v_1(X_1) + v_2(X \setminus X_1)) \\ (v_1 \text{ XOR } v_2)(X) &= \max\{v_1(X), v_2(X)\} \end{aligned}$$

... and of atomic bids (B,p):

$$v(X) = \begin{cases} p & \text{if } X \supseteq B \\ 0 & \text{otherwise} \end{cases}$$

Languages

Here are some obvious candidates for languages to consider:

- OR-of-XOR: OR-comb. of XOR-combinations of atomic bids
- XOR-of-OR: XOR-comb. of OR-combinations of atomic bids
- OR/XOR: arbitrary combinations of OR and XOR (most general language consider so far, subsuming all others)

How do they relate in terms of expressive power and succinctness?

Downward Sloping Valuations

Let us define a special valuation needed for the next result ...

A valuation $v : 2^{\mathcal{R}} \to \mathbb{R}$ is called *symmetric* iff there exists a function $v' : \mathbb{N}_0 \to \mathbb{R}$ such that $v(X) = v'(|X|)$ for all $X \subseteq \mathcal{R}$.

That is, for a symmetric valuation only the *number of goods* matters (rather than *which* goods you get).

Call a symmetric valuation v *downward sloping* iff $v'(k) - v'(k-1) \;\geq\; v'(k+1) - v'(k)$ for all $k \in \mathbb{N}$.

That is, a downward sloping valuation is both symmetric and concave: the marginal benefit of obtaining additional items gets smaller and smaller as we get more items to begin with.

OR-of-XOR and Downward Sloping Valuations

Proposition 3 (Nisan, 2000) *The OR-of-XOR language can represent any downward sloping valuation over n goods in size n^2.*

Proof: Let $x_1, \ldots, x_n$ be the goods; and let $p_k = v'(k) - v'(k-1)$ for $k \leq n$ be the price of the kth good (set $v'(0) = 0$).

This OR-of-XOR bid does the job:

$$\begin{aligned} & \textit{any-one-for}(p_1) \text{ OR } \cdots \text{ OR } \textit{any-one-for}(p_n), \text{ where} \\ \textit{any-one-for}(p_k) \; &= \; (\{x_1\}, p_k) \text{ XOR } \cdots \text{ XOR } (\{x_n\}, p_k) \end{aligned}$$

This formula has length n^2. ✓

The same is *not possible* using OR or XOR bids alone (why?).

▶ Hence, the OR-of-XOR language strictly *more succinct* than either the OR or the XOR language.

Monochromatic Valuations

Another special valuation needed to establish a comparative succinctness result . . .

There are $n/2$ red and $n/2$ blue items. Assume our bidder wants as many items of the same colour as possible:

$$v(X) \quad = \quad \max\{|X \cap \mathit{Red}|, |X \cap \mathit{Blue}|\}$$

This is called the *monochromatic* valuation.

Representing Monochromatic Valuations

The monochromatic valuation can be used to show that XOR-of-OR can be more concise that OR-of-XOR:

Proposition 4 (Nisan, 2000) *The monochromatic valuation over n goods requires a bid of size at least $2 \cdot 2^{n/2}$ in the OR-of-XOR language, but only a bid of size n in the XOR-of-OR language.*

Proof sketch: Easy part: (OR of all reds) XOR (OR of all blues) ✓
Now, try to represent monochromatic valuation in OR-of-XOR:
(1) Any atomic bid can be assumed to be monochromatic and have a price equal to cardinality. (2) All atoms must belong to same XOR-combination: if one red and one blue bid are in different ones, then overall OR would allow accepting both of them. (3) Hence, we have just one XOR bid. Every red and every blue bundle must be represented; there are $2 \cdot 2^{n/2}$ such atomic bids. ✓

OR-of-XOR vs. XOR-of-OR

- The last result shows that the OR-of-XOR language is not more succinct than the XOR-of-OR language.
- We've seen earlier that the OR-of-XOR language can represent any downward sloping valuation in polynomial size.
- Nisan (2000) gives an example of a special downward sloping valuation (the K-budget valuation) that requires a bid of exponential size in the XOR-of-OR language.
- Hence, the XOR-of-OR language is not more succinct than the OR-of-XOR language either.

OR*: Dummy Items

Idea: Allow bidders to include bids for *"dummy items"* to be able to simulate XOR in the OR-language. Example:

$$(X_1, p_1) \text{ XOR } (X_2, p_2) \text{ can be represented as}$$
$$(X_1 \cup \{d\}, p_1) \text{ OR } (X_2 \cup \{d\}, p_2)$$

The OR* language has exactly the same semantics as the OR language. But now each agent i is assigned a set of dummy items D_i and may submit bids over $\mathcal{R} \cup D_i$. All D_i are *pairwise disjoint*.

The example above shows that OR* can simulate XOR bids. So:

Proposition 5 *The OR* language can represent all valuations.*

Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the Computational Complexity of Combinatorial Auctions. Proc. IJCAI-1999.

OR*: Succinctness

The OR* language is no less succinct than the OR/XOR language:

Proposition 6 (Nisan, 2000) *Any valuation representable by an OR/XOR bid of size s can also be represented by an OR* bid of size s with less than s^2 dummy items.*

Proof: (1) First show how to translate an XOR-combination of two OR* bids into one OR* bid without changing the bid size (but by adding exponentially many dummy items).

$$[(X_1, p_1) \text{ OR } \cdots \text{ OR } (X_k, p_k)] \text{ XOR } [(Y_1, q_1) \text{ OR } \cdots \text{ OR } (Y_l, q_l)]$$

Create dummy items $d_{X_i Y_j}$ and add to both (X_i, p_1) and (Y_j, q_j). We can eliminate all XOR-operators like this (starting from inside). (2) Then reduce number of dummy items. At most, we need one dummy item for any pair of atomic bids to make them exclusive. Hence, at most $s \cdot (s-1)/2 \leq s^2$ dummy items. ✓

The Majority Valuation

Another special valuation is the *majority valuation:*

$$v(X) \quad = \quad \begin{cases} 1 & \text{if } |X| \geq \frac{1}{2} \cdot |\mathcal{R}| \\ 0 & \text{otherwise} \end{cases}$$

That is, the agent will assign a value of 1 to any bundle containing at least half of all the available goods (and 0 otherwise).

Representing Majority Valuations

Even the OR* language cannot represent all valuations succinctly. It requires exponential space in the case of the majority valuation:

Proposition 7 (Nisan, 2000) *The majority valuation over n goods requires a bid of size $\binom{n}{n/2}$ in the OR* language.*

Proof: No atomic bid involving less than $n/2$ (real) items can appear in the OR*-combination (otherwise accepting just those items would yield the wrong value). Hence, every possible bid involving exactly $n/2$ (real) items must have price 1. There are $\binom{n}{n/2}$ such bids. ✓

Note: $\binom{n}{k} = \frac{n! \cdot (n-k)!}{k!}$ is the number of different subsets of size k of a given set of size n (exponential for $k = n/2$).

Winner Determination

- Existing algorithms for winner determination can deal with both OR and XOR. However, algorithms for the (standard) OR language have been around for longer and can be expected to work better.
- An advantage of the OR* language is that algorithms expecting input in the OR language are immediately applicable, *and* we have full expressive power.

Summary

We have given an overview of standard language constructs to build bidding languages for combinatorial auctions:

- *OR* and *XOR* and combinations
- OR*: use of *dummy items* to model exclusiveness
- The standard OR language is not fully *expressive*; all other languages considered are (at least with respect to normalised and monotonic valuations).
- Various results on *comparative succinctness*
- Of course, we could think of *other* language constructs as well. They will be *useful* if they can represent interesting valuations more concisely, and if they can be handled by WDP algorithms.

Remember that bidding languages are just another group of languages for modelling preferences in combinatorial domains.