A Domain Theory for Task Oriented Negotiation[4]

Gilad Zlotkin
Jeffrey S. Rosenschein
Computer Science Department
Hebrew University
Givat Ram, Jerusalem, Israel

Abstract

We present a general theory that captures the relationship between certain domains and negotiation mechanisms. The analysis makes it possible to categorize precisely the kinds of domains in which agents find themselves, and to use the category to choose appropriate negotiation mechanisms. The theory presented here both generalizes previous results, and allows agent designers to characterize new domains accurately. The analysis thus serves as a critical step in using the theory of negotiation in real-world applications.

We show that in certain Task Oriented Domains, there exist distributed consensus mechanisms with simple and stable strategies that lead to efficient outcomes, even when agents have incomplete information about their environment. We also present additional novel results, in particular that in concave domains using all-or-nothing deals, no lying by an agent can be beneficial, and that in subadditive domains, there often exist beneficial decoy lies that do not require full information regarding the other agent's goals.

1 Introduction

Negotiation has been a subject of central interest in Distributed Artificial Intelligence (DAI). The word has been used in a variety of ways, though in general it refers to communication processes that further coordination [Smith, 1978; Kuwabara and Lesser, 1989; Conry *et ai,* 1988; Kreifelts and von Martial, 1990]. These negotiating procedures have included the exchange of Partial Global Plans [Durfee, 1988], the communication of information intended to alter other agents' goals [Sycara, 1988; Sycara, 1989], and the use of incremental suggestions leading to joint plans of action [Kraus and Wilkenfeld, 1991].

In previous work [Zlotkin and Rosenschein, 1989; Zlotkin and Rosenschein, 1991a; Zlotkin and Rosenschein, 1990; Zlotkin and Rosenschein, 1991b], we have considered various negotiation protocols in different domains, and examined their properties. The background and motivation of this research can be found in [Rosenschein, 1993]. Agents were assumed to have a goal that specified a set of acceptable final states. These agents then entered into an iterative process of offers and counter-offers, exploring the possibility of achieving their goals at lower cost, and/or resolving conflicts between their goals.

The procedure for making offers was formalized in a *negotiation mechanism,,* it also specified the form that the agents' offers could take (deal types). A deal between agents was generally a joint plan. The plan was "joint" in the sense that the agents might probabilistically share the load, compromise over which agent does which actions, or even compromise over which agent gets which parts of its goal satisfied.

The interaction between agents occurs in two consecutive stages. First the agents negotiate, then they execute the entire joint plan that has been agreed upon. No divergence from the agreed deal is allowed. The sharp separation of stages has consequences, in that it rules out certain negotiation tactics that might be used in an interleaved process.

At each step, both agents simultaneously offer a deal. Our protocol specifies that at no point can an agent demand more than it did previously—in other words, each offer either repeats the previous offer or concedes by demanding less. The negotiation can end in one of two ways:

Conflict: if neither agent makes a concession at some step, they have by default agreed on the (domain dependent) "conflict deal";

Agreement: if at some step an agent A_1, for example, offers agent A_2 more than A_2 himself asks for, they agree on A_1's offer, and if both agents overshoot the other's demands, then a coin toss breaks the symmetry.

The result of these rules is that agents cannot backtrack, nor can they both simultaneously "stand still" in the negotiation more than once (since it causes them to reach a conflict). Thus the negotiation process is strongly monotonic and ensures convergence to a deal.

Deal types explored in our previous work included *pure deals, all-or-nothing deals, mixed deals, joint plans, mixed joint plans, semi-cooperative deals,* and *multi-plan*

*This research has been partially supported by the Leibniz Center for Research in Computer Science, and by the Israeli Ministry of Science and Technology (Grant 032-8284).

deals. Each of these types of agreement proved suitable for solving different kinds of interactions. For example, semi-cooperative deals proved capable of resolving true conflicts between agents, whereas mixed deals did not. Similarly, multi-plan deals are capable of capturing goal relaxation as part of an agreement.

It was also shown that certain other properties were true of some deal types but not of others. In particular, different agent strategies were appropriate ("rational") for different deal types and domains. Agents were shown to have no incentive to lie when certain deal types were used in certain domains, but did have an incentive to lie with other deal type/domain combinations.

The examination of this relationship between the negotiation mechanism and the domain made use of two prototypical examples: the Postmen Domain (introduced in [Zlotkin and Rosenschein, 1989]), and the Slotted Blocks World (presented in [Zlotkin and Rosenschein, 1991a]). It was clear that these two domains exemplified general classes of multi-agent interactions (e.g., the Postmen Domain was inherently cooperative, the Slotted Blocks World not). It was, however, not clear what attributes of the domains made certain negotiation mechanisms appropriate for them. Nor was it clear how other domains might compare with these prototypes. When presented with a new domain (such as agents querying a common database), which previous results were applicable, and which weren't? The research lacked a general theory explaining the relationship between domains and negotiation mechanisms.

In this paper, we present the beginnings of such a general theory. The analysis makes it possible both to understand previous results in the Postmen Domain more generally, and to characterize new domains accurately (i.e., what negotiation mechanisms are appropriate). The analysis thus serves as a critical step in using the theory of negotiation in real-world applications.

1.1 Criteria for Evaluating Mechanisms

How can we, in general, evaluate alternative interaction mechanisms? We are concerned with several criteria in our design of negotiation mechanisms and strategies:
Symmetric Distribution: no agent is to have a special role in the negotiation mechanism;
Efficiency: the solution arrived at through negotiation should be efficient (e.g., satisfy the criterion of Pareto Optimality);
Stability: the strategy should be stable (e.g., strict Nash equilibrium, where no single agent can benefit by changing strategy, though a group might);
Simplicity: there should be low communication cost to the mechanism, as well as relatively low computational complexity.

Our overall goal is to find distributed consensus mechanisms such that an automated agent can use a simple and stable strategy that will lead to an efficient outcome. In this paper, we show that such mechanisms exist for certain domains.

2 Task Oriented Domains

A *Task Oriented Domain* (TOD) describes a certain class of scenarios for multi-agent encounters. In particular, the Postmen Domain [Zlotkin and Rosenschein, 1989] is an instance of a TOD (the Slotted Blocks World from [Zlotkin and Rosenschein, 1991a], however, is not a TOD). Intuitively, it is a domain that is cooperative, with no negative interactions among agents' goals. Each agent welcomes the existence of other agents, for they can only benefit from one another (if they can reach agreement about sharing tasks).

Definition 1 *A* Task Oriented Domain *(TOD) is a tuple* $< \mathcal{T}, \mathcal{A}, c >$ *where:*
1. $\mathcal{T}$ *is the set of all possible tasks;*
2. $\mathcal{A} = \{A_1, A_2, \ldots A_n\}$ *is an ordered list of agents;*
3. c *is a* monotonic *function* $c: [2^{\mathcal{T}}] \to \mathbf{R}^+$. $[2^{\mathcal{T}}]$ *stands for all the finite subsets of* $\mathcal{T}$. *For each finite set of tasks* $X \subseteq \mathcal{T}$, $c(X)$ *is the cost of executing all the tasks in* X *by a single agent.* c *is monotonic, i.e., for any two finite subsets* $X \subseteq Y \subseteq \mathcal{T}, c(X) \leq c(Y)$.
4. $c(\emptyset) = 0$.

Definition 2 *An* encounter *within a TOD* $< \mathcal{T}, \mathcal{A}, c >$ *is an ordered list* $(T_1, T_2, \ldots, T_n)$ *such that for all* $k \in \{1 \ldots n\}$, T_k *is a* finite *set of tasks from* $\mathcal{T}$ *that* A_k *needs to achieve.* T_k *will also be called* A_k*'s* goal.

According to the definition above, the cost function c takes no parameters other than the task set. In general, c might be defined as having other, global, parameters (like the initial state of the world). However, the cost of a set of tasks is independent of others' tasks that need to be achieved. An agent in a TOD is certain to be able to achieve his goal at that cost.

3 Attributes and Examples

Here we give several examples of TOD's, which cover a variety of agent interaction situations. Subsequently, we will further classify each of these TOD examples with respect to certain properties.

Postmen Domain:
Description: Agents have to deliver sets of letters to mailboxes, which are arranged on a weighted graph $G — G(V, E)$. There is no limit to the number of letters that can fit in a mailbox. After delivering all letters, agents must return to the starting point (the post office). Agents can exchange letters at no cost while they are at the post office, prior to delivery.
Task Set: The set of all addresses in the graph, namely V. If address x is in an agent's task set, it means that he has at least one letter to deliver to x.
Cost Function: The cost of a subset of addresses $X \subseteq V$, i.e., $c(X)$, is the length of the minimal path that starts at the post office, visits all members of X, and ends at the post office.

Database Queries:
Description: Agents have access to a common database, and each has to carry out a set of queries. The result of each query is a set of records. For example, agent $A\backslash$ may want the records satisfying the condition "All

female employees of company X earning over $40,000 a year," and agent A_2 may want the records satisfying the condition "All female employees of company X with more than 10 years of seniority." Agents can exchange results of queries and sub-queries at no cost.

Task Set: All possible queries, expressed in the primitives of relational database theory, including operators like Join, Projection, Union, Intersection, and Difference.

Cost Function: The cost of a set of queries is the minimal number of database operations needed to generate all the records. It is possible to use the result of one query as input to other queries, i.e., the operations are not destructive.

The Fax Domain:

Description: Agents are sending faxes to locations on a telephone network (a weighted graph). In order to send a fax, an agent must establish a connection with the receiving node; once the connection is established, multiple faxes can be sent. The agents can, at no cost, exchange messages to be faxed.

Task Set: The set of all possible receiving nodes in the network. If node x is in an agent's task set, it means that he has at least one fax to send to x.

Cost Function: There is a cost associated with establishing a single connection to any node x. The cost of a set of tasks is the sum of the costs of establishing connections to all the nodes in the set. Thus, the cost of a dial-up connection to a given node is independent of other nodes in the task set.

Having introduced the TOD's above, we now turn our attention to attributes that these domains exhibit. These attributes strongly affect their relationships to negotiation mechanisms. We will focus on the attributes of *subadditivity, concavity,* and *modularity* (these terms are borrowed from game theory). The motivation for these definitions are presented in more detail below.

Definition 3 [Subadditivity]: *TOD* $< \mathcal{T}, \mathcal{A}, c >$ *will be called* **subadditive** *if for all finite sets of tasks* $X, Y \subseteq \mathcal{T}$, *we have* $c(X \cup Y) \leq c(X) + c(Y)$.

In other words, by combining sets of tasks we may reduce (and can never increase) the total cost, as compared with the cost of achieving the sets alone. All the TOD examples above are subadditive. In this paper, we are mainly concerned with two agent subadditive domains.

Definition 4 [Concavity]: *TOD* $< \mathcal{T}, \mathcal{A}, c >$ *will be called* **concave** *if for all finite sets of tasks* $X \subseteq Y, Z \subseteq \mathcal{T}$, *we have* $c(Y \cup Z) - c(Y) \leq c(X \cup Z) - c(X)$.

In other words, the cost that arbitrary set of tasks Z adds to set of tasks Y cannot be greater than the cost Z would add to a subset of Y.

Theorem 1 *All concave TOD's are also subadditive.*

Proof. The proof of this theorem and all other theorems can be found in [Zlotkin and Rosenschein, 1992]. o

The general Postmen Domain is *not* concave. The other TOD examples (the Fax Domain and the Database Query Domain) are concave.

Theorem 2 *The Postmen Domain, restricted to graphs that have a tree topology (no cycles), is concave.*

Definition 5 [Modularity]: *TOD* $< \mathcal{T}, \mathcal{A}, c >$ *will be called* **modular** *if for all finite sets of tasks* $X, Y \subseteq \mathcal{T}$, *we have* $c(X \cup Y) = c(X) + c(Y) - c(X \cap Y)$.

In other words, the cost of the combination of two sets of tasks is exactly the sum of their individual costs minus the cost of their intersection.

Theorem 3 *All modular TOD's are also concave.*

Only the Fax Domain from the above TOD examples is modular.

4 Mechanisms for Subadditive TOD's

In this section, we develop the framework for formalizing two agent negotiation mechanisms in subadditive Task Oriented Domains. Similar definitions can be found in our previous work [Zlotkin and Rosenschein, 1989; Zlotkin and Rosenschein, 1991a; Zlotkin and Rosenschein, 1991b].

Definition 6 *Given an encounter* (T_1, T_2) *within a two agent TOD* $< \mathcal{T}, \{A_1, A_2\}, c >$ *we have the following:*

1. A Pure Deal *is a redistribution of tasks among agents. It is an ordered list* (D_1, D_2) *such that* $D_1, D_2 \subseteq \mathcal{T}$, *and* $D_1 \cup D_2 = T_1 \cup T_2$. *The semantics of such a deal is that each agent* A_k *commits itself to executing all tasks in* D_k. *The cost of such a deal to* A_k *is defined to be* $\text{Cost}_k(D_1, D_2) = c(D_k)$.

2. A Mixed Deal *is a pure deal* (D_1, D_2) *and a probability* $p, 0 \leq p \leq 1$. *A mixed deal will be denoted by* $(D_1, D_2){:}p$. *The semantics of this deal is that the agents will perform a lottery such that, with probability* p, D_1 *will be assigned to* A_1 *and* D_2 *will be assigned to* A_2. *With probability* $1-p$, D_1 *will be assigned to* A_2 *while* D_2 *will be assigned to* A_1. *The cost of such a deal to* A_k *is defined to be* $\text{Cost}_k((D_1, D_2){:}p) = (p)c(D_k) + (1-p)c(D_{3-k})$.

3. An All-Or-Nothing *deal is a mixed deal* $(T_1 \cup T_2, \emptyset){:}p$. *Agreeing on such a deal,* A_1 *has a* p *chance of executing all the tasks* $T_1 \cup T_2$ *and has a* $1-p$ *chance of doing nothing.*

With the above definitions of three deal types, we now consider utility, the negotiation set, optimal protocols, and stable negotiation strategies.

Definition 7 *Given an encounter* (T_1, T_2) *within a TOD* $< \mathcal{T}, \{A_1, A_2\}, c >$, *we have the following:*

1. For any deal δ *(pure, all-or-nothing, or mixed) we will define* $\text{Utility}_k(\delta) \equiv c(T_k) - \text{Cost}_k(\delta)$.

2. The (pure) Deal $\Theta \equiv (T_1, T_2)$ *will be called the* conflict deal.

Θ is a conflict because no agent agrees to execute tasks other than its own. Note that for all k, $\text{Utility}_k(\Theta) = 0$. When the agents fail to agree, i.e., run into a conflict, they by default execute the conflict deal Θ. Our assumption is that rational agents are utility maximizers; since they can guarantee themselves utility 0, they will not agree to any deal that gives them negative utility.

Definition 8 *For vectors* $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ *and* $\beta = (\beta_1, \beta_2, \ldots, \beta_n)$, *we will say that* α dominates β *and write* $\alpha \succ \beta$ *if and only if* $\forall k(\alpha_k \geq \beta_k)$, *and* $\exists l(\alpha_l > \beta_l)$. *We will say that* α weakly dominates β *and write* $\alpha \succeq \beta$ *if and only if* $\forall k(\alpha_k \geq \beta_k)$.

Definition 9 *For deals* δ *and* δ' *(pure, all-or-nothing, or mixed), we will say that* δ dominates δ', *and write* $\delta \succ \delta'$, *if and only if* $(\text{Utility}_1(\delta), \text{Utility}_2(\delta)) \succ (\text{Utility}_1(\delta'), \text{Utility}_2(\delta'))$. *We will say that* δ weakly dominates δ', *and write* $\delta \succeq \delta'$, *if and only if* $(\text{Utility}_1(\delta), \text{Utility}_2(\delta)) \succeq (\text{Utility}_1(\delta'), \text{Utility}_2(\delta'))$. *We will say that* δ *is* equivalent *to* δ', *and write* $\delta \equiv \delta'$ *if* $\forall k(\text{Utility}_k(\delta) = \text{Utility}_k(\delta'))$.

If $\delta \succ \delta'$ it means that the deal δ is better for at least one agent and not worse for the other.

Definition 10 *Deal* δ *is* individual rational *if* $\delta \succeq \Theta$.

A simple observation from the above definition and from Definition 7 (of the conflict deal and utility) is that a deal δ is individual rational if and only if $\forall k \in \{1, 2\}$: $\text{Utility}_k(\delta) \geq 0$.

Definition 11 *A deal* δ *is called* pareto optimal *if there does not exist another deal* δ' *such that* $\delta' \succ \delta$ *[Roth, 1979; Luce and Raiffa, 1957; Harsanyi, 1977].*

A pareto optimal deal cannot be improved upon for one agent without lowering the other agent's utility from the deal.

Definition 12 The set of all deals thai are individual rational and pareto optimal is called the negotiation set (NS) [Harsanyi, 1977].

Since agents are by definition indifferent between two deals that give them the same utility, we are interested in negotiation mechanisms that produce pareto optimal deals (i.e., if agent A\ gets the same utility from deals x and y, but A-i prefers y, we don't want them to settle on x). At this point, we are only considering negotiation mechanisms that result in a deal from the NS. These are, in some sense, mechanisms with efficient outcomes.

Theorem 4 For any encounter in a TOD, NS over pure deals is not empty.

Theorem 5 For any encounter within any subadditive TOD, NS over mixed deals is not empty.

Definition 13 An optimal negotiation mechanism over a set of deals ts a mechanism that has a negotiation strategy that is in equilibrium with itself—if all agents use this negotiation strategy, they will agree on a deal in NS that maximizes the product of the agents' utility [Nash, 1950]. If there is more than one such deal that maximizes the product, the mechanism chooses one arbitrarily, with equal probability.

An optimal negotiation mechanism by definition satisfies the stability and efficiency criteria mentioned in Section 1.1.

The protocol defined above in Section 1 has an equilibrium strategy for each deal type that yields agreement on a deal in NS that maximizes the product of the agents' utility. Those strategies are based on Zeuthen risk criteria [Zeuthen, 1930], and were presented in [Zlotkin and Rosenschein, 1989]. Therefore, the above protocol is an example of an optimal negotiation mechanism.

Theorem 6 An optimal negotiation procedure over mixed deals in subadditive two agent TOD's divides the available utility equally between the two agents.

5 Incentive Compatible Mechanisms

Sometimes agents do not have full information about one another's goals. This raises the question of whether agents can benefit from concealing goals, or manufacturing artificial goals. This lying can either occur explicitly, by declaring false goals, or implicitly, by behaving as if these false goals were true, depending on the specific negotiation mechanism. Our work in previous papers [Zlotkin and Rosenschein, 1989; Zlotkin and Rosenschein, 1991a; Zlotkin and Rosenschein, 1991b] partly focused on combinations of negotiation mechanisms and domains where agents have no incentive to lie. A negotiation mechanism is called incentive compatible when the strategy of telling the truth (or behaving according to your true goals) is in equilibrium (i.e., when one agent uses the strategy, the best thing the other agent can do is use the same strategy). In the Postmen Domain [Zlotkin and Rosenschein, 1989], we identified three types of lies:
1. Hiding tasks (e.g., a letter is hidden);
2. Phantom tasks (e.g., the agent claims to have a letter, which is non-existent and cannot be produced by the lying agent);
3. Decoy tasks (e.g., the agent claims to have a letter, which is non-existent but can be manufactured on demand if necessary).

Since certain deals might require the exchange of letters, a phantom lie can be uncovered, while a decoy lie (and of course a hidden lie) cannot. Thus, a phantom lie under certian negotiation mechanisms is "not safe." Different domains differ as to how easy or hard it is to generate decoy tasks.

In this section, we provide a characterization of the relationship between kinds of lies, domain attributes, and deal types. There are three kinds of lies in TOD's, and we have considered three domain attributes (subadditivity, concavity, modularity) and three classes of optimal negotiation mechanisms, based on pure, all-or-nothing, and mixed deals. The resulting three-by-three-by-three matrix is represented in Figure 1. Its notation is described below.

Consider the entry under Subadditive, All-or-Nothing deal, Decoy lie (we'll refer to this as entry [a, j, z]). The entry L at that position means that for every optimal negotiation mechanism that is based on all-or-nothing deals, there exists a subadditive domain and an encounter such that at least one agent has the incentive to lie with a decoy lie (L means lying may be beneficial). The entry T at position [b, k, z] means that for every concave domain and every encounter within this domain, under any optimal negotiation mechanism based on mixed deals, agents do not have an incentive to lie with decoy lies (T means telling the truth is always beneficial).

The entries in the table marked T/P (such as [a, j, y])

$10 - \frac{3}{4}4 - \frac{1}{4}10 = 4.5$ which is greater than 4, the utility he would have gotten if he had told the truth.

The all-or-nothing deal is not beneficial for A_1 because the agents would agree on the probability $p = \frac{5}{12}$, which would give agent A_1 a real utility of $10 - \frac{5}{12}12 - \frac{5}{12}2 = 3\frac{5}{6} < 4$. However, the expected payoff for the lying agent is $4\frac{1}{6}$, i.e., still over 4, even when the negotiation mechanism sometimes chooses the all-or-nothing deal, so the lying agent benefits.

6 Conclusions

We have presented a general domain theory to use in analyzing negotiation protocols. In order to use negotiation protocols for automated agents in real-world domains, it is necessary to have a clear understanding of when different protocols are appropriate. In this paper, we have characterized Task Oriented Domains (TOD's), which cover an important set of multi-agent interactions.

We have presented several examples of TOD's, and examined three attributes that these domains can exhibit, namely subadditivity, concavity, and modularity. We have then enumerated the relationship between deal types, domain attributes, and types of deception, focusing on whether an agent in a TOD with a given attribute and deal type is motivated to always tell the truth. In particular, we have shown that in concave TOD's, there is no benefit to an agent's lying when all-or-nothing deals are in use. In a general subadditive domain, however, when agents are able to generate decoy tasks, even all-or-nothing deals are not sufficient to create stability (discourage lies). In addition, we demonstrated that in subadditive domains, there often exist beneficial decoy lies that do not require full information regarding the other agent's goals.

References

[Conry et a/., 1988] S. Conry, R. Meyer, and V. Lesser. Multistage negotiation in distributed planning. In A. Bond and L. Gasser, editors, Readings in Distributed Artificial Intelligence, pages 367-384. Morgan Kaufmann Publishers, Inc., San Mateo, 1988.

[Durfee, 1988] E. H. Durfee. Coordination of Distributed Problem Solvers. Kluwer Academic Publishers, Boston, 1988.

[Harsanyi, 1977] John C. Harsanyi. Rational Behavior and Bargaining Equilibrium in Games and Social Situations. Cambridge University Press, Cambridge, 1977.

[Kraus and Wilkenfeld, 1991] S. Kraus and J. Wilkenfeld. Negotiations over time in a multi agent environment: Preliminary report. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, pages 56-61, Sydney, August 1991.

[Kreifelts and von Martial, 1990] Thomas Kreifelts and Frank von Martial. A negotiation framework for autonomous agents. In Proceedings of the Second European Workshop on Modeling Autonomous Agents and Multi-Agent Worlds, pages 169-182, Saint-Quentin en Yvelines, France, August 1990.

[Kuwabara and Lesser, 1989] Kazuhiro Kuwabara and Victor R. Lesser. Extended protocol for multistage negotiation. In Proceedings of the Ninth Workshop on Distributed Artificial Intelligence, pages 129-161, Rosario, Washington, September 1989.

[Luce and Raiffa, 1957] R. Duncan Luce and Howard Raiffa. Games and Decisions. John Wiley & Sons, Inc., New York, 1957.

[Nash, 1950] John F. Nash. The bargaining problem. Econometrica, 28:155-162, 1950.

[Rosenschein, 1993] J. S. Rosenschein. Consenting agents: Negotiation mechanisms for multi-agent systems. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Chambery, France, August 1993.

[Roth, 1979] Alvin E. Roth. Axiomatic Models of Bargaining. Springer-Verlag, Berlin, 1979.

[Smith, 1978] Reid G. Smith. A Framework for Problem Solving in a Distributed Processing Environment. PhD thesis, Stanford University, 1978.

[Sycara, 1988] K. P. Sycara. Resolving goal conflicts via negotiation. In Proceedings of the Seventh National Conference on Artificial Intelligence, pages 245-250, St. Paul, Minnesota, August 1988.

[Sycara, 1989] K. P. Sycara. Argumentation: Planning other agents' plans. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pages 517-523, Detroit, August 1989.

[Zeuthen, 1930] Frederik Zeuthen. Problems of Monopoly and Economic Walfare. G. Routledge & Sons, London, 1930.

[Zlotkin and Rosenschein, 1989] G. Zlotkin and J. S. Rosenschein. Negotiation and task sharing among autonomous agents in cooperative domains. In Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pages 912-917, Detroit, Michigan, August 1989.

[Zlotkin and Rosenschein, 1990] G. Zlotkin and J. S. Rosenschein. Negotiation and goal relaxation. In Proceedings of The Workshop on Modelling Autonomous Agents in a Multi-Agent World, pages 115-132, Saint-Quentin en Yvelines, France, August 1990.

[Zlotkin and Rosenschein, 1991a] G. Zlotkin and J. S. Rosenschein. Cooperation and conflict resolution via negotiation among autonomous agents in noncooperative domains. IEEE Transactions on Systems, Man, and Cybernetics, 21(6):1317-1324, December 1991.

[Zlotkin and Rosenschein, 1991b] G. Zlotkin and J. S. Rosenschein. Incomplete information and deception in multi-agent negotiation. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, pages 225-231, Sydney, Australia, August 1991.

[Zlotkin and Rosenschein, 1992] G. Zlotkin and J. S. Rosenschein. A domain theory for task oriented negotiation. Technical Report 92-13, The Leibniz Center, Department of Computer Science, Hebrew University, Jerusalem, Israel, 1992.