Negotiation Using Logic Programming with Consistency Restoring Rules

Tran Cao Son[1] and Chiaki Sakama[2]
[1]New Mexico State University, USA
[2]Wakayama University, Japan

IJCAI, 2009

Outline

Motivation

Example

1. *Seller:* Would you like to have this PC for $1000?
2. *Buyer:* Can I get it for $900?
3. *Seller:* If you are a senior citizen, we can offer you this price.
4. *Buyer:* Oh no, I am a student. Is there any way I can get this price?
5. *Seller:* Will you be able to pay in cash?
6. *Buyer:* Yes.

Agents reason with predefined rules, facts

Example

1. *Seller:* Would you like to have this PC for $1000?
2. *Buyer:* Can I get it for $900?
3. *Seller:* If you are a senior citizen, we can offer you this price.
   IF senior_citizen THEN lower_price
4. *Buyer:* Oh no, I am a student. Is there any way I can get this price?
   FACT: ¬ senior_citizen
5. *Seller:* Will you be able to pay in cash?
   IF pay_cash THEN lower_price
6. *Buyer:* Yes.

Agents reason with assumptions

Example

1. *Seller:* Would you like to have this PC for $1000?
2. *Buyer:* Can I get it for $900?
3. *Seller:* If you are a senior citizen, we can offer you this price.

   IF senior_citizen THEN lower_price

   ASSUME senior_citizen THEN lower_price
4. *Buyer:* Oh no, I am a student. Is there any way I can get this price?

   FACT: ¬ senior_citizen
5. *Seller:* Will you be able to pay in cash?

   IF pay_cash THEN lower_price
6. *Buyer:* Yes.

Goal

Formalizing and modeling of negotiated agents who

1. have predefined knowledge for negotiations;
2. make assumptions during negotiations (e.g., payment method, eligibility for discount, etc.);
3. need to deal with
   1. incomplete information (complete the knowledge about the other party through dialog).
   2. preference (priority among rules and assumptions)

This paper

Objective

Formalizing negotiation using logic programming under the answer set semantics

1. How do the agents come up with the proposal?
2. What are the components of a proposal?
3. What is the reasoning process behind the creation of the proposals?

This paper

Objective

Formalizing negotiation using logic programming under the answer set semantics

1. How do the agents come up with the proposal?
2. What are the components of a proposal?
3. What is the reasoning process behind the creation of the proposals?

Results

A formalism for negotiation based on logic programming with consistency restoring rules (an extension of logic programming), that can deal with

1. preferences
2. incomplete information
3. changing goals

Disjunctive logic programming I

A *disjunctive logic program* P is a set of *rules* of the form

$$c_1 \mid \ldots \mid c_k \quad \leftarrow \quad a_1, \ldots, a_m, not\ a_{m+1}, \ldots, not\ a_n$$

Intuition: if the body is believed to be true then the head must be true.
The *reduct of P* w.r.t. S is a program P^S, obtained from P by

1. deleting all the rules in P whose body contains some $not\ a$ such that $a \in S$;
2. removing all the remaining default literals.

Answer sets

S is an answer set of P if S is a minimal set of literals satisfying P^S.

Answer sets = Possible Worlds

Logic programming with consistency restoring rules (CR-Prolog)

A *CR-Prolog program* P is a pair (P^r, P^c) where

1. P^r is a disjunctive logic program and
2. P^c is a set of *consistency restoring rules* (cr-rule), each is of the form

$$r: \quad c_1 \mid \ldots \mid c_k \stackrel{+}{\leftarrow} a_1, \ldots, a_m, not\ a_{m+1}, \ldots, not\ a_n$$

   where r is the rule name, and

$$c_1 \mid \ldots \mid c_k \leftarrow a_1, \ldots, a_m, not\ a_{m+1}, \ldots, not\ a_n$$

   is a normal disjunctive rule (denoted by r^*).
3. P contains atoms of the form $prefer(r_1, r_2)$ where r_1 and r_2 are names of consistency restoring rules.

CR-Prolog: Semantic of $P = (P^r, P^c)$ I

Intuition

- P^r expresses strict knowledge
- P^c encodes rules for just-in-case situations and assumptions

Semantic

- if P^r has answer sets then answer sets of P^r are answer sets of P;
- if P^r does not have answer sets then S is an answer set of P if S is an answer set of $P^r \cup R^*$ where $R \subseteq P^c$ and $R^* = \{r^* \mid r \in R\}$ and
  - R is a minimal set of rules in P^c such that $P^r \cup R^*$ is consistent; and
  - the transitive closure of $prefer$ is respected by answer sets of $P^r \cup R^*$.

CR-Prolog: Semantic of $P = (P^r, P^c)$ II

Example

$$\overbrace{p \leftarrow not\ p.\quad prefer(r_1, r_2).}^{P^r} \qquad \overbrace{[r_1] : p \overset{+}{\leftarrow} \quad [r_2] : p \overset{+}{\leftarrow} q \quad [r_3] : q \overset{+}{\leftarrow}}^{P^c}$$

- The program P^r is inconsistent.
- $P^r \cup \{r_1^*\}$ is consistent and has the answer set $\{p, prefer(r_1, r_2)\}$, which is the unique answer set of P.
- $P^r \cup \{r_1^*, r_2^*, r_3^*\}$ is also consistent but its answer sets are not answer sets of P (minimality of set of CR-rules is violated).

Formalizing negotiation using CR-Prolog

We will

- encode knowledge for negotiation as CR-programs extended with a set of assumptions and a set of prioritized negotiated literals;
- use semantics of CR-programs to formalize the basic concepts of negotiation (proposal and counter-proposal);
- define the notion of a negotiation and of a negotiation tree;
- develop algorithms for computing negotiations and negotiation trees

Negotiation knowledge bases (n-KB)

Negotiation knowledge base

=

CR-Program + Assumptions + Goals with Strict Partial Order

$\langle P^r, P^c, H, N^{\prec} \rangle$

- (P^r, P^c) is a CR-program (the knowledge base that the agent will use in negotiation),
- $N^{\prec}$ is a set of *negotiated literals* associated with a strict partial order $\prec$ on its elements (specifying the prioritized goals, that the agent can negotiate for), and
- H is a set of literals (called *assumptions*) such that $H \cap head(P^r) = \emptyset$ and $\{H \xleftarrow{+}\} \subseteq P^c$ (assumptions that can be made during the negotiation process).

The n-KB is *consistent* if (P^r, P^c) is consistent.

Example: Seller's knowledge base

Goal: make a sale $\leftarrow not\ sale$

- Any sale is a sale

$sale \leftarrow high\_pr$

$sale \leftarrow low\_pr$

$sale \leftarrow lowest\_pr$

- Only registered customers are whole sale customers.

$whole\_sale\_customer \leftarrow \underline{registered}$

- Customers with a valid student identification are student customers.

$student\_customer \leftarrow \underline{student}$

- Senior customers are entitled to a discount.

$low\_pr \stackrel{+}{\leftarrow} senior\_customer$

- ...
- Some information on the current stock:

$made\_in\_L, maker\_A, \neg maker\_C, \neg maker\_B$

Example: Seller's n-KB $K_S = \langle P_S^r, P_S^c, H_S, N_S^{\prec} \rangle$

$$Facts_S = \{made\_in\_L, maker\_A, \neg maker\_C, \neg maker\_B\}$$

$$P_s^r = Facts_S \cup \left\{ \begin{array}{rcl} & \leftarrow & not\ sale. \\ whole\_sale\_customer & \leftarrow & \underline{registered}. \\ student\_customer & \leftarrow & \underline{student}. \\ senior\_customer & \leftarrow & \underline{age \geq 65}. \\ sale & \leftarrow & high\_pr. \\ sale & \leftarrow & low\_pr. \\ sale & \leftarrow & lowest\_pr. \\ prefer(r_1, r_i).\ (\text{for } i > 1) & & prefer(r_i, r_5).\ (\text{for } i \in \{2,3,4\}) \end{array} \right\}$$

$$P_s^c = lit(H_S) \cup \left\{ \begin{array}{lcl} r_1: & high\_pr & \stackrel{+}{\leftarrow} \\ r_2: & low\_pr & \stackrel{+}{\leftarrow} senior\_customer. \\ r_3: & low\_pr & \stackrel{+}{\leftarrow} student\_customer, \underline{good\_credit}. \\ r_4: & low\_pr & \stackrel{+}{\leftarrow} student\_customer, \underline{pay\_cash}. \\ r_5: & lowest\_pr & \stackrel{+}{\leftarrow} whole\_sale\_customer, \underline{quantity \geq 100}. \end{array} \right\}$$

$$H_S = \{registered, student, age \geq 65, good\_credit, quantity \geq 100, pay\_cash\}$$

$N_S^{\prec} = \{high\_pr, low\_pr, lowest\_pr\}$ with
$\prec = \{lowest\_pr \prec low\_pr \prec high\_pr\}$.

Example: Buyer's n-KB $K_B = \langle P_B^r, P_B^c, H_B, N_B^{\prec} \rangle$

$$Facts_B = \{age = 25, student, pay\_cash, \neg good\_credit, quantity = 1\}$$

$$P_B^r = Facts_B \cup \left\{ \begin{array}{lcl} & \leftarrow & not\ purchase. \\ purchase & \leftarrow & high\_pr. \\ purchase & \leftarrow & low\_pr. \\ purchase & \leftarrow & lowest\_pr. \\ prefer(r_i, r_1) & \leftarrow & (i > 1) \\ prefer(r_4, r_i) & \leftarrow & (i \in \{2, 3\}) \end{array} \right\}$$

$$P_B^c = lit(H_B) \cup \left\{ \begin{array}{lcll} r_1: & high\_pr & \stackrel{+}{\leftarrow} & \underline{maker\_A}, not\ \underline{made\_in\_L} \\ r_2: & low\_pr & \stackrel{+}{\leftarrow} & maker\_A, made\_in\_L \\ r_3: & low\_pr & \stackrel{+}{\leftarrow} & \underline{maker\_B} \\ r_4: & lowest\_pr & \stackrel{+}{\leftarrow} & \underline{maker\_C} \end{array} \right\}$$

$$H_B = \{maker\_A, maker\_B, maker\_C, made\_in\_L\}$$

$N_B^{\prec} = \{high\_pr, low\_pr, lowest\_pr\}$ with
$\prec = \{high\_pr \prec low\_pr \prec lowest\_pr\}$.

Proposal

Given an agent A with the n-KB $KB = \langle P^r, P^c, H, N \rangle$ and a negotiated goal G, a proposal for G should

1. reflect the fact that there exists a possible world in which G is true and S is the set of assumptions
2. provide the assumptions that A made to generate M and possibly facts that are true in his/her knowledge base

Proposal = Goal + Assumptions + (Facts)

Examples of proposals w.r.t. $KB_S = \langle P_S^r, P_S^c, H_S, N_S^{\prec} \rangle$

1. Would you like to have this product with $high\_pr$?
   $\langle \{high\_pr\}, \emptyset \rangle$
2. We could offer $lowest\_pr$ if you are a registered whole sale customer and you buy more than 100 units.
   $\langle \{lowest\_pr\}, \{registered, quantity \geq 100\} \rangle$
3. We only have PCs produced by A, which are made in L. We could offer them for low_pr if you are a senior citizen.
   $\langle \{low\_pr\}, \{age \geq 65\}, \{maker\_A, made\_in\_L\} \rangle$

Proposal classification

An agent receives a proposal $\langle G, S \rangle$ or an extended proposal $\langle G, S, R \rangle$.

- acceptable: if the receiver can be in a possible world which contains G and is consistent with S (resp. S and R)
  $\langle \{low\_pr\}, \{age \geq 65\}, \{maker\_A, made\_in\_L\} \rangle$
  acceptable to a senior citizen who likes product made in L of $maker\_A$

Proposal classification

An agent receives a proposal $\langle G, S\rangle$ or an extended proposal $\langle G, S, R\rangle$.

- acceptable: if the receiver can be in a possible world which contains G and is consistent with S (resp. S and R)
  $\langle\{low\_pr\}, \{age \geq 65\}, \{maker\_A, made\_in\_L\}\rangle$
  acceptable to a senior citizen who likes product made in L of $maker\_A$
- rejectable: if the receiver cannot be in a possible world which is consistent with S (R)
  $\langle\{low\_pr\}, \{age \geq 65\}, \{maker\_A, made\_in\_L\}\rangle$
  rejectable to a student who only pays for the lowest price for product by $make\_A$.

Proposal classification

An agent receives a proposal $\langle G, S \rangle$ or an extended proposal $\langle G, S, R \rangle$.

- acceptable: if the receiver can be in a possible world which contains G and is consistent with S (resp. S and R)
  $\langle \{low\_pr\}, \{age \geq 65\}, \{maker\_A, made\_in\_L\} \rangle$
  acceptable to a senior citizen who likes product made in L of $maker\_A$
- rejectable: if the receiver cannot be in a possible world which is consistent with S (R)
  $\langle \{low\_pr\}, \{age \geq 65\}, \{maker\_A, made\_in\_L\} \rangle$
  rejectable to a student who only pays for the lowest price for product by $make\_A$.
- negotiatable: otherwise.
  $\langle \{low\_pr\}, \{age \geq 65\}, \{maker\_A, made\_in\_L\} \rangle$
  negotiatable for students who could pay for low_pr for product made in L by $maker\_A$.

Response to $\langle G, S \rangle$ or $\langle G, S, R \rangle$

The receiver knows that

- the condition R must be true
- the proposer assumes S to be true

The receiver can

- accept the proposal if it is acceptable;
- reject the proposal if it is rejectable
- respond to the proposal if it is negotiatable, the response should
  - take into consideration S (or S and R)
  - address the assumptions made by the proposer

(constructive response: provides correct information)

Buyer: $\omega_B^1 = \langle \{low\_pr\}, \{maker\_B\}, \emptyset \rangle$.

Seller's response:

$$\omega_S^1 = \langle \{low\_pr\}, \{age \geq 65\}, \{\neg maker\_B\} \rangle$$
$$\omega_S^2 = \langle \{low\_pr\}, \{student, good\_credit\}, \{\neg maker\_B\} \rangle$$
$$\omega_S^3 = \langle \{low\_pr\}, \{student, pay\_cash\}, \{\neg maker\_B\} \rangle$$

Negotiation: Example

A negotiation is a sequence of exchanges between two agents.

$$\langle \{low\_pr\}, \{maker\_B\}, \emptyset \rangle$$
$$\downarrow B$$
$$\langle \{low\_pr\}, \{age \geq 65\}, \{\neg maker\_B\} \rangle$$
$$\downarrow S$$
$$\langle \{low\_pr\}, \{maker\_A, made\_in\_L\}, \{age = 25\} \rangle$$
$$\downarrow B$$
$$\langle \{low\_pr\}, \{student, good\_credit\}, \emptyset \rangle$$
$$\downarrow S$$
$$\langle \{low\_pr\}, \{maker\_A, made\_in\_L\}, \{\neg good\_credit\} \rangle$$
$$\downarrow B$$
$$\langle \{low\_pr\}, \{student, pay\_cash\}, \emptyset \rangle$$
$$\downarrow S$$
$$\langle \top, \emptyset, \emptyset \rangle$$

Negotiation

Definition

A negotiation is *successful* (resp. *unsuccessful*) if it is finite and ends with $\langle \top, \emptyset, \emptyset \rangle$. (resp. $\langle \bot, \emptyset, \emptyset \rangle$). A negotiation is *constructive* if it contains only constructive responses.

Theorem

Every constructive negotiation is finite.

Negotiation process

- can involve several negotiations, and
- can be represented by a *negotiation tree* whose interior nodes are labeled with proposals and whose leaves are labeled with either *accept* or *reject*.

Negotiation trees can be classified into successful or unsuccessful negotiation trees:

- successful: finite, one leaf has the label *accept*
- unsuccessful: finite, no leaf has the label *accept*

Negotiation trees can be constructed (algorithm for construction of negotiation tree).

Theorem

Every constructive negotiation tree is finite.

Relaxation/Strengthening: Changing goals in negotiation

Necessity: recovering from unsuccessful negotiation

- if G is rejectable, new goal can be derived from the relation $N^{\prec}$
- uses new goal to continue

Results

Most results (finiteness of constructive negotiation tree, algorithm) can be extended to deal with goal changes.

Previous negotiation formalisms

1. most are argumentation based
   - [Rahwan et al, 2004]: survey of many argumentation based negotiation formalisms
   - [Kakas and Moraitis, 2006]: argumentation as the basis for negotiation
2. logic programming based negotiation formalisms are based on belief revision
   - [Chen et al., 2006]: negotiation as repeated logic program with answer sets as the basis, no (counter) proposal computation.
   - [Meyer et al., 2004] defined outcomes, concession, and adaptation to characterize outcomes of a negotiation (given the knowledge base of agents, what is the possible outcomes of a negotiation for φ?)
   - [Zhang et al., 2004] considered negotiation as mutual belief revision.

Conclusions and Future Works

Contributions

An approach to formalizing negotiation

- uses CR-Prolog as representation language and answer sets as the basis for the construction of proposals and counter-proposals
- allows agents to deal with preferences, incomplete information, and goal changing.

Future works

- develop a system for automated negotiation using available answer set solvers,
- apply the framework in multiagent planning, and
- consider multiagent negotiation.