Coordination between Logical Agents

Chiaki Sakama
Wakayama University

Katsumi Inoue
National Institute of Informatics

CLIMA-V September 2004

Coordination in Multi-Agent Systems

- In MAS different agents may have different beliefs, and agents negotiate and accommodate themselves to reach acceptable agreements.
- A process of forming such agreements between agents is called coordination.
- The outcome of coordination is required to be consistent and is desirable to retain original information of each agent as much as possible.

Purpose

- The goal of this research is to provide a logical framework of coordination between agents.
- Each agent has a knowledge base as a logic program and the set of beliefs under the answer set semantics.
- Coordination between two agents is captured as the problem of finding a new program which has the meaning balanced between them.

Problem Description

- Given: two programs P_1 and P_2 ;
- Find:

(1) a program Q satisfying

$$AS(Q) = AS(P_1) \cup AS(P_2) ,$$

Q is generous coordination of P_1 and P_2 ;

(2) a program R satisfying

$$AS(R) = AS(P_1) \cap AS(P_2) ,$$

R is rigorous coordination of P_1 and P_2 ;

where AS(P) is the set of answer sets of P.

Example

To decide the Academy Award of Best Pictures, each member of the Academy nominates films. Now there are 3 members – *p1, p2*, and *p3*, and each member can nominate at most 2 films: *p1* nominates *f1* and *f2*, *p2* nominates *f2* and *f3*, and *p3* nominates *f2*. At this moment, 3 nominees *f1, f2*, and *f3* are fixed.

After final voting, the film *f2* is supported by 3 members and becomes the winner of the Award.

Example

The situation is represented by three programs:

P1: $f1 \; ; \; f2 \leftarrow$,

P2: $f2 \; ; \; f3 \leftarrow$,

P3: $f2 \leftarrow$,

where AS(P1)={ {*f1*}, {*f2*} }, AS(P2)={ {*f2*}, {*f3*} }, and AS(P3)={ {*f2*} }.

A program Q having three answer sets {*f1*}.{*f2*} and {*f3*} is generous coordination.

A program R having the single answer set {*f2*} is rigorous coordination.

Logical Framework

- A program is an extended disjunctive program (EDP) which consists of rules of the form:

  $L_1 ; \ldots ; L_l \leftarrow L_{l+1}, \ldots, L_m, \textit{not}\ L_{m+1}, \ldots, \textit{not}\ L_n$

  where L_i is a literal and *not* represents NAF.
  A rule r is also written as $head(r) \leftarrow body(r)$
  with $head(r) = \{ L_1, \ldots, L_l \}$ and
  $body(r) = \{ L_{l+1}, \ldots, L_m, \textit{not}\ L_{m+1}, \ldots, \textit{not}\ L_n \}$.
- The semantics of an EDP is given by answer sets (Gelfond & Lifschitz, 1991).

Terminology and Notation

- A program is consistent if it has a consistent answer set. (The contradictory answer set *Lit* is not considered.)
- A literal L is a consequence of credulous/skeptical reasoning in a program P (written as $L \in crd(P)$ / $L \in skp(P)$) if L is included in some/every answer set.
- Two programs P_1 and P_2 are AS-combinable if every set in $AS(P_1) \cup AS(P_2)$ is minimal under set inclusion.

Properties of Coordination

- For two programs P_1 and P_2 , let Q (resp. R) be a result of generous (resp. rigorous) coordination. We say that generous (resp. rigorous) coordination succeeds if $AS(Q) \neq \Phi$ (resp. $AS(R) \neq \Phi$); otherwise, it fails.
- Generous coordination always succeeds whenever both P_1 and P_2 are consistent. By contrast, rigorous coordination fails when $AS(P_1) \cap AS(P_2) = \Phi$.
- When generous/rigorous coordination of two programs succeeds, the result of coordination is consistent.

Properties of Coordination

Let P_1 and P_2 be two programs.

- If Q is a result of generous coordination,
  (a) $crd(Q) = crd(P_1) \cup crd(P_2)$;
  (b) $skp(Q) = skp(P_1) \cap skp(P_2)$;
  (c) $crd(Q) \supseteq crd(Pi)$, $skp(Q) \subseteq skp(Pi)$ (i=1,2)

: Q increases credulous consequences but decreases skeptical ones. Reflecting the situation that accepting opinions of the other agent increases alternative choices while weakening the original argument of each agent.

Properties of Coordination

- R is a result of rigorous coordination,
  (a) $crd(R) \subseteq crd(P_1) \cup crd(P_2)$;
  (b) $skp(R) \supseteq skp(P_1) \cap skp(P_2)$ if $AS(R) \neq \Phi$;
  (c) $crd(R) \subseteq crd(Pi)$, $skp(R) \supseteq skp(Pi)$ (i=1,2)

: R reduces credulous consequences but increases skeptical ones. Reflecting the situation that excluding opinions of other party costs abandoning some of one's alternative beliefs, which results in strengthening some original argument of each agent.

Properties of Coordination

Let Q (resp. R) be a result of generous (resp. rigorous) coordination between P_1 and P_2. When $AS(Q)=AS(P_1)$ (resp. $AS(R)=AS(P_1)$), we say that P_1 dominates P_2 under generous (resp. rigorous) coordination.

When $AS(P_1) \subseteq AS(P_2)$, P_2 dominates P_1 under generous coordination, and P_1 dominates P_2 under rigorous coordination.

Note

- When P_1 dominates P_2 under generous/rigorous coordination, a result of generous/rigorous coordination becomes $Q=P_1$ (resp. $R=P_1$).
- The problem of interest is the cases where

  $$AS(P_1) \subsetneq AS(P_2) \text{ and } AS(P_2) \subsetneq AS(P_1)$$

  for generous/rigorous coordination, and

  $$AS(P_1) \cap AS(P_2) \neq \phi$$

  for rigorous coordination.

Computing Generous Coordination

Given two programs P_1 and P_2 ,

$P_1 \oplus P_2 = \{$

$$head(r_1) \,;\, head(r_2) \leftarrow body^*(r_1),\, body^*(r_2)$$

$\mid\ r_1 \in P_1,\ r_2 \in P_2\ \}$ where

$body^*(r_1) = body(r_1) \setminus \{ not\ L \mid L \in T \setminus S \}$ and

$body^*(r_2) = body(r_2) \setminus \{ not\ L \mid L \in S \setminus T \}$

for any $S \in AS(P_1)$ and $T \in AS(P_2)$.

Theorem

Let P_1 and P_2 be two AS-combinable programs. Then,

$$AS(P_1 \oplus P_2) = AS(P_1) \cup AS(P_2).$$

Example

P_1: $p \leftarrow not\ q, \quad q \leftarrow not\ p,$

P_2: $\neg p \leftarrow not\ p\ ,$

where $AS(P_1) = \{ \{p\} , \{q\} \}$ and

$AS(P_2) = \{ \{\neg p\} \}.$

Then, $P_1 \oplus P_2$ becomes

$$p\ ;\ \neg p \leftarrow not\ q,$$

$$q\ ;\ \neg p \leftarrow not\ p$$

where $AS(P_1 \oplus P_2) = \{\{p\}, \{q\}, \{\neg p\}\}$

Computing Rigorous Coordination

Given two programs P_1 and P_2,

$$P_1 \otimes P_2 = \bigcup_{S \in AS(P_1) \cap AS(P_2)} R(P_1,S) \cup R(P_2,S)$$

where

$R(P,S) = \{$

$\quad head(r) \cap S \leftarrow body(r), not\ (head(r) \setminus S) \mid$
$\qquad r \in P \text{ and } r^S \in P^S \ \}$ and

$not\ (head(r) \setminus S) = \{\ not\ L \mid L \in head(r) \setminus S\ \}$.

When $AS(P_1) \cap AS(P_2) = \emptyset$, $P_1 \otimes P_2$ is undefined.

Theorem

Let P_1 and P_2 be two programs. Then,

$AS(P_1 \otimes P_2) = AS(P_1) \cap AS(P_2)$.

Example

P_1: $p \leftarrow not\ q,\ \ not\ r,$
$q \leftarrow not\ p,\ \ not\ r,$
$r \leftarrow not\ p,\ \ not\ q,$

P_2: $p\ ;\ q\ ;\ \neg r \leftarrow not\ r,$

where $AS(P_1) = \{\{p\},\{q\},\{r\}\}$, $AS(P_2) = \{\{p\},\{q\},\{\neg r\}\}$, and $AS(P_1) \cap AS(P_2) = \{\{p\},\{q\}\}$.

Then, $P_1 \otimes P_2$ becomes

$p \leftarrow not\ q,\ \ not\ r,$
$q \leftarrow not\ p,\ \ not\ r,$
$p \leftarrow not\ r,\ \ not\ q,\ \ not\ \neg r\ ,$
$q \leftarrow not\ r,\ \ not\ p,\ \ not\ \neg r\ ,$

where $AS(P_1 \otimes P_2) = \{\{p\},\{q\}\}$

Algebraic Properties

- The operations $\oplus$ and $\otimes$ are commutative and associative.
  ➡ When generous/rigorous coordination are done among more than two agents, the order of computing coordination does not affect the result of final outcome.
- Two types of coordination are mixed among agents, but they are neither absorptive nor distributive.

Discussion

When a set of answer sets is given, it is not difficult to construct a program which has exactly those answer sets.

Given a set of answer sets $\{ S_1, \ldots, S_m \}$,

1. Compute the DNF $S_1 \vee \cdots \vee S_m$,
2. Convert it into the CNF $R_1 \wedge \cdots \wedge R_n$,
3. The set of facts $\{R_1, \ldots, R_n\}$ has the answer sets $\{ S_1, \ldots, S_m \}$.

Discussion

P_1: $sweet \leftarrow apple$, $apple \leftarrow$

P_2: $red \leftarrow apple$, $apple \leftarrow$

where $AS(P_1)=\{\{ sweet, apple \}\}$ and $AS(P_2)=\{\{ red, apple \}\}$.

To get generous coordination, taking the DNF of each answer set produces

$(sweet \wedge apple) \vee (red \wedge apple)$.

Converting it into the CNF, it becomes

$(sweet \vee red) \wedge apple$.

Discussion

As a result, the set of facts

Q: *sweet* ; *red* $\leftarrow$

apple $\leftarrow$

is a program which is generous coordination.

On the other hand, $P_1 \oplus P_2$ becomes

sweet ; *red* $\leftarrow$ *apple*,

apple $\leftarrow$

after eliminating redundant rules.

Discussion

Q and $P_1 \oplus P_2$ have the same meaning.

Which program is more preferable as a result of coordination?

We would like to include as much information as possible from the original programs.

Comparing Q with $P_1 \oplus P_2$, information of dependency between *sweet* (or *red*) and *apple* is lost in Q.

Discussion

- Generally, if there exist different candidates for coordination between two programs, a program which is syntactically closer to the original ones is preferred.
- How to measure such “syntactical closeness” between programs?
- We prefer a result of coordination which inherits dependency relations from the original programs as much as possible.

Discussion

Let (L_1,L_2) be a pair of ground literals s.t. L_1 depends on L_2 in the dependency graph of P. Let $\delta(P)$ be the collection of such pairs in P.

Let P_3 and P_4 be two different programs which are candidates of coordination between P_1 and P_2. We say that P_3 is preferable to P_4 if

$$\Delta(\delta(P_3), \delta(P_1) \cup \delta(P_2)) \subset \Delta(\delta(P_4), \delta(P_1) \cup \delta(P_2)),$$

where $\Delta(S,T)$ represents symmetric difference between two sets S and T.

Discussion

P_1: *sweet* ← *apple*, *apple* ←.

P_2: *red* ← *apple*, *apple* ← .

Q: *sweet* ; *red* ← , *apple* ← .

$P_1 \oplus P_2$: *sweet* ; *red* ← *apple*, *apple* ←.

$\delta(P_1)=\{ (sweet, apple) \}$, $\delta(P_2)=\{ (red, apple) \}$,

$\delta(Q)= \Phi$, and

$\delta(P_1 \oplus P_2)=\{ (sweet, apple), (red, apple) \}$.

Then, $\Delta(\delta(P_1 \oplus P_2), \delta(P_1) \cup \delta(P_2))$

$\subset \Delta(\delta(Q), \delta(P_1) \cup \delta(P_2))$,

so we conclude that $P_1 \oplus P_2$ is preferable to Q.

Final Remarks

- From the viewpoint of answer set programming, the process of computing coordination is considered a program development under a specification that requests a program reflecting the meanings of two or more programs.
- Future work includes investigation of other types of coordination and collaboration, and their characterization in terms of computational logic.