Social Default Theories

Chiaki Sakama

Department of Computer and Communication Sciences
Wakayama University, Sakaedani, Wakayama 640 8510, Japan
sakama@sys.wakayama-u.ac.jp

Abstract. This paper studies a default logic for social reasoning in multiagent systems. A *social default theory* is a collection of default theories with which each agent reasons and behaves by taking attitudes of other agents into account. The semantics of a social default theory is given as *social extensions* which represent the agreement of beliefs of individual agents in a society. We show the use of social default theories for representing social attitudes of agents and for reasoning in *cooperative planning* and *negotiation* among multiple agents.

1 Introduction

In a multiagent society, individual agents are requested to act interactively with other agents. Any problem, which is not solved by a single agent, could be solved cooperatively by exchanging information or sharing resources. It is usually the case, however, that an agent does not always have exact information of other agents. In this case, an agent performs *default reasoning* on belief states of other agents. For instance, suppose that two agents 1 and 2 live in the same apartment and share a car. The agent 1 usually uses the car for shopping if the agent 2 does not use it. The agent 2, on the other hand, normally uses the car to go to a school for picking up her child if the agent 1 does not use it. In this situation, the action of one agent depends on the action of another agent, but each agent does not know the exact plan of another agent. One day, the agent 1 plans to go shopping around noon. She knows that the agent 2 usually uses the car in the afternoon, then she goes to a parking lot. If she finds a car at the parking, she can use it; otherwise, she gives up shopping or waits for the agent 2 to come back. In this example, the agent 1 performs default reasoning on the behavior of another agent and takes an action.

A *default theory* [6] provides a logic for representing and reasoning with incomplete belief of an agent, so that it is natural to represent incomplete beliefs of multiple agents as a collection of default theories. In this case, an individual default theory contains default assumption on the beliefs of other agents as well as assumption on his/her own belief. A default theory has an extension which represents a collection of beliefs of an agent. In the presence of multiple default theories, the notion of extensions would be extended to those representing beliefs consented by every agent. To construct a logic for default reasoning in multiagent systems, this paper introduces the framework of *social default theories*. We show the use of social default theories for representing a variety of social attitudes of individual agents, and for reasoning in *cooperative planning* and *negotiation* among multiple agents.

2 Social Default Theory

In this paper we consider theories represented by a propositional logic language. Every propositional atom has an *annotation* representing an *agent identifier*. For a propositional variable p in the language and an integer i, p^i is called an *annotated atom*. An *annotated formula* is inductively defined as follows: (i) An annotated atom p^i is an *annotated formula*. (ii) If α is an annotated formula, so is $\neg\alpha$. (iii) If α and β are annotated formulas, so are $\alpha \vee \beta$, $\alpha \wedge \beta$, $\alpha \supset \beta$, and $\alpha \equiv \beta$. The meanings of logical connectives are the same as those in classical logic. Annotation is introduced to distinguish beliefs among different agents, so the meaning of a formula, $p^i \vee q^i$ for instance, is equivalent to the formula $p \vee q$ that is a belief of the agent i. Note that two annotated atoms p^i and p^j represent different formulas, so $p^i \wedge \neg p^j$ is consistent as far as $i \neq j$. An annotated formula α in which every atom in α has the annotation i is called an *i-annotated formula*. An annotated formula is simply said a *formula* hereafter. $\mathcal{F}$ is the set of all formulas in the language, and $\mathcal{F}_i$ represents the set of all i-annotated formulas ($\mathcal{F}_i \subseteq \mathcal{F}$). A multiagent society is a finite set of *agents*. Formally, a society is represented by a *social default theory* defined as follows.

Definition 2.1 (social default theory) A *social default theory* (SDT, for short) $\mathcal{S}$ is a tuple of theories $(\Delta_1, \ldots, \Delta_m, \Gamma)$ defined as follows.[1]

1. Each Δ_i ($1 \leq i \leq m$) is a *default theory* which is a finite set of *default rules* $\delta = \frac{\alpha:\beta_1,\ldots,\beta_n}{\gamma}$ where $\alpha \in \mathcal{F}$ and $\gamma \in \mathcal{F}_i$, and for each $1 \leq j \leq n$ there is some $1 \leq k \leq m$ such that $\beta_j \in \mathcal{F}_k$. α, $\beta_1, \ldots, \beta_n$, and γ are called a *prerequisite*, *justifications* and a *consequent*, respectively. A default rule δ is called a *social default rule* if its justifications contain at least one formula $\beta_j \in \mathcal{F}_k$ such that $k \neq i$. A default rule $\frac{:}{\gamma}$ is identified with the formula γ.
2. Γ is a finite set of default rules such that each default rule in Γ has the consequent $\gamma = \mathit{false}$. Any default rule in Γ is called a *social constraint*.

Each default theory Δ_i in $\mathcal{S}$ represents the set of beliefs of an individual agent in the society. By contrast, Γ represents constraints that each agent must obey in the society. Informally, the default rule δ means that "if an agent i believes α, and each of $\beta_1, \ldots, \beta_n$ is consistently assumed with respect to the beliefs of agents k ($1 \leq k \leq m$), then the agent i believes γ". Note that in Δ_i the prerequisite and the justifications of any default rule may contain beliefs of other agents as well as beliefs of the agent i, while the consequent only contains beliefs of the agent i. Beliefs of other agents in the prerequisite represent strong conditions which affect the application of the default rule, while those in the justification represent weak conditions to reach the conclusion.

Example 2.1 Compare $\Delta_1 = \{\frac{\neg anti\text{-}smoker^3:}{smoke^1}\}$ and $\Delta_2 = \{\frac{:\neg anti\text{-}smoker^3}{smoke^2}\}$. In Δ_1 the agent 1 smokes if it is known that the agent 3 is not an anti-smoker. In Δ_2, on the other hand, the agent 2 smokes unless the agent 3 is known to be an anti-smoker. In this sense, the social attitude of the agent 1 is more *considerate* than that of the agent 2.

[1] We define a default theory as in [5].

Thus, SDTs can represent different social attitudes of agents. In this paper, a default theory Δ_i is identified with an agent i, and an SDT $\mathcal{S}$ is identified with a society. If the set of social constraints is empty, $(\,\Delta_1,\ldots,\Delta_m,\emptyset\,)$ is simply written as $(\,\Delta_1,\ldots,\Delta_m\,)$. In an SDT $\mathcal{S}=(\Delta_1,\ldots,\Delta_m,\Gamma)$, the notion of *extension* for each Δ_i $(1\le i\le m)$ is defined as usual [6]. A default theory Δ_i is *consistent* if it has a consistent extension; otherwise, it is *inconsistent*. An SDT $\mathcal{S}$ is *rational* if every Δ_i is consistent. In an SDT an individual agent would have its own extensions, while the society would also have extensions.

Definition 2.2 (social extension) A set E of formulas is a *social extension* of an SDT $\mathcal{S}=(\,\Delta_1,\ldots,\Delta_m,\Gamma\,)$ if E is an extension of the default theory $\bigcup_{i=1}^{m}\ \Delta_i\cup\Gamma$.

A social extension represents a collection of beliefs of individual agents, which are consented by each agent and accord with constraints in the society.

Example 2.2 The car-sharing example in the introduction is represented by the SDT $\mathcal{S}=(\Delta_1,\Delta_2)$ where $\Delta_1=\{\,shopping^1,\ \frac{shopping^1:\,use\_car^1,\,\neg use\_car^2}{use\_car^1}\,\}$ and $\Delta_2=\{\,school^2,\ \frac{school^2:\,use\_car^2,\,\neg use\_car^1}{use\_car^2}\,\}$. Then, $\mathcal{S}$ has two social extensions: $E_1=Th(\{\,shopping^1,\ school^2,\ use\_car^1\,\})$ and $E_2=Th(\{\,shopping^1,\ school^2,\ use\_car^2\,\})$, which represent two possibilities of using a shared car.

Note that in Example 2.2 if (Δ_1,Δ_2) is replaced with $(\Delta_1',\Delta_2',\Gamma)$ where $\Delta_1'=\{\,shopping^1,\ \frac{shopping^1:\,use\_car^1}{use\_car^1}\,\}$, $\Delta_2'=\{\,school^2,\ \frac{school^2:\,use\_car^2}{use\_car^2}\,\}$, and $\Gamma=\{\,\frac{use\_car^1\wedge use\_car^2:}{false}\,\}$, $(\Delta_1',\Delta_2',\Gamma)$ has no social extension. In fact, $Th(\{\,shopping^1,\ school^2,\ use\_car^1,\ use\_car^2\,\})$ is the extension of $\Delta_1'\cup\Delta_2'$, but is not the extension of $\Delta_1'\cup\Delta_2'\cup\Gamma$. Thus, social constraints are effective to eliminate useless extensions. Note also that if (Δ_1,Δ_2) in Example 2.2 is replaced with (Δ_1',Δ_2), (Δ_1',Δ_2) has the single social extension E_1. In this situation, Δ_1' does not take care of the usage of the car by the agent 2, and in this sense, the agent 1 is *self-interested*. A default theory Δ_i tends to be more self-interested if it contains less social default rules. When a society consists of self-interested agents, it is hard to reach an agreement. $(\Delta_1',\Delta_2',\Gamma)$ represents such a situation.

Proposition 2.1 *For any social extension E of a rational SDT $\mathcal{S}$, there is an extension G of some Δ_i in $\mathcal{S}$ such that $(E\cap\mathcal{F}_i)\subseteq G$.*

Proposition 2.1 represents that a social extension includes a part of beliefs of some individual agents. This reflects the situation that belief or desire of individual agents are often suppressed in a society. The existence of a single agent which is self-interested and inconsistent, would eliminate social extensions.

Proposition 2.2 *Let $\mathcal{S}$ be an SDT such that some Δ_i in $\mathcal{S}$ is inconsistent and contains no social default rule. Then $\mathcal{S}$ has no social extension.*

To make an agreement, some agents can form a *party* by excluding those who have no interaction with them. This is also effective to isolate agents who are self-interested and inconsistent. For an SDT $\mathcal{S}=(\Delta_1,\ldots,\Delta_m,\Gamma)$, let $\mathcal{S}_P=(\Delta_1,\ldots,\Delta_l,\Gamma')$ $(l\le m)$ where every default rule in Δ_i $(1\le i\le l)$ and Γ' contains no formula from $\mathcal{F}_k$ $(l+1\le k\le m)$.

Proposition 2.3 *If an SDT $\mathcal{S}$ has a social extension E, $\mathcal{S}_P$ has a social extension G such that $E\cap(\mathcal{F}_1\cup\cdots\cup\mathcal{F}_l)=G$.*

The converse of Proposition 2.3 does not hold in general.

3 Social Reasoning by SDT

3.1 Cooperative Planning

In cooperative planning, multiple agents are supposed to have a common goal to be accomplished, and they build a joint plan by working cooperatively.

Definition 3.1 (cooperative planning framework) A *cooperative planning framework* is defined as a tuple $(\mathcal{S}, \mathcal{A}, \omega)$, where $\mathcal{S} = (\Delta_1, \ldots, \Delta_m, \Gamma)$ is an SDT, $\mathcal{A} \subseteq \mathcal{F}_1 \cup \cdots \cup \mathcal{F}_m$ is a set of *actions*, and $\omega \in \mathcal{F}$ is a *goal*. Given $(\mathcal{S}, \mathcal{A}, \omega)$, let $\Gamma^\omega = \Gamma \cup \{ \frac{:\neg\omega}{false} \}$. Then, a set $\Phi \subseteq \mathcal{A}$ is a *solution* of a cooperative planning framework $(\mathcal{S}, \mathcal{A}, \omega)$ if $\bigcup_{i=1}^{m} \Delta_i \cup \Gamma^\omega$ has an extension E such that $\Phi = E \cap \mathcal{A}$.

In cooperative planning, a common goal is given as a constraint to be satisfied in a society, and each agent computes their role of actions to achieve the goal.

Example 3.1 A robot 1 has a blue block and another robot 2 has a red block. There is a yellow block on the floor. The goal is to put the blue block on the yellow one, and to put the red block on the blue one. To achieve the goal, these two robots make a cooperative plan. Each robot can sense the change of the block world. The situation is represented by the cooperative planning framework $(\mathcal{S}, \mathcal{A}, \omega)$ with $\mathcal{S} = (\Delta_1, \Delta_2, \Gamma)$ where

$$\Delta_1 = \{ \frac{X\_on\_Y^1_{(T)} \wedge Z\_to\_X^1_{(T)} : \neg W\_to\_X^2_{(T)}, Z\_on\_X^1_{(T+1)}}{Z\_on\_X^1_{(T+1)}}, \frac{has\_X^1_{(T)} : X\_to\_Y^1_{(T)}}{X\_to\_Y^1_{(T)}}, \frac{X\_on\_Y^2_{(T)} : X\_on\_Y^1_{(T)}}{X\_on\_Y^1_{(T)}}, \frac{X\_on\_Y^1_{(T)} : X\_on\_Y^1_{(T+1)}}{X\_on\_Y^1_{(T+1)}}, has\_blue^1_{(0)}, yellow\_on\_floor^1_{(0)} \}.$$

Here, uppercase letters represent variables which are shorthand of their instances, and (T) means time steps. In Δ_1 the first rule represents that if a block X is on Y and the robot 1 moves another block Z to the location of X, and if it is assumed that the robot 2 does not move another block W to the location of X, the block Z is normally put on the block X at the next time step. The second rule says if a robot 1 has a block X then the robot can take an action of moving X to the location of Y. The third rule represents that if a block X is put on a block Y by the robot 2, the robot 1 can recognize the situation. The fourth rule represents the inertial rule: if a block X is on Y at time T and it is consistent to assume the existence of X on Y at time $T+1$, it is indeed at that location. The fifth fact represents that the robot 1 has a blue block and the sixth fact represents that the yellow block is on the floor at the time 0. Δ_2 has default rules similar to Δ_1 such that agent identifiers 1 and 2 of Δ_1 are exchanged, and Δ_2 has the fact $has\_red^2_{(0)}$ instead of $has\_blue^1_{(0)}$ in Δ_1. Γ is used for specifying *state constraints* in planning, but here we put $\Gamma = \emptyset$ for simplicity.

The set of actions is put $\mathcal{A} = \{ X\_to\_Y^1_{(T)}, X\_to\_Y^2_{(T)} \}$ where variables represent their instances. If the goal ω is to be achieved at time 3, it is represented as $\Gamma^\omega = \{ \frac{: \neg (red\_on\_blue^1_{(3)} \wedge blue\_on\_yellow^1_{(3)} \wedge red\_on\_blue^2_{(3)} \wedge blue\_on\_yellow^2_{(3)})}{false} \}$, which states that two robots recognize the goal to be accomplished at time 3: red block is on the blue one which is on the yellow one. A solution of a plan then becomes $\Phi = \{ blue\_to\_yellow^1_{(1)}, red\_to\_blue^2_{(2)} \}$, which represents that the robot 1 moves the blue block to the location of the yellow block at time 1, and the robot 2 moves the red block to the location of the blue block at time 2.

3.2 Negotiation

In negotiation, individual agents have their own goals and make a deal to accomplish them. Here we consider negotiation between two agents.

Definition 3.2 (negotiation framework) A (one-to-one) *negotiation framework* is defined as a tuple $(\mathcal{S}, \omega_1, \omega_2)$, where $\mathcal{S} = (\Delta_1, \Delta_2, \Gamma)$ is an SDT, $\omega_1 \in \mathcal{F}_1$ and $\omega_2 \in \mathcal{F}_2$ are *goals* of agents 1 and 2, respectively. Put $\Delta_i^{\omega} = \Delta_i \cup \{\frac{:\neg\omega_i}{false}\}$.

In contrast to cooperative planning, each agent (or at least one of two agents) has its own goal in a negotiation framework. If an agent i has no goal, put $\omega_i = true$. Two agents negotiate with each other to achieve their own goals. If a *proposal* is made by an agent, the opponent agent decides whether it is acceptable or not. If it is unacceptable, the opponent tries to make a *counter-proposal*. A negotiation proceeds by exchanging and evaluating mutual proposals until it reaches a (dis)agreement. Suppose two agents Ag_1 and Ag_2, and its negotiation framework $(\mathcal{S}, \omega_1, \omega_2)$. When Ag_1 has its goal, a negotiation proceeds according to the following protocol.

1. If Δ_1^{ω} is inconsistent and $\Delta_1^{\omega} \cup \{\phi\}$ has a consistent extension for some $\phi \in \mathcal{F}_2$, Ag_1 makes a proposal ϕ to Ag_2.
2. If Δ_2 has an extension E such that $E \cup \{\phi\}$ is consistent, Ag_2 returns the proposition $accept_\phi$ to Ag_1. Else if for some $\psi \in \mathcal{F}_1$, $\Delta_2 \cup \{\psi\}$ has an extension G such that $G \cup \{\phi\}$ is consistent, Ag_2 returns a counter-proposal ψ to Ag_1. Otherwise, Ag_2 returns the proposition $reject_\phi$ to Ag_1.
3. If the response made by Ag_2 is $accept_\phi$, the negotiation ends in success. Else if the response is $reject_\phi$, Ag_1 seeks another condition $\phi' \in \mathcal{F}_2$ to satisfy Step 1. If any condition exists, repeat Step 2; if nothing exists, the negotiation ends in failure. Otherwise, repeat Step 2 for evaluating ψ in Δ_1.
4. Iterate Step 2 and Step 3 until one of the agents gets a response of $accept_\phi$ for some ϕ (negotiation succeeds) or $reject_\phi$ for any ϕ (negotiation fails).

Example 3.2 A buyer agent 1 wants to buy a PC with a discount price. She has no cash and wants to pay by card. A seller agent 2 sells a PC with a normal price, but a discount price is applied if the buyer pays by cash or accepts a used PC. The situation is represented by a negotiation framework $(\mathcal{S}, \omega_1, \omega_2)$ where

$$\Delta_1 = \{\, \neg pay\_cash^1,\ \ \frac{discount^2:}{buy^1} \,\},$$

$$\Delta_2 = \{\, \frac{:normal^2}{normal^2},\ \frac{:\neg discount^2}{\neg discount^2},\ \frac{discount^2:}{\neg normal^2},\ \frac{pay\_cash^1:discount^2}{discount^2},\ \frac{used\_pc^1:discount^2}{discount^2} \,\}.$$

The buyer has the goal $\omega_1 = buy^1$, then the agent 1 starts negotiation. As $\Delta_1^{\omega} \cup \{\, discount^2 \,\}$ has a consistent extension, the agent 1 proposes $\phi = discount^2$ to the seller. As Δ_2 has no extension which is consistent with $\{\phi\}$, the seller cannot accept ϕ as it is. The agent 2 then seeks a condition to accept it. Since $\Delta_2 \cup \{pay\_cash^1\}$ has an extension which is consistent with $\{\phi\}$, the seller returns the counter-proposal $\psi = pay\_cash^1$ to the agent 1. The buyer does not accept the proposal ψ because $\Delta_1 \cup \{\psi\}$ is inconsistent. Then, the agent 1 returns $reject_\psi$ to the agent 2. In return to this, the agent 2 seeks another condition and finds $\psi' = used\_pc^1$ as a counter-proposal. As Δ_1 has an extension which is consistent with $\{\psi'\}$, the buyer accepts the proposal ψ' and sends $accept_{\psi'}$ to the agent 2. Then, negotiation succeeds.

4 Related Work

There are some work studying default logic in distributed environments. Baral et al. [1] consider the problem of combining multiple default theories. Given a set $\{\Delta_1, \ldots, \Delta_n\}$ of default theories and a set IC of integrity constraints as first-order formulas, they compute maximal subsets of the combination $\Delta_1 \cup \cdots \cup \Delta_n$ which are consistent with IC. Their goal is to resolve inconsistencies that may arise by combining different default theories. Ryzko and Rybinski [7] introduce *distributed default logic* to realize distributed problem solving in multiagent systems. The purpose is computing extensions of a default theory by its partitions using a distributed algorithm. Brewka et al. [2] introduce *contextual default systems* (CDS) as a tuple $(\Delta_1, \ldots, \Delta_n)$ of contextual default theories. Here, contextual default theories corresponds to default theories of our Definition 2.1. Compared with SDT, CDS has no notion of social constraints in its syntax. In semantics, *contextual extensions* of CDS are defined as tuples of extensions of individual theories, which is different from social extensions in SDT. Moreover, CDS does not exhibit its application to cooperative planning or negotiation. Sakama [8] uses default logic to represent weak belief of an agent, which could be abandoned during negotiation. He uses *super normal defaults* of the form $\frac{:\gamma}{\gamma}$ for this purpose, but does not use other types of default rules in negotiation. Buccafurri and Caminiti [3] introduce a *social logic program* (SOLP) which has rules of the form: $head \leftarrow [selection\_condition]\{body\}$, where $selection\_condition$ specifies social conditions concerning either the cardinality of communities or particular individuals satisfying the body. SDT is different from SOLP in both language and semantics. In particular, SOLP does not have social constraints which describe regulations in a society. Using the connection to logic programming [4], a subclass of SDT is represented by a collection of logic programs and social extensions are computed as *answer sets* of such programs. A number of studies provide logical frameworks for negotiation or cooperative planning. The purpose of this paper is not only formulating particular social reasoning, but providing a logic for representing and reasoning about social attitudes of multiple agents.

References

1. C. Baral, S. Kraus, J. Minker, and V. S. Subrahmanian. Combining default logic databases. *J. Cooperative Information Systems* 3: 319–348, 1994.
2. G. Brewka, F. Roelofsen, and L. Serafini. Contextual Default Reasoning. In: *Proc. IJCAI-07*, pp. 268–273, 2007.
3. F. Buccafurri and G. Caminiti. A social semantics for multi-agent systems. In: *Proc. LPNMR'05*, LNAI 3662, pp. 317–329, 2005.
4. M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–385, 1991.
5. M. Gelfond, V. Lifschitz, H. Przymusinska, and M. Truszczynski. Disjunctive defaults. In: *Proc. KR'91*, pp. 230–237, 1991.
6. R. Reiter. A logic for default reasoning. *Artificial Intelligence* 13:81–132, 1980.
7. D. Ryzko and H. Rybinski. Distributed default logic for multi-agent system. In: *Proc. Int'l Conf. Intelligent Agent Technology*, pp. 204–210, 2006.
8. C. Sakama. Inductive negotiation in answer set programming. In: *Proc. 6th Int'l Wksp. Declarative Agent Languages and Technologies*, LNAI 5397, pp. 143–160, 2008.