A Defeasible Deontic Model for Intelligent Simulation

Kazumi Nakamatsu

School of H.S.E., Univ. Hyogo, HIMEJI 670-0092 Japan

Abstract. We introduce an intelligent drivers' model for traffic simulation in a small area including some intersections, which is formalized in a paraconsistent annotated logic program EVALPSN. The intelligent drivers' model can infer drivers' speed control actions such as "slow down" based on EVALPSN defeasible deontic reasoning and deal with minute speed change of cars in the simulation system.

1 Introduction

We have already developed EVALPSN(Extended Vector Annotated Logic Program) [4, 5] that can deal with defeasible deontic reasoning [10], and applied it to various kinds of control such as traffic signal control [9] and robot action control [7, 8, 6]. In order to evaluate the traffic signal control [9], we made a traffic simulation system based on the cellular automaton method that simulates each car movement around a few intersections. Basically, in the cellular automaton method, roads are divided into many cells and each cell is supposed to have one car, and car movement is simulated based on a simple cell transition rule such that "if the next cell is vacant, the car has to move into the next cell". Therefore, it does not seem that the usual cellular automaton method can simulate each car movement minutely, even though it has many other advantages such as it does not cost long time for traffic simulation.

In this paper, we introduce an intelligent drivers' model to infer drivers' speed control by EVALPSN defeasible deontic reasoning, which can be used for simulating each car movement minutely in the traffic simulation system.

Generally, car speed control actions by human being such as putting brake to slow down the car can be regarded as the result of defeasible deontic reasoning to resolve conflicts. For example, if you are driving a car, you may catch conflicting informations "there is enough distance from your car to the precedent car for speeding up your car" and "I am driving the car at the speed limit". The first information derives permission for the action "speed up" and the second one derives forbiddance from it. Then the forbiddance defeats the permission and you may not speed up your car. On the other hand, if you catch the information "I am driving the car at much less than the speed limit" as the second one, then this information derives permission for speeding up your car and you may speed up your car. Therefore, as shown in the example, human being decision making for action control can be done by defeasible deontic reasoning with some rules such as traffic rules We formalize such a defeasible deontic model for car speed control action in the paraconsistent logic program EVALPSN and introduce a traffic simulation system based on the drivers' model.

Nakamatsu K. (2006).
A Defeasible Deontic Model for Intelligent Simulation.
In *Proceedings of the 3rd International Workshop on Computer Supported Activity Coordination*, pages 35-44
DOI: 10.5220/0002486500350044

This paper is organized as follows : first, we review EVALPSN briefly and introduce defeasible deontic reasoning for the drivers' model in EVALPSN ; next, we describe some sample drivers' rules to control car speed and how those rules are translated into EVALPSN cluases ; and introduce the traffic simulation system based on the EVALPSN drivers' model.

2 EVALPSN

Generally, a truth value called an *annotation* is explicitly attached to each literal in annotated logic programs [1]. For example, let p be a literal, μ an annotation, then $p:\mu$ is called an *annotated literal*. The set of annotations constitutes a complete lattice. An annotation in VALPSN [3] that can deal with defeasible reasoning is a 2-dimensional vector called a *vector annotation* such that each component is a non-negative integer and the complete lattice $\mathcal{T}_v$ of vector annotations is defined :

$$\mathcal{T}_v = \{\ (x,y)\,|\,0 \le x \le n,\ 0 \le y \le n,\ x,y \text{ and } n \text{ are non-negative integers }\}.$$

The ordering of the lattice $\mathcal{T}_v$ is denoted by a symbol $\preceq_v$ and defined : let $\boldsymbol{v_1} = (x_1, y_1) \in \mathcal{T}_v$ and $\boldsymbol{v_2} = (x_2, y_2) \in \mathcal{T}_v$,

$$\boldsymbol{v_1} \preceq_v \boldsymbol{v_2} \iff x_1 \le x_2 \text{ and } y_1 \le y_2.$$

For each vector annotated literal $p:(i,j)$, the first component i of the vector annotation denotes the amount of positive information to support the literal p and the second one j denotes that of negative information. For example, a vector annotated literal $p:(2,1)$ can be intuitively interpreted that the literal p is known to be true of strength 2 and false of strength 1. In order to deal with defeasible deontic reasoning we have extended VALPSN to EVALPSN. An annotation in EVALPSN called an *extended vector annotation* has a form of $[(i,j),\mu]$ such that the first component (i,j) is a 2-dimentional vector as a vector annotation in VALPSN and the second one,

$$\mu \in \mathcal{T}_d = \{\bot, \alpha, \beta, \gamma, *_1, *_2, *_3, \top\},$$

is an index that represents deontic notion or paraconsistency. The complete lattice $\mathcal{T}_e$ of extended vector annotations is defined as the product $\mathcal{T}_v \times \mathcal{T}_d$. The ordering of the lattice $\mathcal{T}_d$ is denoted by a symbol $\preceq_d$ and described by the Hasse's diagrams in **Fig.1**. The intuitive meaning of each member in the lattice $\mathcal{T}_d$ is ; $\bot$ (unknown), α (fact), β (obligation), γ (non-obligation), $*_1$ (both fact and obligation), $*_2$ (both obligation and non-obligation), $*_3$ (both fact and non-obligation) and $\top$ (paraconsistency). Therefore, EVALPSN can deal with not only paraconsistency between usual knowledge but also between permission and forbiddance, obligation and forbiddance, and fact and forbiddance. The Hasse's diagram(cube) shows that the lattice $\mathcal{T}_d$ is a tri-lattice in which the direction $\overrightarrow{\gamma\beta}$ represents *deontic truth*, the direction $\overrightarrow{\bot *_2}$ represents the amount of *deontic knowledge* and the direction $\overrightarrow{\bot\alpha}$ represents *factuality*. Therefore, for example, the annotation β can be intuitively interpreted to be deontically truer than the annotation

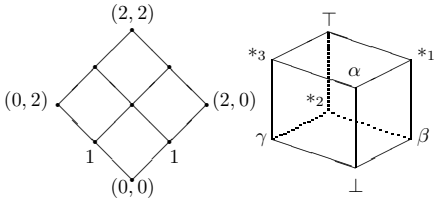

Fig. 1. Lattice $\mathcal{T}_v(n = 2)$ and Lattice $\mathcal{T}_d$.

γ and the annotations $\bot$ and $*_2$ are deontically neutral, i.e., neither obligation nor not-obligation. The ordering over the lattice $\mathcal{T}_e$ is denoted by a symbol $\preceq_e$ and defined as : let $[(i_1, j_1), \mu_1]$ and $[(i_2, j_2), \mu_2]$ be extended vector annotations,

$$[(i_1, j_1), \mu_1] \preceq_e [(i_2, j_2), \mu_2] \Leftrightarrow (i_1, j_1) \preceq_v (i_2, j_2) \quad \text{and} \quad \mu_1 \preceq_d \mu_2.$$

There are two sorts of epistemic negations $\neg_1$ and $\neg_2$ in EVALPSN, which are defined as mappings over $\mathcal{T}_v$ and $\mathcal{T}_d$, respectively.
Definition 1 (Epistemic Negations, $\neg_1$ and $\neg_2$)

$$\begin{aligned}
&\neg_1([(i,j),\mu]) = [(j,i),\mu], && \forall \mu \in \mathcal{T}_d,\\
&\neg_2([(i,j),\bot]) = [(i,j),\bot], && \neg_2([(i,j),\alpha]) = [(i,j),\alpha],\\
&\neg_2([(i,j),\beta]) = [(i,j),\gamma], && \neg_2([(i,j),\gamma]) = [(i,j),\beta],\\
&\neg_2([(i,j),*_1]) = [(i,j),*_3], && \neg_2([(i,j),*_2]) = [(i,j),*_2],\\
&\neg_2([(i,j),*_3]) = [(i,j),*_1], && \neg_2([(i,j),\top]) = [(i,j),\top].
\end{aligned}$$

These epistemic negations, $\neg_1$ and $\neg_2$, can be eliminated by the above syntactic operation. On the other hand, the ontological negation(strong negation $\sim$) [2] in EVALPSN can be defined by the epistemic negations, $\neg_1$ or $\neg_2$, and interpreted as classical negation.
Definition 2 (Strong Negation) [2] Let F be a formula and $\neg$ be $\neg_1$ or $\neg_2$.

$$\sim F =_{def} F \rightarrow ((F \rightarrow F) \wedge \neg(F \rightarrow F)).$$

Definition 3 (well extended vector annotated literal) Let p be a literal. $p : [(i, 0), \mu]$ and $p : [(0, j), \mu]$ are called *well extended vector annotated literals*(weva-literals for short), where $i, j \in \{1, 2, \cdots\}$, and $\mu \in \{\ \alpha,\ \beta,\ \gamma\ \}$.
Definition 4 (EVALPSN) If $L_0, \cdots, L_n$ are weva-literals,

$$L_1 \wedge \cdots \wedge L_i \wedge \sim L_{i+1} \wedge \cdots \wedge \sim L_n \rightarrow L_0$$

is called an *Extended Vector Annotated Logic Program clause with Strong Negation* (EVALPSN clause for short). An *Extended Vector Annotated Logic Program with Strong Negation* is a finite set of EVALPSN clauses.
Note : if an EVALPSN or an EVALPSN clause contain no strong negation, they may be just called an EVALP or an EVALP clause, respectively.

Deontic notions and fact are represented by extended vector annotations in EVALPSN as follows :

– "fact of strength m", "obligation of strength m", "forbiddance of strength m" and "permission of strength m" are represented by extended vector annotations $[(m, 0), \alpha]$, $[(m, 0), \beta]$, $[(0, m), \beta]$ and $[(0, m), \gamma]$, respectively, where m is a positive integer.

Therefore, for example, a weva-literal $p : [(2, 0), \alpha]$ can be intuitively interpreted as "it is known that the literal p is a fact of strength 2", and a weva-literal $q : [(0, 1), \beta]$ can be intuitively interpreted as "the literal q is forbidden of strength 1".

3 Defeasible Deontic Drivers' Model

Suppose that a man is driving a car. Then, how does the car driver decide the next action for controlling car speed such as braking or acceleration ? It is easily supposed that, for example, if the traffic light in front of the car is red, the driver has to slow down the car, or if there is enough distance from the driver's car to the precedent car, the driver may speed up the car. If we model such drivers' car speed control, we should consider conflicting informations such as "traffic light is red" and "enough distance to speed up", and its conflict resolving. It also should be considered that car drivers reason car speed control based on not only detected physical information such as the current car speed but also traffic rules such as "keep driving at less than speed limit". For example, if a driver is driving a car over the speed limit of the road, the driver would slow down the car even if there is no car ahead of the car, then, it is supposed that there exists strong forbiddance from driving over the speed limit, and eventually it may turn into obligation to slow down the car. On the other hand, if a driver is driving a car at very slow speed, the driver would speed up the car even if the traffic light far ahead of the car is red, then, it is also supposed that there exist both strong permission and weak forbiddance to speed up the car, then only the permission is obtained by defeasible deontic reasoning, and eventually it may turn into obligation to speed up the car. Therefore, we can easily model such drivers' decision making for car speed control by EVALPSN defeasible deontic reasoning as described in the above example. In this section, we introduce the EVALPSN drivers' model that can derive the three car speed control actions, "slow down", "speed up", or "keep the current speed" in EVALPSN programming. We define the drivers' model in the following subsections.

3.1 Framework for EVALPSN Drivers' Model

1. Forbiddance or permission for the car speed control action, "speed up" are derived based on the traffic rules,
   – it is obligatory to obey traffic signal,
   – it is obligatory to keep the speed limit, etc.,

   and the following detected information,
   – the object car speed,
   – the precedent car speed,
   – the distance between the precedent and objective cars,
   – the distance to the intersection or the curve ahead of the objective car ;

2. obligation for one of the three car speed control actions, “speed up”,“slow down” and “continue the current speed” is derived by defeasible deontic reasoning in EVALPSN programming ;
3. basically, a similar method to the cellular automaton method is used as the traffic simulation method.

3.2 Annotated Literals

In the EVALPSN drivers’ model, the following annotated literals are used to represent various information,

$mv(t)$ represents one of the three car actions, “speed up”, “slow down”, or “keep the current speed” at the time t ; this predicate has the complete lattice $\mathcal{T}_v$ of vector annotations,

$$\begin{aligned}\mathcal{T}_v = \{ \quad & (0,0)\text{“no information”} , \quad (1,0)\text{“weak speed up”} , \\ & (0,1)\text{“weak slow down”} , \quad (2,0)\text{“strong speed up”} , \\ & (0,2)\text{“strong slow down”}, \quad \cdots , \ (2,2) \quad \},\end{aligned}$$

for example, if we have the EVALP clause $mv(t) : [(0,1), \beta]$, it represents the weak forbiddance from the action “speed up” at the time t, on the other hand, if it has the annotation $[(2,0), \gamma]$, it represents the strong permission for the action “slow down”, etc. ;

$v_o(t)$ represents the speed of the objective car at the time t, then we suppose the complete lattice of vector annotations for representing the objective car speed,

$$\mathcal{T}_v = \{(i,j) | i,j \in \{0,1,2,3,4,5\}\},$$

we may have the following informal interpretation, if we have the EVALP clause $v_o(t) : [(2,0), \alpha]$, it represents that the car is moving forward at the speed of over 20km/h at the time t, on the other hand, if we have the EVALP clause $v_o(t) : [(0,1), \alpha]$, it represents that the car is moving backward at the speed of over 10km/h at the time t, etc. ;

$v_n(t)$ represents the speed of the precedent car at the time t ; the complete lattice lattice structure and informal interpretation of the vector annotations are the same as the case of the predicate $v_o(t)$;

$d_p(t)$ represents the distance between the precedent and the objective cars at the time t ; the complete lattice of vector annotations for representing the distance,

$$\mathcal{T}_v = \{(i,j) | i,j \in \{0,1,2,\ldots,n\}\},$$

if we have the EVALP clause $d_p(t) : [(2,0), \alpha]$, it represents that the distance is more than 2 cells at the time t, moreover, if we have the EVALP clause $d_p(t) : [(5,0), \beta]$, it represents that the distance has to be more than 5 cells at the time t, on the other hand, if we have the EVALP clause $d_p(t) : [(0,3), \beta]$, it represents that the distance must not be more than 3 cells at the time t, etc. ;

$d_c(t)$ represents the distance from the objective car to the curve in front of the car at the time t ; the complete lattice structure and informal interpretation of the vector annotations are the same as the case of the predicate $d_p(t)$;

$go(t)$ represents the direction where the objective car turns to at the time t ; this predicate has the complete lattice of vector annotations,

$$\begin{aligned}\mathcal{T}_v = \{\ & (0,0)\text{“no information”}, && (1,0)\text{“right turn”}\ ,\\ & (0,1)\text{“left turn”}\ , && (2,0)\text{“right turn”}\ ,\\ & (0,2)\text{“left turn”}\ , && \cdots,\ (2,2)\quad \},\end{aligned}$$

if we have the EVALP clause $go(t) : [(2,0), \alpha]$, it represents that the car turns to the right at the time t, if we have the EVALP clause $go(t) : [(0,2), \beta]$, it represents that the car must not turn to the right, that is to say, must turn to the right, etc..

3.3 Inference Rules

We also have some inference rules to derive the next car control action in the EVALPSN drivers' model and introduce the basic three inference rules, **Traffic Signal Rule**, **Straight Road Rule** and **Curve and Turn Rule**. We suppose that there is a cross intersection with a traffic light in front of the objective car in the following rules.

Traffic Signal Rule If the traffic light indicates

- **red**, it is considered as there is an obstacle on the stop line before the traffic light, that is to say, there is strong forbiddance from entering into the intersection ;
- **yellow**, it is considered as the same as the red light rule except that if the distance between the car and the stop line is less than 2 cells, it is weakly permitted for entering into the intersection ;
- **green**, it has no forbiddance from going into the intersection except that if the car turning at the intersection, it is described in **Curve and Turn Rule**.

Straight Road Rule If the road is straight, the objective car behavior is inferred by

- distance between the precedent car and the objective car ;
- each speed of the precedent car and the objective car ;
- obeying the traffic rule, speed limit of roads and traffic signal, etc..

Curve and Turn Rule If the objective car is headed to the curve or going to turn at the intersection, forbiddance to speed up the car is derived.

Basic idea of the EVALPSN Drivers' model based simulation is as follows : as the first step, forbiddance or permission for one of the car actions, “speed up” or “slow down”, are derived by EVALPSN defeasible deontic reasoning ; as the next step, if the forbiddance for a car action is derived, the objective car has to do the opposite action ; if the permission for a car action is derived, the objective car has to do the action ; if neither forbiddance nor permission is derived, the objective car does not have to do any action, that is to say, it has to keep the current speed. We show an example for the EVALPSN drivers' model.

3.4 Example

suppose that the objective car is moving at the speed of 1, then we have the following EVALP clauses to reason the next action of the objective car according to the current information.

Case 1 If the distance between the precedent car and the objective car is longer than 2 cells, we have permission to accelerate the car at the time t. This rule is translated into the EVALP clause,

$$v_o(t):[(1,0),\alpha] \wedge d_p(t):[(2,0),\alpha] \rightarrow mv(t):[(0,1),\gamma]. \quad (1)$$

Case 2 If the precedent car stopped at the next cell and the objective car is moving at the speed of 1, we have strong forbiddance from speed up at the time t, which means strong obligation to slow down. This rule is translated into the EVALP clause,

$$v_o(t):[(1,0),\alpha] \wedge v_n(t):[(0,0),\alpha] \wedge d_p(t):[(0,0),\alpha] \\ \rightarrow mv(t):[(0,2),\beta]. \quad (2)$$

Case 3 If the precedent car is faster than the objective car whose speed is 1, we have permission to accelerate the objective car at the time t. This rule is translated into the EVALP clause,

$$v_o(t):[(1,0),\alpha] \wedge v_n(t):[(2,0),\alpha] \rightarrow mv(t):[(0,1),\gamma]. \quad (3)$$

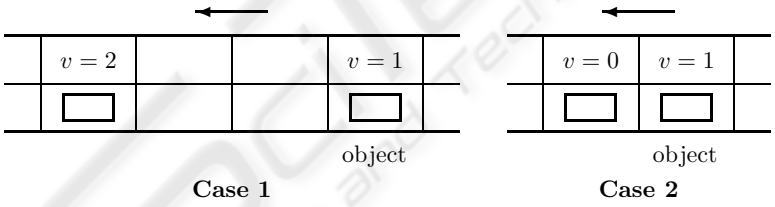

Fig. 2. Cell States in the **Case 1** and **2**.

Case 4 If the car is moving at the speed of 3 and the distance between the car and the curve is 2 cells at the time t. This rule is translated into :

$$v_o(t):[(3,0),\alpha] \wedge d_c(t):[(2,0),\alpha] \wedge go(t):[(2,0),\alpha] \\ \rightarrow mv(t):[(0,1),\beta] \quad (4)$$

If both the permission $mv(t):[(0,1),\gamma]$ and the forbiddance $mv(t):[(0,2),\beta]$ from speed up are derived, we have obligation to slow down the objective car at the next step by defeasible deontic reasoning, since the forbiddance is stronger than the permission.

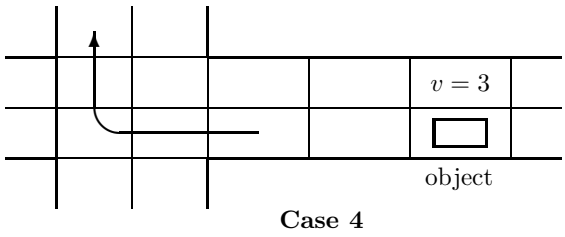

Fig. 3. Cell States in the **Case 4**.

3.5 Traffic Signal Simulation System

Fig. 4 shows the drivers' model based traffic signal simulation around a typical cross intersection with traffic lights. In the figure, each square box with an integer 0 to 4 indicates a car, and the integer indicates its speed at that time. For example, if the integer 2 is attached to the car, it indicates that the car is moving at the speed of 20km/h. In the traffic signal simulation based on the drivers' model, one of the three car control actions is reasoned for each car. For example, suppose that the objective car is moving at the speed of 20km/h at the time t, then if the drivers' model reasons defeasibly the action "slow down" as obligation, the objective car will be moving at the speed of 10km/h at the next time $t + 1$. Moreover, the simulation system can simulate the traffic signal control based on EVALPSN defeasible deontic reasoning [9] in which the length of each traffic light (red, yellow, green, etc.) is controlled by EVALPSN programming.

Comparing to usual cellular automaton model based simulation, although the drivers' model based simulation can simulate each car movement more minutely, it costs much time to reason each car speed due to EVALPSN programming time for each car in the simulation stage. Therefore, if we implement large scale simulation based on the drivers' model, it should be necessary to consider simulation time reduction.

4 Conclusion

In this paper, we have introduced an intelligent drivers' model based on EVALP- SN as one applicable example of EVALPSN defeasible deontic model. We also have a similar problem in terms of developing a precise simulation system in railway train operation. Actually, we have already developed train operators' model as another applicable example of EVALPSN defeasible deontic model and developed a railway train simulation system, which is utilized for simulating train speed presicely and recovering delayed train schedules. We could not construct the precise railway operation simulator without the train operators' model based on EVALPSN defeasible deontic reasoning. Although the drivers' model based simulation can simulate each car movement minutely, it costs much time to reason each car speed. Therefore, it is necessary to consider simulation time reduction when the drivers' model is implemented.

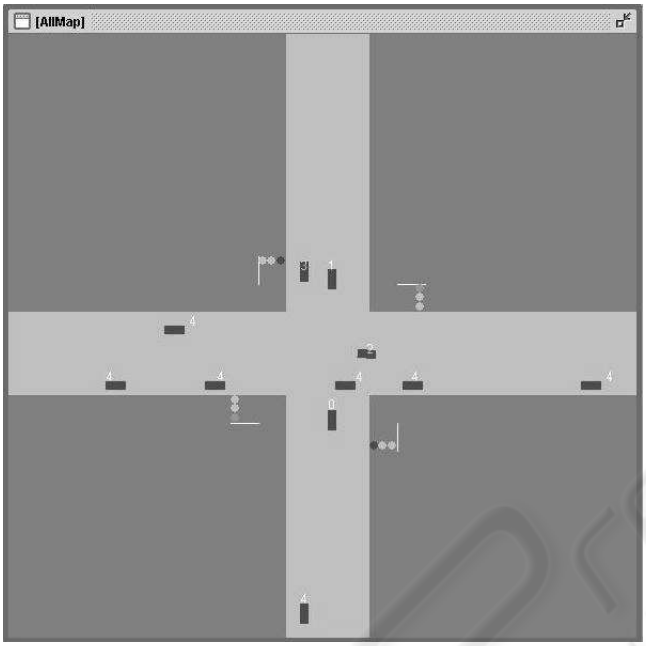

Fig. 4. Traffic Simulation at Intersection.

References

1. Blair,H.A. and Subrahmanian,V.S., Paraconsistent Logic Programming, *Theoretical Computer Science*, **68**, pp.135-154, 1989.
2. da Costa,N.C.A., Subrahmanian,V.S., and Vago,C., The Paraconsistent Logics P$\mathcal{T}$, *Zeitschrift für Mathematische Logic und Grundlangen der Mathematik*, **37**, pp.139–148, 1991.
3. Nakamatsu,K., Abe,J.M., and Suzuki,A., Defeasible Reasoning Between Conflicting Agents Based on VALPSN, *Proc. AAAI Workshop Agents' Conflicts*, pp.20–27, AAAI Press, 1999.
4. Nakamatsu,K., Abe,J.M., and Suzuki,A., A Defeasible Deontic Reasoning System Based on Annotated Logic Programming, *Proc. the Fourth International Conference on Computing Anticipatory Systems*, AIP Conference Proceedings **573**, pp.609–620, AIP, 2001.
5. Nakamatsu,K., Abe,J.M., and Suzuki,A., Annotated Semantics for Defeasible Deontic Reasoning, *Proc. the Second International Conference on Rough Sets and Current Trends in Computing*, LNAI **2005**, pp.432–440, Springer-Verlag, 2001.
6. Nakamatsu,K., Abe,J.M., and Suzuki,A., Defeasible Deontic Robot Control Based on Extended Vector Annotated Logic Programming, *Proc. the Fifth International Conference on Computing Anticipatory Systems*, AIP Conference Proceedings **627**, pp.490–500, AIP, 2002.
7. Nakamatsu,K., Mita,Y., and Shibata,T., "An Intelligent Action Control System Based on Extended Vector Annotated Logic Program and its Hardware Implementation", *Intelligent Automation and Soft Computing*, **12**, TSI Press, 2006 (to appear).
8. Nakamatsu,K., Mita,Y., Shibata,T., and Abe,J.M., Defeasible Deontic Action Control Based on Paraconsistent Logic Program and its Hardware Implementation, *Proc. 3rd International*

Conference on Computational Intelligence for Modelling Control and Automation (CD-ROM), 2003.
9. Nakamatsu,K., Seno,T., Abe,J.M., and Suzuki,A., "Intelligent Real-time Traffic Signal Control Based on a Paraconsistent Logic Program EVALP", *Proc. the 9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, LNCS **2639**, pp.719–723, Springer-Verlag, 2003.
10. Nute,D.(ed.) *Defeasible Deontic Reasoning*, Synthese Library, **263**, Kluwer Academic Publishers, 1997.