

EVALPに基づく交通信号制御

中松 和巳[†] 妹尾 年晃[‡] 鈴木 淳之[‡]

[†] 姫路工業大学 〒670-0092 姫路市新在家 1-1-12

[‡] 静岡大学 〒432-8011 浜松市城北 3-5-1

E-mail: [†] nakamatu@hept.himeji-tech.ac.jp, [‡] {cs9051, suzuki}@cs.inf.shizuoka.ac.jp

あらまし 不適切な交通信号制御によって引き起こされた交通渋滞は、解決すべき重大な問題である。この問題を解決するために EVALPSN と呼ばれる *paraconsistent logic program* に基づく知的実時間の交通信号制御システムを紹介する。基本的な考えは以下の通りである。センサーにより検知された交通量が入力され、*defeasible deontic reasoning* によって点灯変化の許可あるいは禁止が導出され、許可が導出されたら点灯変化を実行し、禁止が導出されたら実行しない。EVALPSN を説明した後に、日本での典型的な交差点を例にとり交通信号制御を EVALPSN でどうやって記述できるかを示す。EVALPSN に基づいた交通信号制御と固定時間の交通信号制御と両方のシミュレーション結果を載せる。

キーワード 交通信号制御, *paraconsistent logic program*, *defeasible deontic reasoning*, 知的制御

1. 導入

我々はすでに矛盾を扱うことができ、*defeasible deontic reasoning*[4, 6] できる EVALPSN (Extended Vector Annotated Logic Program with Strong Negation) という *paraconsistent logic program* を提案している。世の中の種々の状況において、競合する複数の候補がありエージェントがその中で一つを選択し意思決定する。このようなエージェントの推論を定式化したものに *defeasible* 論理がある。また、エージェントが法律、方針、規制などといった規範に従って行為を行う。このような場合、義務、禁止、許可という様相概念に基づき次になすべき行為を推論することがある。このような推論は *deontic* 論理として定式化されている。2種類の論理の組み合わせたものに *defeasible deontic* 論理があり、すでに Nute et al[10]によって研究されている。ロボットの行動制御といった EVALPSN に基づくアプリケーションがいくらかあり、鉄道の自動安全性検証や航空管制はすでに[7,8]で紹介した。本論文で我々はもう一つ別のアプリケーションについて紹介する。

不適切な交通信号機管理によって引き起こる交通渋滞は、解決すべき深刻な問題である。本論文で我々はこの問題を解決するための一つの方法として、EVALPSN に基づく知的実時間の交通信号制御システムを紹介する。もし交差点で赤信号に直面して緑信号に変わるのを待っているなら、心の中で信号が変わるように要求するだろう。この要求は信号変化に対する

許可とみなすことができる。一方、交差点を緑信号で通過しているなら、緑信号のままであってほしいと要求するはずだ。この要求は信号変化に対する禁止とみなすことができる。このとき、これらの許可、禁止の間には競合がある。交通信号制御の基本的な考え方は、EVALPSN の *defeasible deontic reasoning* を使うことによって競合を管理することができるということである。我々は EVALPSN の *defeasible deontic reasoning* によって交通信号制御を形式化する方法を示す。

本論文は次のように構成されている。始めに正式に EVALPSN について述べた後で、日本における典型的な交差点を例にとり、*defeasible deontic* 論理での交通信号制御を形式化する。次に、*defeasible deontic* 論理を EVALPSN に変換する。最後にセルラ・オートマトン交通モデルに基づく交通信号制御のシミュレーション結果を示す。

2. EVALPSN

一般に、*annotated logic program* において *annotation* と呼ばれる真理値は各 *literal* に明確につけられている。例えば、 p を *literal*、 μ を *annotation* としたとき、 $p:\mu$ は *annotated literal* と呼ぶ。*annotation* の集合は完備束を構成している。VALPSN[3,5]における *annotation* は *vector annotation* と呼ばれる2次元のベクトルである。各要素は非整数値であり、*vector annotation* の完備束 T_v は次のように定義する。

$$T_v = \{(x,y) | 0 \leq x \leq n, 0 \leq y \leq n, x,y,n \text{は整数}\} \quad (1)$$

完備束 T_v の順序関係は \leq 記号で記述され、 $\bar{v}_1 = (x_1, y_1)$ 、 $\bar{v}_2 = (x_2, y_2)$ としたとき次のように定義できる。

$$\bar{v}_1 \leq \bar{v}_2 \text{ iff } x_1 \leq x_2 \text{ かつ } y_1 \leq y_2 \quad (2)$$

vector annotated literal $p : (i, j)$ の vector annotation の第1要素 i は literal p を支持する正の情報量を表わしており、第2要素 j は負の情報量を表わしている。本論文の中では整数 n の値を2と仮定している。例えば、vector annotated literal $p : (2,1)$ は、literal p は強さ2の真であり、強さ1の偽であると知られていると解釈できる。一方、EVALPSNにおける extended vector annotation と呼ばれる annotation の形は $[(i,j), \mu]$ であり、最初の要素 (i,j) は VALPSNにおける vector annotation と同様の2次元ベクトルであり、第2要素

$$\mu \in T_d = \{\perp, \alpha, \beta, \gamma, *_1, *_2, *_3, T\} \quad (3)$$

は様相概念あるいは矛盾を表わしたものである。extended vector annotation の完備束 T は $T_v \times T_d$ として定義されている。完備束 T_d の順序関係は \leq_d 記号で記述され、図1のハッセのダイアグラムになっている。 T_d の要素の直感的な意味は、 \perp (不明)、 α (事実)、 β (義務)、 γ (非義務)、 $*_1$ (事実と義務の両方)、 $*_2$ (義務と非義務の両方)、 $*_3$ (義務と事実の両方)、 T (矛盾) である。ハッセのダイアグラム (直方体の方) は完備束 T_d が $\beta\bar{\gamma}$ 方向は様相の真理値を表わしており、 $\perp \bar{*}_2$ 方向は様相の知識の量を表わしており、 $\perp \bar{\alpha}$ 方向は事実を表わしている。従って、例えば annotation β は annotation γ よりも直感的に様相的に真であると解釈できる。また、annotation \perp と $*_2$ は様相的に中間であり、すなわち義務でも非義務でもない。完備束 T 上の順序関係は \leq 記号で記述され、 $[(i_1, j_1), \mu_1]$ と $[(i_2, j_2), \mu_2]$ を extended vector annotation としたとき次のように定義される。

$$\begin{aligned} & [(i_1, j_1), \mu_1] \leq [(i_2, j_2), \mu_2] \\ & \text{iff} \\ & (i_1, j_1) \leq_v (i_2, j_2) \text{ かつ } \mu_1 \leq_d \mu_2 \end{aligned} \quad (4)$$

EVALPSN においては認識論的否定 \neg_1 と \neg_2 の2種類があり、それぞれに T_v と T_d 上のマッピングとして定義される。

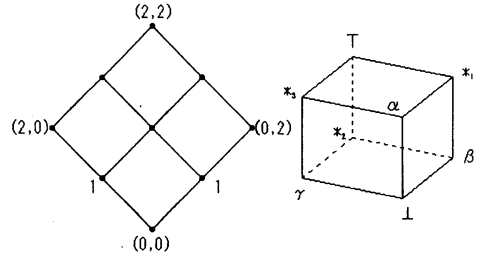


図1: 完備束 T_v 完備束 T_d

定義1 (EVALPSN の認識論的否定 \neg_1 、 \neg_2)

$$\begin{aligned} \neg_1([(i,j), \mu]) &= [(j,i), \mu], \quad \forall \mu \in T_d \\ \neg_2([(i,j), \perp]) &= [(i,j), \perp], \quad \neg_2([(i,j), \alpha]) = [(i,j), \alpha], \\ \neg_2([(i,j), \beta]) &= [(i,j), \gamma], \quad \neg_2([(i,j), \gamma]) = [(i,j), \beta], \\ \neg_2([(i,j), *_1]) &= [(i,j), *_3], \quad \neg_2([(i,j), *_2]) = [(i,j), *_2], \\ \neg_2([(i,j), *_3]) &= [(i,j), *_1], \quad \neg_2([(i,j), T]) = [(i,j), T]. \end{aligned} \quad (5)$$

extended vector annotated literal による認識論的否定 (\neg_1 、 \neg_2) は上記の構文操作によって削除できる。EVALPSN における強い否定 (\sim) は認識論的否定 \neg_1 あるいは \neg_2 によって次のように定義することができ、古典的否定として解釈できる。

定義2 (EVALPSN の強い否定 (存在論的否定))

F を論理式とする。

$$\sim F =_{\text{def}} F \rightarrow ((F \rightarrow F) \wedge \neg(F \rightarrow F)) \quad (6)$$

ここで、 \neg は \neg_1 あるいは \neg_2 である。

様相概念や事実は以下のように extended vector annotation で表現される。

- “事実” は extended vector annotation $[(m,0), \alpha]$ で表現される。
- “義務” は extended vector annotation $[(m,0), \beta]$ で表現される。
- “禁止” は extended vector annotation $[(0,m), \beta]$ で表現される。
- “許可” は extended vector annotation $[(0,m), \gamma]$ で表現される。

ここで、 $m=1$ か $m=2$ である。例えば、extended vector annotated literal $p : [(2,0), \alpha]$ は直感的に「literal p は強さ2の事実である。」と解釈できる。また、extended vector annotated literal $q : [(0,1), \beta]$ は「literal q は強さ1の禁止である」と解釈できる。

定義3(well extended vector annotated literal)

p を literal する. $p: [(i,0),\mu]$ や $p: [(0,j),\mu]$ は well extended vector annotated literal とよばれる. このとき $i, j \in \{1,2\}$ かつ $\mu \in \{\alpha, \beta, \gamma\}$ である.

定義4(EVALPSN)

もし L_0, \dots, L_n が well extended vector annotated literal ならば,

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

は extended vector annotated logic program with strong negation 節 (EVALPSN 節) とよばれる. extended vector annotated logic program with strong negation は EVALPSN 節の有限集合である. 一般に, EVALPSN はセマンティックな stable model [2] を持っていて, stable model の計算は長時間かかる. しかしながら, もし EVALPSN が階層化されれば, それは well-founded model [1] を持っている. そして, EVALPSN の strong negation は Prolog における失敗のように否定として扱うことができる. 我々は, EVALPSN は階層化されたプログラムを使用するので stable model を考慮する必要はなく, EVALPSN は普通の論理プログラムのように実装できる. EVALPSN プログラムの簡単な例を示す.

例題1 次のような EVALPSN を仮定する.

$$P = \left\{ \begin{array}{l} p: [(1,0),\alpha], \\ \sim p: [(0,3),\alpha] \rightarrow q: [(0,3),\beta] \\ \sim q: [(0,2),\beta] \rightarrow r: [(0,2),\gamma] \end{array} \right\}.$$

一般的に, annotated logic program に関して, もし literal p が $\mu \leq \lambda$ となる annotation λ への解釈によって移されているならば annotated literal $p: \mu$ は真であると評価される. そして強い否定つきの literal はいつも通り評価される. $p: [(0,3),\alpha]$ は存在しないので, $q: [(0,3),\beta]$ は導出される. しかし, $q: [(0,3),\beta]$ が存在しており, $\sim q: [(0,2),\beta]$ は保持されない. それゆえに EVALPSN P において $r: [(0,2),\gamma]$ は導出され得ない.

3. EVALPSN による交通信号制御

3.1. 予備知識

図 2 に示されている十字路の交差点を例にとって EVALPSN に基づく方法を紹介する. これは日本の交差点と仮定している (左走行). 交差点には 4 つの交通信号機 $T_{(1,2,3,4)}$ があり, 信号の色は緑色, 黄色, 赤色,

右折矢印信号の四種類ある. 交差点につながっているそれぞれの道路には交通量を検知するためのセンサーがある. それぞれのセンサーは図 2 に $S_i (1 \leq i \leq 8)$ と記述されている. 例えばセンサー S_6 は交通信号機 T_3 に直面している右折する車の交通量を検知する. 基本的に交通信号制御は交通量センサーの値に基づいて行われる. 信号変化の一連のサイクルは,

→赤色→緑色→黄色→右折矢印→全赤→

である. 本論文では, 黄色と全赤の時間を一定のものとして扱い, 黄信号は緑信号に含まれ, 全赤は右折矢印信号に含んで考える. よって, 以下の信号変化のサイクルになる.

$T_{1,2}$ →赤色→赤色→緑色→右折矢印→赤色→

$T_{3,4}$ →緑色→右折矢印→赤色→赤色→緑色→

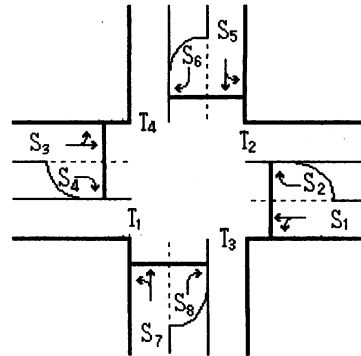


図 2: 交差点

3.2. 交通信号制御の基本的な考え方

交通信号制御の基本的な考え方を示す. 緑信号から右折矢印信号への変化と右折矢印信号から赤信号への変化だけを主として制御する. 直面の交通信号機の赤信号から緑信号への変化は, 隣接する信号機の右折矢印信号から赤信号への変化に追従する. また, 点灯時間は単一時間 $t \in \{0,1,2, \dots, n\}$ で制御される. 各道路の交通量は緑信号から右折矢印信号への点灯変化に対する許可または禁止とみなされる. 例えば, もし緑信号から右折矢印信号へ点灯変化待ちの車がたくさんいたならば, 緑信号から右折矢印信号への点灯変化に対する許可とみなされる. 一方, もし緑信号で交差点を通過している車が多くいたならば, 緑信号から右折矢印信号への点灯変化に対する禁止とみなすことができる. 従って許可と禁止の間に競合がある. EVALP の

defeasible deontic reasoning によって、もし許可が導出されたら信号変化を実行し、一方もし禁止が導出されたら信号変化を行わない。前もって各点灯色ごとに最小点灯時間と最長点灯時間を与えている。信号の各点灯時間の長さは最小点灯時間と最長点灯時間の間で制御されなければならない。

3.3. 交通信号制御における defeasible 規則と definite 規則

EVALPSN に基づく defeasible deontic 交通信号制御を明確にするために、第一に defeasible deontic 論理式で表現する。その後これを EVALPSN に変換する。

一般に、defeasible 論理には definite 規則 (\rightarrow) と defeasible 規則 (\Rightarrow) の 2 種類の規則がある。そして、競合が起きている defeasible 規則の間に優先関係があるかもしれない。definite 規則は事実に関する強い規則として扱うことができる。defeasible reasoning は defeasible 規則間の優先関係に基づいて実行される。defeasible reasoning の詳細は[9]にある。

我々は交通信号を 4 つの状態としてみなす。

- 場合 1 ($T_{1,2}$ が赤信号, $T_{3,4}$ が緑信号)
- 場合 2 ($T_{1,2}$ が赤信号, $T_{3,4}$ が右折矢印信号)
- 場合 3 ($T_{1,2}$ が緑信号, $T_{3,4}$ が赤信号)
- 場合 4 ($T_{1,2}$ が右折矢印信号, $T_{3,4}$ が赤信号)

枚数の制限のために、場合 1 のみを取り上げて EVALPSN による交通信号制御の紹介をする。次の状態を考慮して、defeasible 規則あるいは definite 規則を作る。

1. もしセンサー $S_{1,3}$ の 1 つが時刻 t で基準以上の交通量を検知したら、直面する信号機とその反対の信号機 $T_{1,2}$ の赤から緑への変化に対する許可と隣接する信号機 $T_{3,4}$ の緑から右折矢印への変化に対する許可が導出される。しかし、緑信号から右折矢印信号のみを焦点にあてているので、以下の 2 つの defeasible 規則をもつ。

$$S_{1,3}^{rg}(t) \wedge T_{1,2}(r,t) \wedge T_{3,4}(g,t) \Rightarrow \neg \circ \neg T_{3,4}(a,t) \quad (7)$$

$$S_{1,3}^g(t) \wedge T_{1,2}(r,t) \wedge T_{3,4}(g,t) \Rightarrow \neg \circ \neg T_{3,4}(a,t) \quad (8)$$

$S_1^{rg}(t)$ は時刻 t においてセンサー S_1 が基準値以上の交通量を検知したことを表わしており、 $S_3^{rg}(t)$ も同様で

ある。 $T_{1,2}(r,t)$ は信号機 $T_{1,2}$ が時刻 t において赤信号であることを表わしており、 $T_{3,4}(g,t)$ も同様である。 \circ は deontic を表現している形式的な記号であり、従って、 $\neg \circ \neg$ は許可を表わしており、 $\circ \neg$ は禁止を表わしている。

2. もしセンサー $S_{2,4}$ の 1 つが時刻 t で基準値以上の交通量を検知したら、直面する信号機とその反対の信号機 $T_{1,2}$ の赤から緑への変化に対する許可と隣接する信号機 $T_{3,4}$ の緑から右折矢印への変化に対する許可が導出される。よって、以下の 2 つの defeasible 規則をもつ。

$$S_{2,4}^{rg}(t) \wedge T_{1,2}(r,t) \wedge T_{3,4}(g,t) \Rightarrow \neg \circ \neg T_{3,4}(a,t) \quad (9)$$

$$S_{2,4}^g(t) \wedge T_{1,2}(r,t) \wedge T_{3,4}(g,t) \Rightarrow \neg \circ \neg T_{3,4}(a,t) \quad (10)$$

3. もしセンサー $S_{5,7}$ の 1 つが時刻 t で基準値以上の交通量を検知したら、隣接する信号機 $T_{1,2}$ の赤から緑への変化に対する禁止と直面する信号機とその反対の信号機 $T_{3,4}$ の緑から右折矢印への変化に対する禁止が導出される。よって、以下の 2 つの defeasible 規則をもつ。

$$S_{5,7}^{rg}(t) \wedge T_{1,2}(r,t) \wedge T_{3,4}(g,t) \Rightarrow \circ \neg T_{3,4}(a,t) \quad (11)$$

$$S_{5,7}^g(t) \wedge T_{1,2}(r,t) \wedge T_{3,4}(g,t) \Rightarrow \circ \neg T_{3,4}(a,t) \quad (12)$$

4. もしセンサー $S_{6,8}$ の 1 つが時刻 t で基準値以上の交通量を検知し、センサー $S_{7,5}$ の 1 つが時刻 t で右折を遮っていると判断できる値以上の交通量を検知したら、隣接する信号機 $T_{1,2}$ の赤から緑への変化に対する許可と直面する信号機とその反対の信号機 $T_{3,4}$ の緑から右折矢印への変化に対する許可が導出される。よって、以下の 2 つの defeasible 規則をもつ。

$$S_{6,8}^{rg}(t) \wedge S_{7,5}^{rg'}(t) \wedge T_{1,2}(r,t) \wedge T_{3,4}(g,t) \Rightarrow \neg \circ \neg T_{3,4}(a,t) \quad (13)$$

$$S_{6,8}^{rg}(t) \wedge S_{7,5}^g(t) \wedge T_{1,2}(r,t) \wedge T_{3,4}(g,t) \Rightarrow \neg \circ \neg T_{3,4}(a,t) \quad (14)$$

$S_7^{rg'}(t)$ は時刻 t におけるセンサー S_7 が基準値以上の交通量を検知したことを表わしており、 $S_5^{rg'}(t)$ も同様である。

また、交通信号を制御するために、以下の definite 規則を必要とする。

[最小点灯時間ルール]

各信号は最小点灯時間が保証される ($\text{MIN}_i(\mu, t)$, $\mu \in \{r, g, a\}$, $i \in \{1, 2, 3, 4\}$)。もし各点灯時間が時刻 t において最小点灯時間内なら、点灯変化は絶対禁

止である。以下の definite 規則をもつ。

$$MIN_{1,2}(g,t) \wedge T_{1,2}(g,t) \rightarrow \neg \neg T_{1,2}(a,t) \quad (15)$$

$$MIN_{1,2}(a,t) \wedge T_{1,2}(a,t) \rightarrow \neg \neg T_{1,2}(r,t) \quad (16)$$

$$MIN_{3,4}(g,t) \wedge T_{3,4}(g,t) \rightarrow \neg \neg T_{3,4}(a,t) \quad (17)$$

$$MIN_{3,4}(a,t) \wedge T_{3,4}(a,t) \rightarrow \neg \neg T_{3,4}(r,t) \quad (18)$$

[最長点灯時間ルール]

また、各信号は最長点灯時間によって制限される ($MAX_i(\mu,t)$, $\mu \in \{r,g,a\}$, $i \in \{1,2,3,4\}$)。もし各点灯時間が時刻 t において最長点灯時間を超えたら、点灯変化は絶対許可である。以下の definite 規則をもつ。

$$MAX_{1,2}(g,t) \wedge T_{1,2}(g,t) \rightarrow \neg \neg T_{1,2}(a,t) \quad (19)$$

$$MAX_{1,2}(a,t) \wedge T_{1,2}(a,t) \rightarrow \neg \neg T_{1,2}(r,t) \quad (20)$$

$$MAX_{3,4}(g,t) \wedge T_{3,4}(g,t) \rightarrow \neg \neg T_{3,4}(a,t) \quad (21)$$

$$MAX_{3,4}(a,t) \wedge T_{3,4}(a,t) \rightarrow \neg \neg T_{3,4}(r,t) \quad (22)$$

上記の defeasible 規則と definite 規則の中で、それらの許可および禁止の間で競合がある。もし時刻 t において信号変化に対する許可が defeasible reasoning より導出されれば、点灯変化は時刻 $t+1$ において義務として実行される。また、もし時刻 t において信号変化に対する禁止が defeasible reasoning より導出されれば、点灯変化は時刻 $t+1$ において実行される必要がない。従って、definite 規則として以下の許可、禁止導出をもつ。

許可導出

$$T_{1,2}(g,t) \wedge \neg \neg T_{1,2}(a,t) \rightarrow \neg \neg T_{1,2}(a,t+1) \quad (23)$$

$$T_{1,2}(a,t) \wedge \neg \neg T_{1,2}(r,t) \rightarrow \neg \neg T_{1,2}(r,t+1) \quad (24)$$

$$T_{3,4}(g,t) \wedge \neg \neg T_{3,4}(a,t) \rightarrow \neg \neg T_{3,4}(a,t+1) \quad (25)$$

$$T_{3,4}(a,t) \wedge \neg \neg T_{3,4}(r,t) \rightarrow \neg \neg T_{3,4}(r,t+1) \quad (26)$$

禁止導出

$$T_{1,2}(g,t) \wedge \neg \neg T_{1,2}(a,t) \rightarrow \neg \neg T_{1,2}(g,t+1) \quad (27)$$

$$T_{1,2}(a,t) \wedge \neg \neg T_{1,2}(r,t) \rightarrow \neg \neg T_{1,2}(a,t+1) \quad (28)$$

$$T_{3,4}(g,t) \wedge \neg \neg T_{3,4}(a,t) \rightarrow \neg \neg T_{3,4}(g,t+1) \quad (29)$$

$$T_{3,4}(a,t) \wedge \neg \neg T_{3,4}(r,t) \rightarrow \neg \neg T_{3,4}(a,t+1) \quad (30)$$

また、隣接する交通信号機と直面する交通信号機およびそれと対面している交通信号機の間同期をとるためいくつかの definite 規則を必要とする。例えば、もし直面する交通信号機 T_1 が赤色を点灯しなくてはならないのならば、対面する交通信号機 T_2 もまた赤色を点

灯しなければいけない。そして、直面する交通信号機 T_1 が赤色を点灯しなくてはならないのならば、隣接する交通信号機 T_3 は緑色あるいは右折矢印信号を点灯しなくてはならない。

信号機の同期

$$\neg T_{1,2}(r,t) \rightarrow \neg T_{2,1}(r,t), \quad \neg T_{3,4}(r,t) \rightarrow \neg T_{4,3}(r,t) \quad (31)$$

$$\neg T_{1,2}(g,t) \rightarrow \neg T_{2,1}(g,t), \quad \neg T_{3,4}(g,t) \rightarrow \neg T_{4,3}(g,t) \quad (32)$$

$$\neg T_{1,2}(a,t) \rightarrow \neg T_{2,1}(a,t), \quad \neg T_{3,4}(a,t) \rightarrow \neg T_{4,3}(a,t) \quad (33)$$

$$\neg T_{1,2}(g,t) \rightarrow \neg T_{3,4}(r,t), \quad \neg T_{1,2}(g,t) \rightarrow \neg T_{3,4}(r,t) \quad (34)$$

$$\neg T_{3,4}(g,t) \rightarrow \neg T_{1,2}(r,t), \quad \neg T_{3,4}(a,t) \rightarrow \neg T_{1,2}(r,t) \quad (35)$$

3.4. EVALPSN による交通信号制御

defeasible 規則と definite 規則を EVALPSN に変換する。そのために defeasible 規則間の優先関係 (<) を仮定する。交差する 2 つの道路の間にはいかなる優先度がなければ、現在の点灯色を維持することが他の点灯色にすることよりも優先されると仮定する。それゆえに、場合 1 において信号機 $T_{3,4}$ の緑から右折矢印への点灯変化からの禁止は許可よりも優先であり、すなわち defeasible 規則(11)と(12)は defeasible 規則(7), (8), (9), (10), (13), (14)より優先である。従って以下の優先関係がある。

$$(7) < (11), (8) < (11), (9) < (11),$$

$$(10) < (11), (13) < (11), (14) < (11),$$

$$(7) < (12), (8) < (12), (9) < (12),$$

$$(10) < (12), (13) < (12), (14) < (12).$$

これに加えて、最小点灯時間と最長点灯時間の definite 規則が他のどのルールよりも優先されると仮定する。

ここで、例を用いて defeasible deontic 論理を EVALPSN に変換する方法を簡単に紹介する。2 つの競合している defeasible 規則, $R1: a \Rightarrow \neg \neg q$ と $R2: b \Rightarrow \neg \neg q$ 優先関係は $R1 < R2$ があると仮定する。もし a と b の両方を保持しているなら、優先関係より $\neg \neg q$ が導出される。もし a を保持して b を保持していないのなら、競合する規則 $R2$ が発火しないので $\neg \neg q$ が導出される。もし a を保持して b を保持しているのなら、競合する規則 $R1$ が発火しないので $\neg \neg q$ が導出される。defeasible 規則, definite 規則から EVALPSN への変換方法の詳細は[4]に記載されている。

defeasible 規則(7), (8), (9), (10), (13), (14), (11), (12), は次のように変換する。

$$\begin{aligned}
& S_1^{\alpha}(t) : [(2,0),\alpha] \wedge T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
& \sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \sim S_5^{\alpha}(t) : [(2,0),\alpha] \wedge \\
& \sim S_7^{\alpha}(t) : [(2,0),\alpha] \rightarrow T_{3,4}(a,t) : [(0,1),\gamma]
\end{aligned} \tag{36}$$

$$\begin{aligned}
& S_3^{\alpha}(t) : [(2,0),\alpha] \wedge T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
& \sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \sim S_5^{\alpha}(t) : [(2,0),\alpha] \wedge \\
& \sim S_7^{\alpha}(t) : [(2,0),\alpha] \rightarrow T_{3,4}(a,t) : [(0,1),\gamma]
\end{aligned} \tag{37}$$

$$\begin{aligned}
& S_2^{\alpha}(t) : [(2,0),\alpha] \wedge T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
& \sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \sim S_5^{\alpha}(t) : [(2,0),\alpha] \wedge \\
& \sim S_7^{\alpha}(t) : [(2,0),\alpha] \rightarrow T_{3,4}(a,t) : [(0,1),\gamma]
\end{aligned} \tag{38}$$

$$\begin{aligned}
& S_4^{\alpha}(t) : [(2,0),\alpha] \wedge T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
& \sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \sim S_5^{\alpha}(t) : [(2,0),\alpha] \wedge \\
& \sim S_7^{\alpha}(t) : [(2,0),\alpha] \rightarrow T_{3,4}(a,t) : [(0,1),\gamma]
\end{aligned} \tag{39}$$

$$\begin{aligned}
& S_6^{\alpha}(t) : [(2,0),\alpha] \wedge T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
& \sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \sim S_5^{\alpha}(t) : [(2,0),\alpha] \wedge \\
& \sim S_7^{\alpha}(t) : [(2,0),\alpha] \rightarrow T_{3,4}(a,t) : [(0,1),\gamma]
\end{aligned} \tag{40}$$

$$\begin{aligned}
& S_8^{\alpha}(t) : [(2,0),\alpha] \wedge T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
& \sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \sim S_5^{\alpha}(t) : [(2,0),\alpha] \wedge \\
& \sim S_7^{\alpha}(t) : [(2,0),\alpha] \rightarrow T_{3,4}(a,t) : [(0,1),\gamma]
\end{aligned} \tag{41}$$

$$\begin{aligned}
& S_5^{\alpha}(t) : [(2,0),\alpha] \wedge T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
& \sim MAX_{3,4}(g,t) : [(2,0),\alpha] \wedge \rightarrow T_{3,4}(a,t) : [(0,1),\beta]
\end{aligned} \tag{42}$$

$$\begin{aligned}
& S_7^{\alpha}(t) : [(2,0),\alpha] \wedge T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
& \sim MAX_{3,4}(g,t) : [(2,0),\alpha] \wedge \rightarrow T_{3,4}(a,t) : [(0,1),\beta]
\end{aligned} \tag{43}$$

definite規則 (17), (21), (25), (29)は次のように変換する。

$$\begin{aligned}
& MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \\
& \rightarrow T_{3,4}(a,t) : [(0,2),\beta]
\end{aligned} \tag{44}$$

$$\begin{aligned}
& MAX_{3,4}(g,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \\
& \rightarrow T_{3,4}(a,t) : [(0,2),\gamma]
\end{aligned} \tag{45}$$

$$\begin{aligned}
& T_{3,4}(g,t) : [(2,0),\alpha] \wedge T_{3,4}(a,t) : [(0,1),\gamma] \\
& \rightarrow T_{3,4}(a,t+1) : [(2,0),\beta]
\end{aligned} \tag{46}$$

$$\begin{aligned}
& T_{3,4}(g,t) : [(2,0),\alpha] \wedge T_{3,4}(a,t) : [(0,1),\beta] \\
& \rightarrow T_{3,4}(g,t+1) : [(2,0),\beta]
\end{aligned} \tag{47}$$

枚数制限のためdefinite規則 { (1 5), ..., (3 5) } の変換は省略する。

4. 課題とシミュレーション

例題 2

交通信号機 $T_{1,2}$ が赤信号, $T_{3,4}$ が緑信号であり, 緑色の最小点灯時間はすでに過ぎていと仮定する。

もし時刻 t においてセンサー $S_{1,3,5}$ が基準値以上の交通量を検知して, センサー $S_{2,4,6,7,8}$ は検知しないのならば, EVALPS 節(42)は発火され, 禁止 $T_{3,4}(g,t) : [(0,1),\beta]$ が導出される。さらに, EVALPSN 節(47)もまた発火され義務である結果 $T_{3,4}(g,t+1) : [(2,0),\beta]$ が得られる。

もし時刻 t においてセンサー $S_{1,3}$ が基準値以上の交通量を検知して, センサー $S_{2,4,5,6,7,8}$ は検知しないのならば, EVALPS 節(43)は発火され, 許可 $T_{3,4}(a,t) : [(0,1),\gamma]$ が導出される。さらに, EVALPSN 節(46)もまた発火され義務である結果 $T_{3,4}(a,t) : [(0,1),\beta]$ が得られる。

[シミュレーション]

我々は交通流モデルにセルラ・オートマトンモデルを使用し、以下の条件のもと停止した車と動いた車の数に関して EVALPSN の交通信号制御と固定時間で点灯変化する信号機の比較をした。

条件 1

十字路で 4 方向全てとも右折車 5%、左折車 5%、直進車 20% の確率で交差点へ流入すると仮定する。固定時間の信号機は緑色 30、黄色 3、右折矢印信号 4、赤色 40 (全赤の時間 3 も含む) 単位時間である。緑色の点灯時間は 3~14 単位時間の間の長さで、右折矢印信号は 1~4 単位時間の間の長さである。

条件 2

交差点へ流入する車の確率は以下のように仮定する。

南 : 右折車 5%、左折車 15%、直進車 10%

北 : 右折車 15%、左折車 5%、直進車 10%

西 : 右折車 5%、左折車 5%、直進車 5%

東 : 右折車 5%、左折車 5%、直進車 15%

他の条件は条件 1 と同様である。

我々は、1000 単位時間の間止まった車と動いた車の合計を測って、条件のもとで 10 回それを繰り返

した。止まった車と動いた車の平均の台数は表1で示されている。このシミュレーション結果から、動いた車の数は EVALPSN の方が固定時間のものよりも多く、止まった車の数は EVALPSN の方が固定時間のものよりも小さいことがわかる。シミュレーションを考慮すると EVALPSN で制御したものの方が固定時間のものよりも能率的であると帰結される。

表1：シミュレーション結果

		止まった車	動いた車
固定時間	条件 1	17690	19641
	条件 2	16764	18664
EVALPSN	条件 1	16285	23151
	条件 2	12738	20121

5. 結果と課題

我々は EVALPSN がセンサー値に依存している実時間の交通信号制御に基づいていることを提案した。我々が計画している実時間の交通信号制御の実用的な実装は、本論文ではそれを述べてないのだが、EVALPSN がマイクロチップ・ハードウェアで容易に実行することができる条件のもとである。

我々の将来の課題は、EVALPSN に基づくマルチ・エージェントの知的な交通信号制御であると考えている。基本的な考え方は、各交差点の信号機が知的エージェントのようにみなされ、各エージェントが近隣のエージェントへ信号制御の要求ができることである。

文 献

- [1] Gelder, A.V., Ross, K.A. and Schlipf, J.S. : The Well-Founded Semantics for General Logic Programs. *J. the Association for Computing Machinery*, **38** (1991) 620-650
- [2] Gelfond, M. and Lifschitz, V. : The Stable Model Semantics for Logic Programming. *Proc. 5th International Conference and Symposium on Logic Programming* (1989) 1070-1080
- [3] Nakamatsu, K., Abe, J.M. and Suzuki, A. : Defeasible Reasoning Between Conflicting Agents Based on VALPSN. *Proc. AAAI Workshop Agents' Conflicts*, AAAI Press (1999) 20-27
- [4] 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*, Springer-Verlag, (2001) 470-478
- [5] Nakamatsu, K. : On the Relation Between Vector Annotated Logic Programs and Defeasible Theories. *Logic and Logical Philosophy*, Vol.8 (2001) 181-205, UMK Press, Poland.
- [6] Nakamatsu, K., Abe, J.M. and Suzuki, A. : A Defeasible Deontic Reasoning System Based on

Annotated logic Programming. *Computing Anticipatory Systems, CASYS2000*, AIP Conference Proceedings Vol.573, (2001) 60-478 AIP Press.

- [7] Nakamatsu, K., Abe, J.M. and Suzuki, A. : Applications of EVALP Based Reasoning. *Logic, Artificial Intelligence Robotics, Frontiers in Artificial Intelligence and Applications Vol.71*, (2001) 174--185 IOS Press.
- [8] Nakamatsu, K., Suito, H., Abe, J.M. and Suzuki, A. : Paraconsistent Logic Program Based Safety Verification for Air Traffic Control. *Proc. 2002 IEEE Int'l Conf. Systems, Man and Cybernetics*. (2002) CD-ROM.
- [9] Nute, D. : *Basic Defeasible Logic, Intensional Logics for Programming*. Oxford Science Publications (1992) 126-154
- [10] Nute, D. : *Apparent Obligation, Defeasible Deontic Logic*. Kluwer Academic Publishers (1997) 287-316