

2022 年度卒業論文

時間経過による感情度合いの減衰を扱う論理体系に
基づいたエージェントの実装

奈良女子大学生生活環境学部情報衣環境学科生活情報通信科学コース 4 回生
新出研究室 19480307 佐藤はづ希

2023 年 2 月 15 日

概要

近年、ロボットに感情を付与する研究が進んでいる。感情を付与されることで、人間と意思疎通が行いやすくなることや、より人間らしい行動選択ができるようになるなどの効果が期待される。先行研究 [8] では、OCC theory と BDI モデルに基づいて定義された感情表現を Jason による BDI エージェントに実装していた。しかし、そこで用いていた論理体系は、感情が生起していることを表す論理式のみが定義されており、時間経過による感情度合いの減衰を扱えなかった。それにもかかわらず、同研究では実装の段階で強制的に減衰を付け加えており、論理体系と実装が整合しないという課題があった。それに対し、今井らによる論理体系 [9, 7] は、感情が生起されたことを表す論理式と、感情が時間経過により減衰していくことを表す論理式の 2 つを定義しており、時間経過による感情度合いの減衰を扱うことが可能である。そこで、本研究では、この論理体系に基づき、感情の生起と、時間経過による感情度合いの減衰の実装を行うことで、論理体系との不整合の問題を解消した。本研究での新たな実装は、従来の実装と同等の能力を持つ必要があるため、検証実験により確認を行った。また、未実装であった相反する感情の生起や、時間経過による推測感情の減衰の実装を新しく行うことで、より自然な感情生起ができるようになった。さらに、実装の正当性が不明瞭であったオペレータの実装や、OCC theory に照らして定義が不適切であった論理式の再定義を行い、より適切な実装となった。

1 はじめに

近年、感情を付加されたことで人間と意思疎通できるロボットの開発が進んでいる。このように、ロボットに感情を付加することで、感情に基づいた、より人間らしい行動選択ができるようになることが期待される。

周囲の環境が常に変化する実世界において、人間のように目標を達成する手段を選んで行動するロボットの実現には、BDI アーキテクチャが有効である。BDI アーキテクチャとは、BDI モデル [4] に基づいた行動決定方式を計算機上で実現したものである。BDI モデルとは、目的とそれを達成するための手段を信念・願望から選定しその手段を意図とする意図の理論に基づき人間の合理的行動をモデル化したものである。これにより、合理的思考に基づいた行動を選択するロボットを構築することが可能になる。

また、OCC theory [3] とは、心的状態を用いて人間の感情を 22 種類に分類し特徴付ける理論である。この理論は、論理モデルによって表現することが可能で、BDI logic を持つ BDI モデルとの親和性が高い。そのため、既存の BDI アーキテクチャの実装上の信念ベースで実現することが可能である。

Adam ら [1] の論文では、OCC theory で定義されている 22 種類の感情のうち、20 種類の感情を BDI モデルに組み込み論理式で表現していた。本研究の先行研究 [8] では、BDI エージェントの実装プラットフォームである Jason を用いて、Adam らの先行研究では実装されていなかった Love と Hate を含む 22 種類の感情全てが、度合いや時間経過に伴う度合いの減衰も含め実装されている。しかし、この実装で基にした論理体系は、時間経過に伴う感情度合い減衰を考慮しないものであった。この論理体系の論理式は、ある信念の存在を感情の生起条件とするため、信念が存続する限り同じように感情が生起されるはずである。信念はある程度の時間保持されるため、その間感情は生起されることになるにもかかわらず、実装では減衰関数を用いて強制的に感情度合い減衰を付け加えており、論理体系と整合しないという点に課題があった。

そこで本研究では、新しく提案された時間経過に伴う感情度合いの減衰を扱える論理体系 [9, 7] に基づき、この論理体系で導入された、信念及び度合い付きの信念の追加を表すオペレータ (Bel_{arise} ・ $Prob_{arise}$) を用いて、感情の生起条件を書き換え実装した。新しい論理体系では、これらのオペレータを用いて、ある感情が生起されたことを表す論理式を定義している。そして、それを用いて感情が時間経過に伴って減衰することを表す論理式を定義している。本研究では、まずこれらのオペレータを実装し、その後、感情が生起されたことを表す論理式を実装した。 Bel_{arise} や $Prob_{arise}$ 付きの論理式は信念の「追加」を表し、これは持続し

ないため、感情の生起の有無を検査し終わったらこれらのオペレータ付きの論理式はエージェントの信念ベースから削除した。これにより、感情が生起されたときに限り真になる論理式を実装できた。その後、時間経過に伴う感情の減衰は、論理式による表現をそのままルールとしては実装せず、減衰関数により等価な実装を行った。その際、本研究の実装が、従来の実装と比べ能力を失っていないこと、すなわち同等以上の能力を持つことが要求されるため、検証実験を行うことで確認した。加えて、従来の実装になかった推測感情の時間経過による度合いの減衰と、相反する感情の実装を新しく行うことで、より自然な感情生起が可能になった。また、新しい論理体系を提案した先行研究 [9, 7] の一部の論理式が OCC theory に即していなかったため、再定義を行った。さらに、先行研究 [8] では、一部に現れる「 P (現在を含む過去のある時点で)」というオペレータの実装の正当性が不明瞭であったため、その再実装を行った。新たな実装では、ある信念が追加されたら、追加された時刻や信念の内容を time 述語の引数に記憶し、記憶された時刻が現在時刻と同じかそれより古ければ、その信念は「現在を含む過去のある時点」のものであるとする方法で、「 P 」を実装した。

2 OCC theory

OCC theory[3] とは、心理学的見地により分類した 22 種類の感情タイプをモデル化した理論である。感情の特徴づけが明確であるため計算機科学分野においてよく用いられる。表 1 に 22 種類の感情の分類を示す。

3 先行研究

3.1 Adam らによる形式化

Adam ら [1] の形式化では、感情を BDI logic に基づく論理式で表現している。OCC theory[3] の 22 種類の分類を表 2 のように 6 グループに分類することで、グループ毎に同じ形式を持つ感情の生起条件が定義される。Adam らは、Love, Hate を除く感情のみ形式化を行ったが、本研究室の先行研究 [8] では 22 種類の感情すべてが形式化されている。

3.2 従来の実装における課題

本研究室の先行研究 [8] では 22 種類の感情すべてが形式化され、度合いや時間経過による度合いの減衰を実装していた。その実装で基にした論理体系 [1] は、時間経過に伴う感情度合いの減衰を考慮していないものである。その体系では、感情の生起条件はある信念が存在することであり、従って、信念が存続する限り感情が同じように生起するはずである。そして、信念はある程度の時間保持されるため、その間感情は生起されることになる。そうであるにもかかわらず、実装では減衰関数を用いて強制的に感情度合いの減衰を付け加えていた。つまり、論理体系にない減衰という概念を実装していたという課題があった。また、論理式の定義が一部 OCC theory の記述に即していないという課題や、生起条件の論理式に現れる「現在を含む過去のある時点で」を表すオペレータ P の実装の正当性が不明瞭という課題があった。さらに、より自然な感情生起の実現という観点からは、他者の感情を推測した場合の時間経過による度合い減衰の実現や、相反する感情の生起を可能にすることも課題であった。

表1 OCC theory による感情の分類

1. 事象の結果の望ましさに 関して	(a) 自分にとって の望ましさに 関して	i. イベントに関して		Joy(喜び) Distress(悲しみ)
		ii. 予想に関して	A. 単なる予想	Hope(望み) Fear(恐れ)
			B. 予想していた 事象が起こった	Satisfaction(満足) FearConfirmed(恐れていた ことが確定)
			C. 予想していた 事象が起こらな かった	Relief(安堵) Disappointment(落胆)
	(b) 他者にとって の望ましさに 関して	i. 他者が良い結果を得たことに関して		HappyFor(ともに喜ぶ) Resentment(憤る)
		ii. 他者が悪い結果を得たことに関して		SorryFor(ともに残念に思 う) Gloating(ほくそ笑む)
2. 行動に対する 賞賛度に関して	a. 自分の行動に関して			Pride(自尊心) Shame(羞恥心)
	b. 自分の行動とイベントに関して (混合型)			Gratification(満足) Remorse(後悔)
	c. 他者の行動に関して			Admiration(賞賛) Reproach(非難)
	d. 他者の行動とイベントに関して (混合型)			Gratitude(謝意) Anger(怒り)
3. 対象に対する好き嫌いに関して				Love(好き) Hate(嫌い)

表2 OCC theory のグループ分け

Well-being emotions	Joy, Distress
Prospect emotions	Hope, Fear
Confirmation emotions	Satisfaction, FearConfirmed, Relief, Disappointment
Fortunes-of-others	HappyFor, Resentment, SorryFor, Gloating
Attribution emotions	Pride, Shame, Admiration, Reproach, Gratification, Remorse, Gratitude, Anger
Attraction emotions	Love, Hate

4 新たな実装と検証

3.2 節で述べた問題を解決するため、時間経過による感情度合い減衰を扱える論理体系 [9, 7] に基づいた実装を新たに行った。また、新たな実装が以前の実装に比べ能力を失っていないので、以前の実装と同等以上の能力を持つことを確認するための検証実験も行った。それらについては次節以降で順次述べる。

5 生起条件の再定義

3.2 節で述べたように、先行研究 [9, 7] では、論理式の一部の定義において、OCC theory の記述に即していない部分があったため、再定義を行った。以後、本節では、「先行研究」とは [9, 7] を指す。

5.1 Fortune-others-emotions

SorryFor, Gloating, Guess_SorryFor および Gusee_Gloating

先行研究では、まず SorryFor_arise オペレータが以下のように定義されている。

$$\text{SorryFor_arise}_{f_{SF}(d_i, x, v, LD)}^{i, j} \varphi := \text{Guess_Distress_arise}_x^{i, j} \varphi \wedge \text{Des}_{d_i}^i \text{Bel}^j \varphi \wedge \text{Bel}^i \text{Deserve}_v^j \varphi \wedge \text{Love}_{LD}^i j$$

そして、これを用いて新たなオペレータ SorryFor を定義し、 $\text{SorryFor}_{f_{SF}(d, t)}^{i, j} \varphi$ の解釈を以下のように定義している。

$$\llbracket \text{SorryFor}_{f_{SF}(d, t)}^{i, j} \varphi \rrbracket_{(M, w)} = \top \text{ iff } \llbracket \text{SorryFor_arise}_d^{i, j} \varphi \rrbracket_{(M, x^{-t}(w))} = \top$$

ただし、 t は 0 以上の整数とし、上記の iff の右側を満たすような t が複数存在する場合、最小の t をとることにする。また、関数 f_{SF} は、 LD に関する増加関数、 d_i, x, v に関する減少関数、関数 f_{SF} は、 t に関する減少関数で、 $f_{SF}(d, 0) = d$ を満たすものとする。関数 f_{SF} 、関数 f_{SF} のどちらも、値域は $[0, 1]$ とする。

先行研究では、関数 f_{SF} は、 d_i, x, v に関する減少関数、 LD に関する増加関数と定義されていた。ここで、OCC theory では、「相手にとってどの程度望ましくない出来事であると推測されるのか」についての度合いが大きければ大きいほど SorryFor の度合いが大きくなると述べられている。これは、相手にとって望ましくないことが発生したと自身が信じたときに生起される相手の悲しみを推測する感情 (Guess_Distress) の度合い x が大きければ大きいほど SorryFor の度合いも大きくなることと解釈できる。従って、本研究では、関数 f_{SF} は、 LD, x に関する増加関数、 d_i, v に関する減少関数に変更した。

同様に、Gloating の定義は、度合いを決定する関数 f_{GV} が、 d_i, v, HTD に関する増加関数、相手の悲しみを推測する度合い x に関する減少関数となっているが、相手の悲しみを推測する度合いが大きければ大きいほど Gloating の度合いが大きくなるため、本研究では、関数 f_{GV} を、 d_i, v, HTD, x に関する増加関数に変更した。

また、Guess_SorryFor の度合い計算関数は、SorryFor の度合い計算関数と同じものを使っているため、SorryFor と同様に変更する。Guess_Gloating についても、Gloating の度合い計算関数と同じものを使っているため、Gloating と同様に変更する。

5.2 Attribution emotions

Pride および Shame

先行研究では、まず Pride_arise の生起条件は以下のように定義されている。

$$\text{Pride_arise}_{f_{Pr}(p,l)}^i \varphi := \text{Praise}_p^j \varphi \wedge \text{Bel}^i \text{PProb}_l^i \varphi \wedge \text{Bel_arise}^i \varphi \wedge l \neq 1$$

この定義では、 i が意図的に φ を行ったことが論理式に表現されていない。しかし、OCC theory による定義を見ると、 i は、 i が意図的に φ をしたことに對して賞賛しているため、この部分を論理式に入れる必要がある。そのため、本研究では、 $\text{Bel}^i \text{PEffort}_a^i \varphi$ によって、 i が φ を真にするために a 程度努力したという意味を加える。 i が φ を真にするために努力したということは、 i が φ をしたことと解釈する。

そこで、本研究では Pride_arise の生起条件を以下のように変更する。

$$\text{Pride_arise}_{f_{Pr}(p,l,a)}^i \varphi := \text{Praise}_p^j \varphi \wedge \text{Bel}^i \text{PProb}_l^i \varphi \wedge \text{Bel_arise}^i \varphi \wedge \text{Bel}^i \text{PEffort}_a^i \varphi \wedge l \neq 1$$

そして、これを用いて新たなオペレータ Pride を定義し、 $\text{Pride}_{f_{Pr}(d,t)}^i \varphi$ の解釈を以下のように定義する。

$$\llbracket \text{Pride}_{f_{Pr}(d,t)}^i \varphi \rrbracket_{(M,w)} = \top \text{ iff } \llbracket \text{Pride_arise}_{f_{Pr}(d,t)}^i \varphi \rrbracket_{(M,x^{-t}(w))} = \top$$

ただし、 t は 0 以上の整数とし、上記の iff の右側を満たすような t が複数存在する場合、最小の t をとることにする。また、関数 f_{Pr} は、 p, a に関する増加関数、 l に関する減少関数であり、関数 f_{Pr} は、 t に関する減少関数で、 $f_{Pr}(d,0) = d$ を満たすものとする。関数 f_{Pr} 、関数 f_{Pr} のどちらも、値域は $[0,1]$ とする。

同様に、 Shame_arise の生起条件にも、 $\text{Bel}^i \text{PEffort}_a^i \varphi$ を加えた。

Admiration および Reproach

先行研究では、まず Admiration_arise の生起条件は以下のように定義されている。

$$\text{Admiration_arise}_{f_{Ad}(p,l)}^i \varphi := \text{Bel}^i \text{Praise}_p^j \varphi \wedge \text{Bel}^i \text{PProb}_l^i \varphi \wedge \text{Bel_arise}^i \varphi \wedge l \neq 1$$

この定義では、 j が意図的に φ を行ったことが論理式に表現されていない。しかし、OCC theory による定義を見ると、 i は、 j が意図的に φ をしたことに對して賞賛しているため、この部分を論理式に入れる必要がある。そのため、本研究では、 $\text{Bel}^i \text{PEffort}_a^i \varphi$ によって、 j が φ を真にするために a 程度努力したという意味を加える。 j が φ を真にするために努力したということは、 j が φ をしたことと解釈する。

そこで、本研究では Admiration_arise の生起条件を以下のように変更する。

$$\text{Admiration_arise}_{f_{Ad}(p,l,a)}^{i,j} \varphi := \text{Bel}^i \text{Praise}_p^j \varphi \wedge \text{Bel}^i \text{PProb}_l^i \varphi \wedge \text{Bel_arise}^i \varphi \wedge \text{Bel}^i \text{PEffort}_a^i \varphi \wedge l \neq 1$$

そして、これを用いて新たなオペレータ Admiration を定義し、 $\text{Admiration}_{f_{Ad}(d,t)}^{i,j} \varphi$ の解釈を以下のように定義する。

$$\llbracket \text{Admiration}_{f_{Ad}(d,t)}^{i,j} \varphi \rrbracket_{(M,w)} = \top \text{ iff } \llbracket \text{Admiration_arise}_{f_{Ad}(d,t)}^{i,j} \varphi \rrbracket_{(M,x^{-t}(w))} = \top$$

ただし、 t は 0 以上の整数とし、上記の iff の右側を満たすような t が複数存在する場合、最小の t をとることにする。また、関数 f_{Ad} は、 p, a に関する増加関数、 l に関する減少関数であり、関数 f_{Ad} は、 t に関する減少関数で、 $f_{Ad}(d,0) = d$ を満たすものとする。関数 f_{Ad} 、関数 f_{Ad} のどちらも、値域は $[0,1]$ とする。

同様に、 Reproach_arise の生起条件にも、 $\text{Bel}^i \text{PEffort}_a^i \varphi$ を加えた。

Guess_Pride および Guess_Shame

先行研究では、まず Guess_Pride_arise の生起条件は以下のように定義されている。

$$\text{Guess_Pride_arise}_{f_{Pr}(p,l)}^{i,j} \varphi := \text{Bel}^i \text{Praise}_p^j \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PProb}_l^j \varphi \wedge \text{Bel}^i \text{Bel_arise}^j \varphi \wedge l \neq 1$$

この定義では、 j が意図的に φ を行ったことが論理式に表現されていない。Pride や Admiration と同様に、本研究では、 $\text{Bel}^i \text{Bel}^j \text{PEffort}_a^j \varphi$ によって、 j が φ を真にするために a 程度努力したという意味を加える。これにより、 j が φ をしたことと解釈する。

そこで、本研究では Guess_Pride_arise の生起条件を以下のように定義する。

$$\begin{aligned} \text{Guess_Pride_arise}_{f_{Pr}(p,l,a)}^{i,j} \varphi &:= \text{Bel}^i \text{Praise}_p^j \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PProb}_l^j \varphi \\ &\wedge \text{Bel}^i \text{Bel_arise}^j \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PEffort}_a^j \varphi \wedge l \neq 1 \end{aligned}$$

そして、これを用いて新たなオペレータ Guess_Pride を定義し、 $\text{Guess_Pride}_{f_{Pr}(d,t)}^{i,j} \varphi$ の解釈を以下のように定義する。

$$\llbracket \text{Guess_Pride}_{f_{Pr}(d,t)}^{i,j} \varphi \rrbracket_{(M,w)} = \top \text{ iff } \llbracket \text{Guess_Pride_arise}_d^{i,j} \varphi \rrbracket_{(M,x^{-t}(w))} = \top$$

ただし、 t は 0 以上の整数とし、上記の iff の右側を満たすような t が複数存在する場合、最小の t をとることにする。関数 f_{Pr} と f_{Pr} は Pride_arise の定義に用いた関数と同様のものである。

同様に、 Guess_Shame_arise の生起条件にも、 $\text{Bel}^i \text{Bel}^j \text{PEffort}_a^j \varphi$ を加えた。

Guess_Admiration および Guess_Reproach

先行研究では、まず $\text{Guess_Admiration_arise}$ の生起条件は以下のように定義されている。

$$\begin{aligned} \text{Guess_Admiration_arise}_{f_{Ad}(p,l)}^{i,j,k} \varphi &:= \text{Bel}^i \text{Bek}^j \text{Praise}_p^k \varphi \wedge \\ &\text{Bel}^i \text{Bel}^j \text{PProb}_l^j \varphi \wedge \text{Bel}^i \text{Bel_arise}^j \varphi \wedge l \neq 1 \end{aligned}$$

この定義では、 k が意図的に φ を行ったことが論理式に表現されていない。Pride や Admiration と同様に、本研究では、 $\text{Bel}^i \text{Bel}^j \text{PEffort}_a^k \varphi$ によって、 k が φ を真にするために a 程度努力したという意味を加える。これにより、 k が φ をしたことと解釈する。

そこで、本研究では $\text{Guess_Admiration_arise}$ の生起条件を以下のように定義する。

$$\begin{aligned} \text{Guess_Admiration_arise}_{f_{Ad}(p,l,a)}^{i,j,k} \varphi &:= \text{Bel}^i \text{Bel}^j \text{Praise}_p^k \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PProb}_l^j \varphi \\ &\wedge \text{Bel}^i \text{Bel_arise}^j \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PEffort}_a^k \varphi \wedge l \neq 1 \end{aligned}$$

そして、これを用いて新たなオペレータ Guess_Admiration を定義し、 $\text{Guess_Admiration}_{f_{Ad}(d,t)}^{i,j,k} \varphi$ の解釈を以下のように定義する。

$$\llbracket \text{Guess_Admiration}_{f_{Ad}(d,t)}^{i,j,k} \varphi \rrbracket_{(M,w)} = \top \text{ iff } \llbracket \text{Guess_Admiration_arise}_d^{i,j,k} \varphi \rrbracket_{(M,x^{-t}(w))} = \top$$

ただし、 t は 0 以上の整数とし、上記の iff の右側を満たすような t が複数存在する場合、最小の t をとることにする。関数 f_{Ad} と f_{Ad} は Admiration_arise の定義に用いた関数と同様のものである。

同様に、 $\text{Guess_Reproach_arise}$ の生起条件にも、 $\text{Bel}^i \text{Bel}^j \text{PEffort}_a^k \varphi$ を加えた。

6 実装の不具合の解消

これまでの実装 [8] では、実装の正当性が不明瞭であったり、実装の際に不具合が生じたりすることがあったため、その解消を行った。

6.1 「現在を含む過去のある時点で」を表すオペレータの実装

例えば、「現在を含む過去のある時点で φ に対して hope を生起した」ことが、 φ に対する satisfaction の生起条件の一部になるように、現在を含む過去のある時点で感情が生起されたことが他の感情の生起条件の一部になる場合がある。しかし、先行研究 [8] では、「現在を含む過去のある時点で」を表すオペレータ P の実装について、正当性が不明瞭だった。そこで、本研究では、 P オペレータの実装を、感情の生起時間を記録するための time 述語を用いたものに変更した。

感情が生起されたら、感情の種類 (joy, distress など)、生起主、感情の対象、生起された時の度合い、生起された時刻を time 述語の引数に記憶する。happyfor や sorryfor など、誰に対する感情であるかが必要な場合や、誰の感情を推測しているのかが必要な推測感情の場合はそれらも記憶するようにしている。感情の生起を検査するプログラム allem_check.asl の中で、感情の生起を確認した後、その感情の time 述語を追加する。また、感情以外の Bel や Prob 付きの信念が追加されたときにも、time 述語を導入して、追加された時刻を記憶するようにした。「現在を含む過去のある時点で φ を l 程度信じている」ことが φ に対する pride の生起条件の一部になるように、現在を含む過去のある時点で追加された信念が他の感情の生起条件の一部になる場合があるためである。Bel 付きの信念が追加された場合は、信念の内容と追加された時間を time 述語の引数に記憶するようにした。例えば、bel(X,Z) が追加されたときには、time(Z, Bel オペレータ付きの信念が追加された時間) を追加するようにした。Prob 付きの信念が追加された場合には、prob、生起主、度合い付き信念の内容、度合い、信念が追加された時間を time 述語の引数に記憶するようにした。例えば、prob(X,Z)[degOfCert(D)] が追加されたときには、time(prob, X, Z, Prob オペレータ付きの信念が追加されたときの時間, D) を追加するようにした。

以上から、具体的に P を実装する方法を説明する。Jason に標準に用意されている関数 system.time を用いて現在時刻を取得し、time 述語に記憶された時刻が現在時刻と同じかそれより古ければ、time 述語に記憶された信念は現在を含む過去のものであるとした。

6.2 感情の生起を表す論理式の実現

Bel や Prob 付きの信念が追加されたとき、7.2 節で述べるように、*Prob_arise* と *Bel_arise* 付き信念が真となる条件が満たされていたらそれらを追加する。また、Bel や Prob 付きの信念はエージェントの信念ベースから削除し、6.1 節で記述したように time 述語にその内容を記憶する。

従来の実装を踏襲して、信念が追加されたら感情の生起の有無を検査するようにしているが、その後、*Prob_arise* と *Bel_arise* 付きの信念をエージェントの信念ベースから削除するようにした。そうすることで、*Prob_arise* と *Bel_arise* 付きの信念が真になったときに限り、感情が生起されたことを表す論理式を真にすることができた。Bel や Prob 付きの信念が追加された時の様子を図 1 に示す。

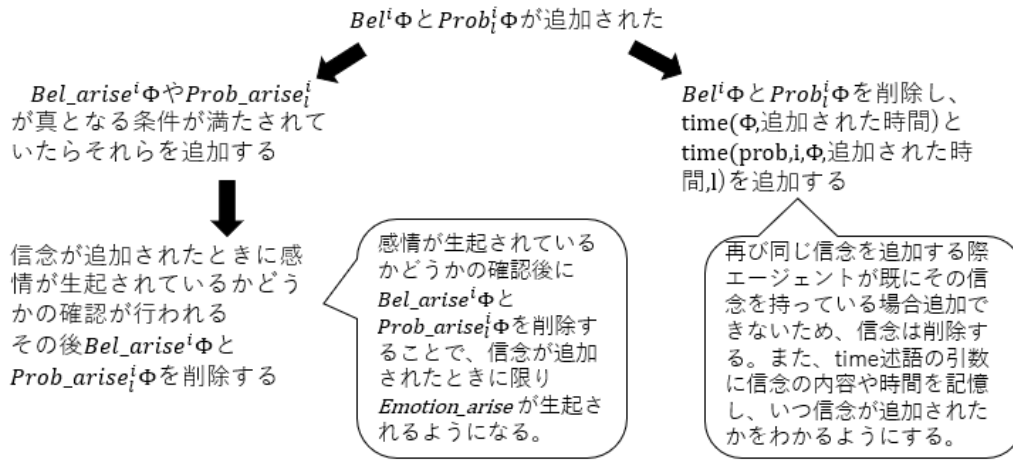


図1 Bel や Prob 付きの論理式が追加された時

6.3 否定の実装

これまでの実装では、論理式中の \neg はすべて *not* で表現されていた。Jason[2] では、ある信念 l が信念ベースにないことを表すためには *not l* を用いるが、強い否定 (l が偽であると信じる場合) は $\sim l$ を用いる。感情の生起条件には、「あるイベントが真にならないように努力する」、「あるイベントが真にならないことを hope する」などがある。「あるイベントが真にならないように努力する」は、「あるイベントが偽になるように努力する」と解釈でき、「あるイベントが真にならないことを hope する」は、「あるイベントが偽になることを hope する」と解釈できる。これらは、ある信念が偽であることに対して hope や努力をしているため、強い否定で表すのが妥当だと考えられる。

そこで本研究では、これらの表現には \sim を用いるように変更した。しかし、すでに \sim がついているリテラルの場合、もう1つ \sim は付けられないので、否定するには \sim を外す必要がある。そこでリテラルの否定を求めるための述語 *deny* を以下のように定義した。

```
deny(X, Y):- not .ground(Y)&.atom(X)&Y= $\sim$ X.
deny( $\sim$ X, Y):- not .ground(Y)& Y=X.
deny(X, Y):- not .ground(X)&.atom(Y)&X= $\sim$ Y.
deny(X,  $\sim$ Y):- not .ground(X)& X=Y.
deny(X,  $\sim$ Y):- X= $\sim$ Y.
deny( $\sim$ X, Y):- X=Y.
```

1・2番目の節は、第2引数が未代入の場合、3・4番目の節は第1引数が未代入の場合に使われ、既に代入済みの引数同士で一方が他方の否定になっていることを判定する場合は5・6番目の節が使われる。この述語を導入することで、イベントに \sim がついている場合は外すことができ、 \sim がついていない場合は \sim をつけることができるようになる。

6.4 信念を他エージェントに伝える機能の実装

信念を他のエージェントに伝えるときには Jason の chat 機能を用いる。先行研究 [8] の実装では、伝える側のエージェントを X 、伝えられる側のエージェントを Y 、伝えたい内容を Z とすると、 $\text{bel}(Y,Z)$ を内部アクション `.send` を用いて Y に送っていた。`.send` の第二引数には、話者が発話を通して遂行しようとする意思のタイプを指定するが、ここでは `tell` を指定していた。これは単に伝えるだけである。そうすると、 Y の信念には、 $[\text{source}(X)]$ という annotation 付きの $\text{bel}(Y,Z)$ が追加される。実装は以下の通りであった。

```
+!chat(Y,Z)
:friend(Y)
<- .send(Y,tell,bel(Y,Z));
.print(send).
```

本研究では、`Prob` や `Bel` 付きの信念が追加されるたびに、その信念をエージェントの信念ベースから削除し、`time` 述語に内容を記憶させるようにした。しかし、これまでの実装のままでは、信念の削除ができないという問題に直面した。追加された信念の annotation には、伝えてきた相手が `source` であると記述されており、削除したい信念にはそのような annotation はつけられない。そのため、削除したい信念と追加された信念が一致せず、追加された信念を削除できないからである。この問題を解決するため、chat 機能の実装の修正と新しいプランの作成を行った。伝えられる側のエージェントを Y 、伝えたい内容を Z とすると、引数に $\text{bel}(Y,Z)$ を入れた達成目標 `add_bel` を内部アクション `.send` で送る。ここで `.send` の第二引数に指定したのは `achieve` である。`achieve` は、メッセージの内容にあるリテラルが真となる状態を実現するように依頼する(相手に目標を委任する)という意味を持つ。また、`.send` で伝えられたときに追加される達成目標に対応する `+!add_bel(X)` をトリガーイベントとするプランも作成した。実装は以下の通りである。

```
+!chat(Y,Z)
:friend(Y)
<- .send(Y,achieve,add_bel(bel(Y,Z)));
.print(send).

+!add_bel(X)
<- +X.
```

こうすることで、`!add_bel()` が追加されたことをトリガーイベントとするプランが選ばれ、そのプランのボディで、引数に入っている信念を追加するイベントを、伝えられる相手に発生させる。これにより、追加される信念の `source` は、伝えてきた相手でなく自分自身になった。このように修正することで、削除したい信念と追加された信念が一致し、信念を削除することが可能になった。例えば、エージェント X がエージェント Y に $\text{des}(X,W)$ という信念を送るとすると、`+!add_bel(bel(Y,des(X,W)))` という信念追加イベントが発生する。そして、これがトリガーイベントとなり、プランが実行され、新しいイベント `+bel(Y,des(X,W))` が発生する。また、chat 機能で信念を伝えるということは、`Bel` 付きの信念が追加されることと同じである。

7 時間経過による感情度合い減衰を扱える論理体系に基づいた実装

本研究では、信念及び度合い付きの信念が追加されたことを表すオペレータ Bel_{arise} , Prob_{arise} を新たに導入し、感情が生起されたことを表す論理式の実装を行った。

7.1 基にした論理体系

基にした論理体系の意味論と構文は、先行研究 [9, 7] で提案されていたものである。

7.2 $Prob_arise$ と Bel_arise の定義と実装

先行研究 [9, 7] で提案された、時間経過による感情度合い減衰を扱える論理体系にて導入されたオペレータである $Prob_arise$ と Bel_arise を実装する。 $Prob_arise$ と Bel_arise はこのように定義されている。

$$Prob_arise^i \varphi := Prob^i \varphi \wedge YProb^i_0 \varphi$$

$$Bel_arise^i \varphi := Bel^i \varphi \wedge \neg YBel^i \varphi$$

「 $Y\varphi$ 」は 1 時刻前で φ が真だと信じていることを表す。直感的には、「 φ が現在時刻より 1 時刻前で 0 程度成り立つ (全く成り立たない)」かつ「 φ が現在時刻で l 程度成り立つ」の略記として $Prob_arise^i \varphi$ を定義している。また、「 φ が現在時刻より 1 時刻前で成り立たない」かつ「 φ が現在時刻で成り立つ」の略記として $Bel_arise^i \varphi$ を定義している。

本研究では $Prob_arise$ を新たに以下のように定義する。

$$Prob_arise^i \varphi := Prob^i_m \varphi \wedge YProb^i_n \varphi \wedge m - n \geq \text{基準値}$$

「 φ が現在時刻より 1 時刻前で n 程度成り立つ」かつ「 φ が現在時刻で m 程度成り立つ」かつ「 $m - n \geq$ 基準値」であることを $Prob_arise^i \varphi$ の定義とした。旧定義で $Prob_arise^i \varphi$ が生じるためには、1 時刻前に φ を信じている度合いが 0 である必要があった。現実世界では、全く信じていないわけではないがある程度信じている状態から、現在にかけて信じる度合いが増える場合もあるため、基準値以上 (実装の際は 0.3 とした) 信じる度合いが増えたら、 $Prob_arise^i \varphi$ が生じると定義した。

$Prob_arise$ と Bel_arise の実装について、図 2、図 3 に示す。Prob・Bel 付きの信念が追加されたことをトリガーイベントとするプランを作成した。 $Prob_arise \cdot Bel_arise$ の定義の右辺の論理式が真であれば、 $Prob_arise \cdot Bel_arise$ 付きの信念を追加することをプランのボディに記述する。Jason では「1 時刻前にその信念があること」または「1 時刻前にその信念がないこと」は実装できないため、時刻に関係なく、信念ベースに存在するかどうかでこれらを表現した。 Bel_arise の実装は、Prob 付きの信念がすでに信念ベースにある場合は、その度合いが 1 でない場合に Bel_arise 付きの信念と、Bel 付きの信念の内容を記憶した time 述語を追加した。Prob 付きの信念がすでに信念ベースにない場合は、 Bel_arise 付きの信念と、Bel 付きの信念の内容を記憶した time 述語を追加した。 $Prob_arise$ の実装は、Prob 付きの信念がすでに信念ベースにある場合は、追加された度合いがその度合いより 0.3 以上増えていたら、 $Prob_arise$ 付きの信念と Prob 付きの信念の内容を記憶した time 述語を追加した。 $Prob_arise$ 付きの信念がすでに信念ベースにない場合は、 $Prob_arise$ 付きの信念と Prob 付きの信念の内容を記憶した time 述語を追加した。time 述語については、6.1 節で記述した。

7.3 感情が生起されたことを表す実装

感情が生起されたことを表す論理式の実装方法を示す。従来の実装に対して改めた部分が多いため、以下には一部の感情について略すことなく、全ての感情に関する新たな実装を記す。アトムsの第一引数を主体、第二

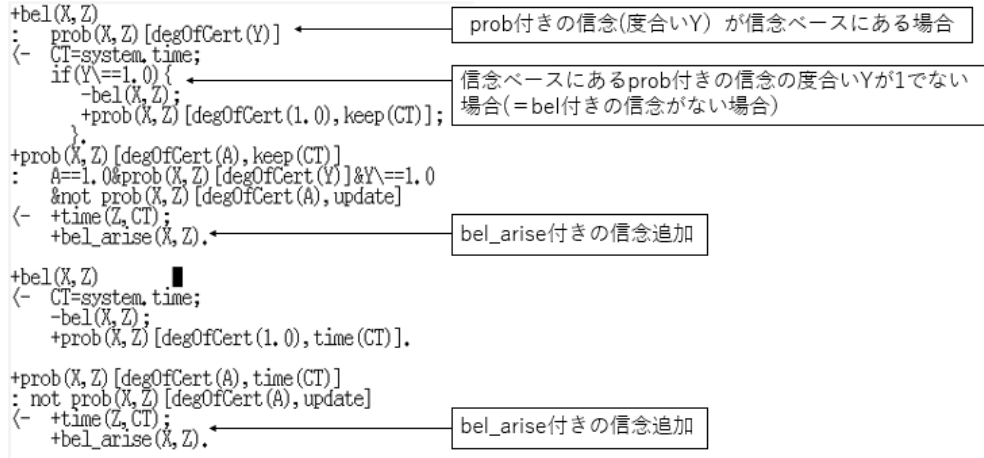


図2 Bel_ariseの実装



図3 Prob_ariseの実装

引数を対象とし、annotationに度合いを表した。Jasonで実装する際、形式化に現れる論理式の各項を信念で表現するが、以下では、≡の左側に項、右側にJasonでの実装方法を記述した。各感情は、ルールとして生起条件を記述した。以後、本節で「先行研究」とは[9, 7]を指す。

7.3.1 Well-being emotions

Joy

先行研究では、Joy_ariseの生起条件は以下のように定義されている。

$$Joy_arise_{f,r(d)}^i \varphi := Des_d^i \varphi \wedge Bel_arise^i \varphi$$

Jason で実装する際、変数 JD, D を用いて論理式の各項を以下のように表現した。また、 $f_{JD}(d)$ は、D を受けとって JD に代入する述語 joydeg(JD,D) で表現している。

$$\begin{aligned} Joy_arise_{f_{JD}(d)}^i \varphi &\equiv \text{joy}(X,Z)[\text{degOfCert}(JD)] \\ Des_d^i \varphi &\equiv \text{des}(X,Z)[\text{degOfCert}(D)] \\ Bel_arise^i \varphi &\equiv \text{bel_arise}(X,Z) \\ JD: Joy \text{ の度合い} \quad D: X \text{ が } Z \text{ を望む度合い} \end{aligned}$$

その上で、上述した joy_arise の生起条件の右辺に相当する信念が得られた場合、左辺に相当する信念を生起している。なお、この点については、これ以後に述べる感情に関しても同様である。

Distress

先行研究では、Distress_arise の生起条件は以下のように定義されている。

$$Distress_arise_{f_{D'}(d)}^i \varphi := Des_d^i \varphi \wedge Bel_arise^i \varphi$$

Jason で実装する際、変数 DD, D を用いて論理式の各項を以下のように表現した。1 から、X にとっての Z の望ましさを度合い D を引いた値をイベントの望ましくなさの度合いとする。 $f_{D'}(d)$ は、D を受けとって $(1 - D)$ を DD に代入する述語 disdeg(DD,D) で表現している。

$$\begin{aligned} Distress_arise_{f_{D'}(d)}^i \varphi &\equiv \text{distress}(X,Z)[\text{degOfCert}(DD)] \\ Des_d^i \varphi &\equiv \text{des}(X,Z)[\text{degOfCert}(D)] \\ Bel_arise^i \varphi &\equiv \text{bel_arise}(X,Z) \\ DD: Distress \text{ の度合い} \quad D: X \text{ が } Z \text{ を望む度合い} \end{aligned}$$

7.3.2 Fortune-others-emotions

HappyFor

先行研究では、HappyFor_arise の生起条件は以下のように定義されている。

$$HappyFor_arise_{f_{HFD}(d_i, x, v, LD)}^{i,j} \varphi := Guess_Joy_arise_x^{i,j} \varphi \wedge Des_{d_i}^i Bel^j \varphi \wedge Bel^i Deserve_v^j \varphi \wedge Love_{LD}^i j$$

Jason で実装する際、変数 HFD, D1, D2, D3, D4 を用いて論理式の各項を以下のように表現した。また、 $f_{HFD}(d_i, x, v, LD)$ は、変数を受けとりその総和を 4 で割る述語 happyfordeg(HFD,D1,D2,D3,D4) で表現している。

$Bel^i Deserve_v^j \varphi$ は、「j にとって φ が v 程度ふさわしい」ことを i が信じているという論理式である。この信念は、 $\text{bel}(X, \text{deserve}(Y,Z)[\text{degOfCert}(D3)])$ を外部入力から追加することによって表現する。Bel 付きの信念が追加されたときには、信念の内容と時間を記憶する time 述語が追加される。そのため、 $\text{time}(\text{deserve}(Y,Z)[\text{degofCert}(D3)], \text{Bel 付きの信念が追加された時間})$ という信念が追加されることになる。Bel 付きの信念が追加された時間は任意であるため、無名変数「_」で表している。

$$\begin{aligned} HappyFor_arise_{f_{HFD}(d_i, x, v, LD)}^{i,j} \varphi &\equiv \text{happyfor}(X,Y,Z)[\text{degOfCert}(HFD)] \\ Guess_Joy_arise_x^{i,j} \varphi &\equiv \text{guess_joy}(X,Y,Z)[\text{degOfCert}(D1)] \\ Des_{d_i}^i Bel^j \varphi &\equiv \text{des}(X, \text{bel}(Y,Z))[\text{degOfCert}(D2)] \\ Bel^i Deserve_v^j \varphi &\equiv \text{time}(\text{deserve}(Y,Z)[\text{degOfCert}(D3)], -) \end{aligned}$$

$Love_{LD}^i j \equiv \text{love}(X, Y)[\text{degOfCert}(D4)]$

HFD: *HappyFor* の度合い D1: *Guess-Joy* の度合い D2: Y が Z を真だと信じることを X が望む度合い D3: Y にとっての Z のふさわしさ D4: X が Y を好む度合い

SorryFor

SorryFor_arise の生起条件は以下のように定義した。

$SorryFor_arise_{f_{SF}(d_i, x, v, LD)}^{i, j} \varphi := \text{Guess_Distress_arise}_x^{i, j} \varphi \wedge \text{Des}_{d_i}^i \text{Bel}^j \varphi \wedge \text{Bel}^i \text{Deserve}_v^j \varphi \wedge \text{Love}_{LD}^i j$

Jason で実装する際、変数 SFD, D1, D2, D3, D4 を用いて論理式の各項を以下のように表現した。「 j が φ を真だと信じる」ことを i が望まない度合いを $(1 - D2)$ 、 j にとっての φ のふさわしさを $(1 - D3)$ で表した。 $f_{SF}(d_i, x, v, LD)$ は変数を受けとり、D1、 $(1 - D2)$ 、 $(1 - D3)$ 、D4 の和を 4 で割る述語 $\text{sorryfordeg}(SFD, D1, D2, D3, D4)$ で表現している。 $\text{Bel}^i \text{Deserve}_v^j \varphi$ は、*HappyFor* と同じように実装した。

$SorryFor_arise_{f_{SF}(d_i, x, v, LD)}^{i, j} \varphi \equiv \text{sorryfor}(X, Y, Z)[\text{degOfCert}(SFD)]$

$\text{Guess_Distress_arise}_x^{i, j} \varphi \equiv \text{guess_distress}(X, Y, Z)[\text{degOfCert}(D1)]$

$\text{Des}_{d_i}^i \text{Bel}^j \varphi \equiv \text{des}(X, \text{bel}(Y, Z))[\text{degOfCert}(D2)]$

$\text{Bel}^i \text{Deserve}_v^j \varphi \equiv \text{time}(\text{deserve}(Y, Z)[\text{degOfCert}(D3)], -)$

$\text{Love}_{LD}^i j \equiv \text{love}(X, Y)[\text{degOfCert}(D4)]$

SFD: *SorryFor* の度合い D1: *Guess-Distress* の度合い D2: Y が Z を真だと信じることを X が望む度合い D3: Y にとっての Z のふさわしさ D4: X が Y を好む度合い

Resentment

先行研究では、*Resentment_arise* の生起条件は以下のように定義されている。

$Resentment_arise_{f_{RS}(d_i, x, v, HD)}^{i, j} \varphi := \text{Guess_Joy_arise}_x^{i, j} \varphi \wedge \text{Des}_{d_i}^i \text{Bel}^j \varphi \wedge \text{Bel}^i \text{Deserve}_v^j \varphi \wedge \text{Hate}_{HD}^i j$

Jason で実装する際、変数 RSD, D1, D2, D3, D4 を用いて論理式の各項を以下のように表現した。「 j が φ を真だと信じる」ことを i が望まない度合いを $(1 - D2)$ 、 j にとっての φ のふさわしさを $(1 - D3)$ で表した。 $f_{RS}(d_i, x, v, HD)$ は変数を受けとり、D1、 $(1 - D2)$ 、 $(1 - D3)$ 、D4 の和を 4 で割る述語 $\text{resentmentdeg}(RSD, D1, D2, D3, D4)$ で表現している。 $\text{Bel}^i \text{Deserve}_v^j \varphi$ は、*HappyFor* と同じように実装した。

$Resentment_arise_{f_{RS}(d_i, x, v, HD)}^{i, j} \varphi \equiv \text{resentment}(X, Y, Z)[\text{degOfCert}(RSD)]$

$\text{Guess_Joy_arise}_x^{i, j} \varphi \equiv \text{guess_joy}(X, Y, Z)[\text{degOfCert}(D1)]$

$\text{Des}_{d_i}^i \text{Bel}^j \varphi \equiv \text{des}(X, \text{bel}(Y, Z))[\text{degOfCert}(D2)]$

$\text{Bel}^i \text{Deserve}_v^j \varphi \equiv \text{time}(\text{deserve}(Y, Z)[\text{degOfCert}(D3)], -)$

$\text{Hate}_{HD}^i j \equiv \text{hate}(X, Y)[\text{degOfCert}(D4)]$

RSD: *Resentment* の度合い D1: *Guess-Joy* の度合い D2: Y が Z を真だと信じることを X が望む度合い D3: Y にとっての Z のふさわしさ D4: X が Y を嫌う度合い

Gloating

Gloating_arise の生起条件は以下のように定義した。

$$Gloating_arise_{f_{GI}(d_i, x, v, HD)}^{i, j} \varphi := Guess_Distress_arise_x^{i, j} \varphi \wedge Des_{d_i}^i Bel^j \varphi \wedge Bel^i Deserve_v^j \varphi \wedge Hate_{HD}^i j$$

Jason で実装する際、変数 GD, D1, D2, D3, D4 を用いて論理式の各項を以下のように表現した。また、 $f_{GI}(d_i, x, v, HD)$ は変数を受けとり、その総和を 4 で割る述語 gloatingdeg(GD, D1, D2, D3, D4) で表現している。 $Bel^i Deserve_v^j \varphi$ は、HappyFor と同じように実装した。

$$Gloating_arise_{f_{GI}(d_i, x, v, HD)}^{i, j} \varphi \equiv gloating(X, Y, Z)[degOfCert(GD)]$$

$$Guess_Distress_arise_x^{i, j} \varphi \equiv guess_distress(X, Y, Z)[degOfCert(D1)]$$

$$Des_{d_i}^i Bel^j \varphi \equiv des(X, bel(Y, Z))[degOfCert(D2)]$$

$$Bel^i Deserve_v^j \varphi \equiv time(deserve(Y, Z)[degOfCert(D3)], -)$$

$$Hate_{HD}^i j \equiv hate(X, Y)[degOfCert(D4)]$$

GD: Gloating の度合い D1: Guess_Joy の度合い D2: Y が Z を真だと信じることを X が望む度合い D3: Y にとっての Z のふさわしさ D4: X が Y を嫌う度合い

7.3.3 Prospect emotions

Hope

先行研究では、Hope_arise の生起条件は以下のように定義されている。

$$Hope_arise_{f_{H'}(d, l)}^i \varphi := Des_d^i \varphi \wedge Prob_arise_l^i \varphi \wedge l \neq 1$$

Jason で実装する際、変数 HD, D1, D2 を用いて論理式の各項を以下のように表現した。また、 $f_{H'}(d, l)$ は変数を受けとり、それを 2 で割る述語 hopedeg(HD, D1, D2) で表現している。

$$Hope_arise_{f_{H'}(d, l)}^i \varphi \equiv hope(X, Z)[degOfCert(HD)]$$

$$Des_d^i \varphi \equiv des(X, Z)[degOfCert(D1)]$$

$$Prob_arise_l^i \varphi \equiv prob_arise(X, Z)[degOfCert(D2)]$$

$$l \neq 1 \equiv D2 \neq 1$$

HD: Hope の度合い D1: X が Z を望む度合い D2: X が Z の可能性を信じる度合い

Fear

先行研究では、Fear_arise の生起条件は以下のように定義されている。

$$Fear_arise_{f_{F'}(d, l)}^i \varphi := Des_d^i \varphi \wedge Prob_arise_l^i \varphi \wedge l \neq 1$$

Jason で実装する際、変数 FD, D1, D2 を用いて論理式の各項を以下のように表現した。1 から、X にとっての Z の望ましさの度合い D1 を引いた値をイベントの望ましくなさの度合いとする。また、 $f_{F'}(d, l)$ は変数を受けとり、 $(1 - D1)$ と D2 の和を 2 で割る述語 feardeg(FD, D1, D2) で表現している。

$$Fear_arise_{f_{F'}(d, l)}^i \varphi \equiv fear(X, Z)[degOfCert(FD)]$$

$$Des_d^i \varphi \equiv des(X, Z)[degOfCert(D1)]$$

$$Prob_arise^i_l \varphi \equiv prob_arise(X,Z)[degOfCert(D2)]$$

$$l \neq 1 \equiv D2 \neq 1$$

FD: *Fear* の度合い D1: X が Z を望む度合い D2: X が Z の可能性を信じる度合い

7.3.4 Confirmation emotions

Satisfaction

先行研究では、Satisfaction_arise の生起条件は以下のように定義されている。

$$Satisfaction_arise^i_{f_{SV}(x,e,l)} \varphi := Bel^i PHope^i_x \varphi \wedge Bel^i PEffort^i_e \varphi \wedge Prob_arise^i_l \varphi$$

Jason で実装する際、変数 SD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。 $f_{SV}(x, e, l)$ は変数を受けとり、その総和を 3 で割る述語 $satisdeg(SD, D1, D2, D3)$ で表現している。この論理式の定義には、 $Prob_arise^i_l \varphi$ の度合い l が 1 でないという制限がないため、度合いが 1 の場合も起こりうることになる。 $Prob_arise^i_l \varphi$ の度合いが 1 とは、 $Bel_arise^i \varphi$ と同じことであるため、 $Bel_arise^i \varphi$ の場合も実装した。この場合、 $Prob_arise$ の度合い D2 に 1 を代入することで、同じ述語で度合いを計算するようにした。

$Bel^i PHope^i_x \varphi$ は「現在を含む過去のある時点で i が φ を hope した」ことを i が信じているという論理式である。hope が生起されたときに time 述語に記憶された時間が、Jason で標準に用意されている関数 $system.time$ で取得した現在時刻と同じかそれより古ければ、この hope は現在を含む過去のある時点で生起された感情とした。自身の感情は自覚しているため、Bel オペレータの実装は省略した。

$Bel^i PEffort^i_e \varphi$ は「現在を含む過去のある時点で i が φ を真にするために e 程度努力した」ことを、 i が信じているという論理式である。この信念は、 $bel(X, time(effort, 生起主, 対象, effort \text{ の度合い}, effort \text{ が追加された時間}))$ を外部入力から追加することで表現する。Bel 付きの信念が追加されたときには、信念の内容と時間を記憶する time 述語が追加される。そのため、 $time(time(effort, 生起主, 対象, \text{度合い}, effort \text{ が追加された時間}), Bel \text{ 付きの信念が追加された時間})$ という信念が追加されることになる。effort が追加された時間が、 $system.time$ で取得した現在時刻と同じかそれより古ければ、この effort は現在を含む過去のある時点で追加された信念とした。Bel 付きの信念が追加された時間は任意であるため、無名変数「_」で表している。

$$Satisfaction_arise^i_{f_{SV}(x,e,l)} \varphi \equiv satisfaction(X,Z)[degOfCert(SD)]$$

$$Bel^i PHope^i_x \varphi \equiv time(hope, X, Z, D1, CT) \& CT1 = system.time \& CT1 \geq CT$$

$$Bel^i PEffort^i_e \varphi \equiv time(time(effort, X, Z, D3, T), _) \& CT1 = system.time \& CT1 \geq T$$

$$Prob_arise^i_l \varphi \equiv ((prob_arise(X,Z)[degOfCert(D2)] \& satisdeg(SD, D1, D2, D3))$$

$$| (bel_arise(X,Z) \& D2 = 1 \& satisdeg(SD, D1, D2, D3)))$$

SD: *Satisfaction* の度合い D1: X が Z を望む度合い D2: Z が真になるだろうと X が信じる度合い D3: X が Z を真にするために努力する度合い

FearConfirmed

先行研究では、Fearconfirmed_arise の生起条件は以下のように定義されている。

$$FearConfirmed_arise^i_{f_{Fd}(x,e,l)} \varphi := Bel^i PFear^i_x \varphi \wedge Bel^i PEffort^i_e \neg \varphi \wedge Prob_arise^i_l \varphi$$

Jason で実装する際、変数 FCD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。また、 $f_{Fd}(x, e, l)$ は変数を受けとり、その総和を 3 で割る述語 $fearcondeg(FCD, D1, D2, D3)$ で表現している。 $Bel^i PFear^i_x \varphi$

と $Bel^i PEffort_e^i \neg\varphi$ の P は、Satisfaction と同じように実装した。

$Bel^i PEffort_e^i \neg\varphi$ は、「現在を含む過去のある時点で φ が真にならないように i が努力した」ことを i が信じているという論理式である。真にならないように努力したということは、偽になるように努力したという解釈ができるため、その解釈で実装した。Z と Z1 を否定の関係にあるイベントとし、Z と Z1 に値が代入されてから、6.3 節で述べた $deny(Z,Z1)$ の規則に合わせて、Z と Z1 が否定の関係であることを確かめた。

$$\begin{aligned}
FearConfirmed_arise_{f_{FCD}(x,e,l)}^i \varphi &\equiv fearconfirmed(X,Z)[degOfCert(FCD)] \\
Bel^i PFear_x^i \varphi &\equiv time(fear,X,Z,D1,CT) \& CT1=system.time \& CT1 \geq CT \\
Bel^i PEffort_e^i \neg\varphi &\equiv time(time(effort,X,Z1,D3,T),-) \& CT1=system.time \& CT1 \geq T \\
Prob_arise_i^i \varphi &\equiv ((prob_arise(X,Z)[degOfCert(D2)] \& fearcondeg(FCD,D1,D2,D3)) \\
&|(bel_arise(X,Z) \& D2=1 \& fearcondeg(FCD,D1,D2,D3))) \& deny(Z,Z1) \\
FCD: FearConfirmed \text{ の度合い} \quad D1: X \text{ が } Z \text{ を恐れる度合い} \quad D2: Z \text{ が真になるだろうと } X \\
&\text{ が信じる度合い} \quad D3: X \text{ が } Z \text{ を真にならないように努力する度合い}
\end{aligned}$$

Relief

先行研究では、Relief_arise の生起条件は以下のように定義されている。Relief_arise は、 φ を恐れていて φ を避けるために努力していて、 φ が起きなかった可能性があると思える場合に $\neg\varphi$ に対して生起される感情である。しかし、すべて φ に対しての感情として実装しているため、意味を反対にした。つまり、 $\neg\varphi$ を恐れていて φ が真になるように努力して、 φ が起きた可能性があると思える場合に φ に対して生起される感情であると変更した。ただ、 φ 自体をなにかが起きなかったことを表すものとした場合には、本来の意味で Relief_arise が生起されるものになった。

$$Relief_arise_{f_{RR'}(x,e,l)}^i \varphi := Bel^i PFear_x^i \neg\varphi \wedge Bel^i PEffort_e^i \varphi \wedge Prob_arise_i^i \varphi$$

Jason で実装する際、変数 RD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。また、 $f_{RR'}(x,e,l)$ は変数を受けとり、その総和を 3 で割る述語 reliefdeg(RD,D1,D2,D3) で表現している。 $Bel^i PFear_x^i \neg\varphi$ と $Bel^i PEffort_e^i \varphi$ の P と $\neg\varphi$ は、FearConfirmed と同じように実装した。

$$\begin{aligned}
Relief_arise_{f_{RR'}(x,e,l)}^i \varphi &\equiv relief(X,Z)[degOfCert(RD)] \\
Bel^i PFear_x^i \neg\varphi &\equiv time(fear,X,Z1,D1,CT) \& CT1=system.time \& CT1 \geq CT \\
Bel^i PEffort_e^i \varphi &\equiv time(time(effort,X,Z,D3,T),-) \& CT1=system.time \& CT1 \geq T \\
Prob_arise_i^i \varphi &\equiv ((prob_arise(X,Z)[degOfCert(D2)] \& reliefdeg(RD,D1,D2,D3)) \\
&|(bel_arise(X,Z) \& D2=1 \& reliefdeg(RD,D1,D2,D3))) \& deny(Z,Z1) \\
RD: Relief \text{ の度合い} \quad D1: X \text{ が } Z \text{ を恐れる度合い} \quad D2: Z \text{ が真にならないだろうと } X \text{ が信じる} \\
&\text{ 度合い} \quad D3: X \text{ が } Z \text{ を真にならないように努力する度合い}
\end{aligned}$$

Disappointment

先行研究では、Disappointment_arise の生起条件は以下のように定義されている。Disappointment_arise は、Relief_arise 同様、意味を反対にして定義した。

$$Disappointment_arise_{f_{DS'}(x,e,l)}^i \varphi := Bel^i PHope_x^i \neg\varphi \wedge Bel^i PEffort_e^i \neg\varphi \wedge Prob_arise_i^i \varphi$$

Jason で実装する際、変数 DPD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。また、 $f_{Ds}(x, e, l)$ は変数を受けとり、その総和を 3 で割る述語 $\text{dispointdeg}(\text{DPD}, \text{D1}, \text{D2}, \text{D3})$ で表現している。 $\text{Bel}^i \text{PHope}_x^i \neg \varphi$ と $\text{Bel}^i \text{PEffort}_e^i \neg \varphi$ の P は、Satisfaction と同じように実装した。 $\neg \varphi$ は、FearConfirmed と同じように実装した。

$$\text{Disappointment_arise}_{f_{Ds}(x,e,l)}^i \varphi \equiv \text{disappointment}(X, Z)[\text{degOfCert}(\text{DPD})]$$

$$\text{Bel}^i \text{PHope}_x^i \neg \varphi \equiv \text{time}(\text{hope}, X, Z1, \text{D1}, \text{CT}) \& \text{CT1} = \text{system.time} \& \text{CT1} \geq \text{CT}$$

$$\text{Bel}^i \text{PEffort}_e^i \neg \varphi \equiv \text{time}(\text{time}(\text{effort}, X, Z1, \text{D3}, \text{T}), -) \& \text{CT1} = \text{system.time} \& \text{CT1} \geq \text{T}$$

$$\text{Prob_arise}_i^i \varphi \equiv \text{prob_arise}(X, Z)[\text{degOfCert}(\text{D2})] \& \text{dispointdeg}(\text{DPD}, \text{D1}, \text{D2}, \text{D3})$$

$$|(\text{bel_arise}(X, Z) \& \text{D2} = 1 \& \text{dispointdeg}(\text{DPD}, \text{D1}, \text{D2}, \text{D3})) \& \text{deny}(Z, Z1)$$

DPD: *Disappointment* の度合い D1: X が Z を望む度合い D2: Z が真にならないだろうと X が信じる度合い D3: X が Z を真になるように努力する度合い

7.3.5 Attribution emotions

Pride

Pride_arise の生起条件は以下のように定義した。

$$\text{Pride_arise}_{f_{Pr}(p,l,a)}^i \varphi := \text{Praise}_p^i \varphi \wedge \text{Bel}^i \text{PProb}_i^i \varphi \wedge \text{Bel_arise}_i^i \varphi \wedge \text{Bel}^i \text{PEffort}_a^i \varphi \wedge l \neq 1$$

Jason で実装する際、変数 PD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。 i が φ を真だと信じない度合いを $(1 - \text{D1})$ で表現している。 $f_{Pr}(p, l, a)$ は変数を受けとり、 $(1 - \text{D1})$ と D2 と D3 の和を 3 で割る述語 $\text{prideg}(\text{PD}, \text{D1}, \text{D2}, \text{D3})$ で表現している。 $\text{Bel}^i \text{PEffort}_a^i \varphi$ は、Satisfaction と同じように実装した。

$\text{Bel}^i \text{PProb}_i^i \varphi$ は、「現在を含む過去のある時点で i が φ を l 程度信じた」ことを i が信じているという論理式である。自身の Prob 付き信念は自覚しているとして、Bel オペレータの実装は省略した。この信念は、 $\text{prob}(X, Z)[\text{degOfCert}(\text{D1})]$ を追加することで表現する。Prob 付きの信念が追加されたときは、time 述語の引数に Prob、生起主、度合い付きの信念の内容、追加された時間、度合いを記憶するようにしている。その追加された時間が system.time で取得した現在時刻と同じかそれより古ければ、現在を含む過去のある時点の信念であるとした。

$$\text{Pride_arise}_{f_{Pr}(p,l,a)}^i \varphi \equiv \text{pride}(X, Z)[\text{degOfCert}(\text{PD})]$$

$$\text{Praise}_p^i \varphi \equiv \text{praise}(X, Z)[\text{degOfCert}(\text{D2})]$$

$$\text{Bel}^i \text{PProb}_i^i \varphi \equiv \text{time}(\text{prob}, X, Z, \text{CT}, \text{D1}) \& \text{CT1} = \text{system.time} \& \text{CT1} \geq \text{CT}$$

$$\text{Bel_arise}_i^i \varphi \equiv \text{bel_arise}(X, Z)$$

$$\text{Bel}^i \text{PEffort}_a^i \varphi \equiv \text{time}(\text{time}(\text{effort}, X, Z, \text{D3}, \text{T}), -)$$

$$l \neq 1 \equiv \text{D1} \neq 1$$

PD: *Pride* の度合い D1: Z が真になるだろうと X が信じる度合い D2: X が Z を真にすることが評価される度合い D3: X が Z を真にするために努力する度合い

Shame

Shame_arise の生起条件は以下のように定義した。

$$\text{Shame_arise}_{f_{Sh}(p,l,a)}^i \varphi := \text{Praise}_p^i \varphi \wedge \text{Bel}^i \text{PProb}_i^i \varphi \wedge \text{Bel_arise}_i^i \varphi \wedge \text{Bel}^i \text{PEffort}_a^i \varphi \wedge l \neq 1$$

Jason で実装する際、変数 SMD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。 i が φ を真だと信じない度合いを $(1 - D1)$ で表現している。また、 φ を真にすることが評価されない度合いを $(1 - D2)$ で表現している。 $f_{SH}(p, l, a)$ は変数を受けとり、 $(1 - D1)$ と $(1 - D2)$ と D3 の和を 3 で割る述語 $\text{shamedeg}(SMD, D1, D2, D3)$ で表現している。 $Bel^i PProb_i^i \varphi$ と $Bel^i PEffort_a^i \varphi$ は Pride と同じように実装した。

$$Shame_arise_{f_{SH}(p,l,a)}^i \varphi \equiv \text{shame}(X, Z)[\text{degOfCert}(SMD)]$$

$$Praise_p^i \varphi \equiv \text{praise}(X, Z)[\text{degOfCert}(D2)]$$

$$Bel^i PProb_i^i \varphi \equiv \text{time}(\text{prob}, X, Z, CT, D1) \& CT1 = \text{system.time} \& CT1 \geq CT$$

$$Bel_arise^i \varphi \equiv \text{bel_arise}(X, Z)$$

$$Bel^i PEffort_a^i \varphi \equiv \text{time}(\text{time}(\text{effort}, X, Z, D3, T), -)$$

$$l \neq 1 \equiv D1 \neq 1$$

SMD: *Shame* の度合い D1: Z が真になるだろうと X が信じる度合い D2: X が Z を真にすることが評価される度合い D3: X が Z を真にするために努力する度合い

Admiration

Admiration_arise の生起条件は以下のように定義した。

$$Admiration_arise_{f_{Ad}(p,l,a)}^{i,j} \varphi := Bel^i Praise_p^j \varphi \wedge Bel^i PProb_i^i \varphi \wedge Bel_arise^i \varphi \wedge Bel^i PEffort_a^j \varphi \wedge l \neq 1$$

Jason で実装する際、変数 AD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。 i が φ を真だと信じない度合いを $(1 - D1)$ で表現している。 $f_{Ad}(p, l, a)$ は変数を受けとり、 $(1 - D1)$ と D2 と D3 の総和を 3 で割る述語 $\text{admideg}(AD, D1, D2, D3)$ で表現している。 $Bel^i PProb_i^i \varphi$ と $Bel^i PEffort_a^j \varphi$ は、Pride と同じように実装した。

$Bel^i Praise_p^j \varphi$ は、「 j が φ を真にすることは、 p で評価される」と、 i が信じているという論理式である。この信念は、 $\text{bel}(X, \text{praise}(Y, Z)[\text{degOfCert}(D2)])$ を外部入力から追加することで表現する。Bel 付きの信念が追加されたときには、信念の内容と時間を記憶する time 述語が追加される。 $\text{time}(\text{praise}(Y, Z)[\text{degOfCert}(D2)])$, Bel 付きの信念が追加された時間」という信念が追加されることになる。Bel 付きの信念が追加された時間は任意であるため、無名変数「 $-$ 」で表している。

$$Admiration_arise_{f_{Ad}(p,l,a)}^{i,j} \varphi \equiv \text{admiration}(X, Y, Z)[\text{degOfCert}(AD)]$$

$$Bel^i Praise_p^j \varphi \equiv \text{time}(\text{praise}(Y, Z)[\text{degOfCert}(D2)], -)$$

$$Bel^i PProb_i^i \varphi \equiv \text{time}(\text{prob}, X, Z, CT, D1) \& CT1 = \text{system.time} \& CT1 \geq CT$$

$$Bel^i PEffort_a^j \varphi \equiv \text{time}(\text{time}(\text{effort}, Y, Z, D3, T), -) \& CT1 = \text{system.time} \& CT1 \geq T$$

$$Bel_arise^i \varphi \equiv \text{bel_arise}(X, Z)$$

$$l \neq 1 \equiv D1 \neq 1$$

AD: *Admiration* の度合い D1: Z が真になるだろうと Y が信じる度合い D2: Y が Z を真にすることが評価される度合い D3: Y が Z を真にするために努力する度合い

Reproach

Reproach_arise の生起条件は以下のように定義した。

$$\text{Reproach_arise}_{f_{RP}(p,l,a)}^{i,j} \varphi := \text{Bel}^i \text{Praise}_p^j \varphi \wedge \text{Bel}^i \text{PProb}_l^i \varphi \wedge \text{Bel_arise}^i \varphi \wedge \text{Bel}^i \text{PEffort}_a^j \varphi \wedge l \neq 1$$

Jason で実装する際、変数 RPD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。 i が φ を真だと信じない度合いを $(1 - D1)$ で表現している。また、 φ を真にすることが評価されない度合いを $(1 - D2)$ で表現している。 $f_{RP}(p, l, a)$ は変数を受けとり、 $(1 - D1)$ と $(1 - D2)$ と $D3$ の和を 3 で割る述語 $\text{reprodeg}(RPD, D1, D2, D3)$ で表現している。 $\text{Bel}^i \text{PProb}_l^i \varphi$ と $\text{Bel}^i \text{PEffort}_a^j \varphi$ と $\text{Bel}^i \text{Praise}_p^j \varphi$ は Admiration と同じように実装した。

$$\text{Reproach_arise}_{f_{RP}(p,l,a)}^{i,j} \varphi \equiv \text{reproach}(X, Y, Z) [\text{degOfCert}(RPD)]$$

$$\text{Bel}^i \text{Praise}_p^j \varphi \equiv \text{time}(\text{praise}(Y, Z) [\text{degOfCert}(D2)], -)$$

$$\text{Bel}^i \text{PProb}_l^i \varphi \equiv \text{time}(\text{prob}, X, Z, CT, D1) \& CT1 = \text{system.time} \& CT1 \geq CT$$

$$\text{Bel}^i \text{PEffort}_a^j \varphi \equiv \text{time}(\text{time}(\text{effort}, Y, Z, D3, T), -) \& CT1 = \text{system.time} \& CT1 \geq T$$

$$\text{Bel_arise}^i \varphi \equiv \text{bel_arise}(X, Z)$$

$$l \neq 1 \equiv D1 \neq 1$$

RPD: *Reproach* の度合い D1: Z が真になるだろうと Y が信じる度合い D2: Y が Z を真にすることが評価される度合い
D3: Y が Z を真にするために努力する度合い

7.3.6 Well-being/Attribution compounds emotions

Gratification

先行研究では、Gratification_arise の生起条件は以下のように定義されている。

$$\text{Gratification_arise}_{f_{Grf}(x,y)}^i \varphi := \text{Joy_arise}_x^i \varphi \wedge \text{Pride_arise}_y^i \varphi$$

Jason で実装する際、変数 GFD, JD, PD を用いて論理式の各項を以下のように表現した。また、 $f_{Grf}(x, y)$ は変数を受けとり、JD と PD の和を 2 で割る述語 $\text{gratideg}(GFD, JD, PD)$ で表現している。

$$\text{Gratification_arise}_{f_{Grf}(x,y)}^i \varphi \equiv \text{gratification}(X, Z) [\text{degOfCert}(GFD)]$$

$$\text{Joy_arise}_x^i \varphi \equiv \text{joy}(X, Z) [\text{degOfCert}(JD)]$$

$$\text{Pride_arise}_y^i \varphi \equiv \text{pride}(X, Z) [\text{degOfCert}(PD)]$$

GFD: *Gratification* の度合い JD: X が Z について喜ぶ度合い PD: X が Z に誇りを持つ度合い

Remorse

先行研究では、Remorse_arise の生起条件は以下のように定義されている。

$$\text{Remorse_arise}_{f_{Rm}(x,y)}^i \varphi := \text{Distress_arise}_x^i \varphi \wedge \text{Shame_arise}_y^i \varphi$$

Jason で実装する際、変数 RMD, DD, SMD を用いて論理式の各項を以下のように表現した。また、 $f_{Rm}(x, y)$ は変数を受けとり、SMD と DD の和を 2 で割る述語 $\text{remodeg}(RMD, DD, SMD)$ で表現している。

$$Remorse_arise_{f_{Rm'}(x,y)}^i \varphi \equiv remorse(X,Z)[degOfCert(RMD)]$$

$$Distress_arise_x^i \varphi \equiv distress(X,Z)[degOfCert(DD)]$$

$$Shame_arise_y^i \varphi \equiv shame(X,Z)[degOfCert(SMD)]$$

RMD: *Remorse* の度合い DD: X が Z について悲しむ度合い SMD: X が Z を恥じる度合い

Gratitude

先行研究では、Gratitude_arise の生起条件は以下のように定義されている。

$$Gratitude_arise_{f_{Grt'}(x,y)}^{i,j} \varphi := Joy_arise_x^i \varphi \wedge Admiration_arise_y^{i,j} \varphi$$

Jason で実装する際、変数 GTD, JD, AD を用いて論理式の各項を以下のように表現した。また、 $f_{Grt'}(x, y)$ は変数を受けとり、AD と JD の和を 2 で割る述語 $gratitdeg(GTD,JD,AD)$ で表現している。

$$Gratitude_arise_{f_{Grt'}(x,y)}^{i,j} \varphi \equiv gratitude(X,Y,Z)[degOfCert(GTD)]$$

$$Joy_arise_x^i \varphi \equiv joy(X,Z)[degOfCert(JD)]$$

$$Admiration_arise_y^{i,j} \varphi \equiv admiration(X,Y,Z)[degOfCert(AD)]$$

GTD: *Gratitude* の度合い JD: X が Z について喜ぶ度合い AD: X が Y の Z について賞賛する度合い

Anger

先行研究では、Anger_arise の生起条件は以下のように定義されている。

$$Anger_arise_{f_{An'}(x,y)}^{i,j} \varphi := Distress_arise_x^i \varphi \wedge Reproach_arise_y^{i,j} \varphi$$

Jason で実装する際、変数 AGD, DD, RD を用いて論理式の各項を以下のように表現した。また、 $f_{An'}(x, y)$ は変数を受けとり、RD と DD の和を 2 で割る述語 $angerdeg(AGD,DD,RD)$ で表現している。

$$Anger_arise_{f_{An'}(x,y)}^{i,j} \varphi \equiv anger(X,Y,Z)[degOfCert(AGD)]$$

$$Distress_arise_x^i \varphi \equiv distress(X,Z)[degOfCert(DD)]$$

$$Reproach_arise_y^{i,j} \varphi \equiv reproach(X,Y,Z)[degOfCert(RD)]$$

AGD: *Anger* の度合い DD: X が Z について悲しむ度合い RD: X が Y の Z について批判する度合い

7.3.7 Attraction emotions

Love

先行研究では、Love の生起条件は以下のように定義されている。

$$Love_{f_L(a,f)}^i j := Appealing(i, j, a) \wedge Familiar(i, j, f)$$

Jason で実装する際、変数 LD, APD, FMD を用いて論理式の各項を以下のように表現した。また、 $f_L(a, f)$ は変数を受けとり、APD と FMD の和を 2 で割る述語 $lovedeg(LD,APD,FMD)$ で表現している。

$$Love_{f_L(a,f)}^i j \equiv love(X,Y)[degOfCert(LD)]$$

$Appealing(i, j, a) \equiv appealing(Y)[degOfCert(APD)]$

$Familiar(i, j, f) \equiv familiar(Y)[degOfCert(FMD)]$

LD: *Love* の度合い APD: X が Y を魅力に感じる度合い FMD: X が Y と接触する度合い

Hate

先行研究では、Hate の生起条件は以下のように定義されている。

$$Hate_{f_{Ha}(a,f)}^i j := Unappealing(i, j, a) \wedge Familiar(i, j, f)$$

Jason で実装する際、変数 HAD, UAD, FMD を用いて論理式の各項を以下のように表現した。また、 $f_{Ha}(a, f)$ は変数を受けとり、UAD と FMD の和を 2 で割る述語 $hatedeg(HAD, UAD, FMD)$ で表現している。

$Hate_{f_{Ha}(a,f)}^i j \equiv hate(X, Y)[degOfCert(HAD)]$

$Unappealing(i, j, a) \equiv unappealing(Y)[degOfCert(UAD)]$

$Familiar(i, j, f) \equiv familiar(Y)[degOfCert(FMD)]$

HAD: *Hate* の度合い UAD: X が Y を魅力に感じない度合い FMD: X が Y と接触する度合い

7.3.8 推測感情

他者の感情を推測する推測感情について実装した。推測感情は、他者の信念生起を認識することで感情を推測するものとしている。自身の感情の生起条件に、 Bel^i がついたものが、推測感情の生起条件となる。他者の信念は chat 機能で伝えられるため、Bel 付きの信念が追加されることと同じである。そのため、自身の感情生起において Bel 付きの信念が追加されたときと同じように実装した。それぞれの推測感情の度合いを計算する関数は、自身の感情生起で用いた関数と同じものである。

Guess_Joy_arise

先行研究では、Guess_Joy_arise の生起条件は以下のように定義されている。

$$Guess_Joy_arise_{f_{J'}(d)}^{i,j} \varphi := Bel^i Des_d^j \varphi \wedge Bel^i Bel_arise^j \varphi$$

Jason で実装する際、変数 GJD, D を用いて論理式の各項を以下のように表現した。

$Guess_Joy_arise_{f_{J'}(d)}^{i,j} \varphi \equiv guess_joy(X, Y, Z)[degOfCert(GJD)]$

$Bel^i Des_d^j \varphi \equiv time(des(Y, Z)[degOfCert(D)], -)$

$Bel^i Bel_arise^j \varphi \equiv time(bel_arise(Y, Z), -)$

GJD: *Guess_Joy* の度合い D: Y が Z を望む度合い

Guess_Distress_arise

先行研究では、Guess_Distress_arise の生起条件は以下のように定義されている。

$$Guess_Distress_arise_{f_{D'}(d)}^{i,j} \varphi := Bel^i Des_d^j \varphi \wedge Bel^i Bel_arise^j \varphi$$

Jason で実装する際、変数 GDD, D を用いて論理式の各項を以下のように表現した。

$Guess_Distress_arise_{fD'(d)}^{i,j}\varphi \equiv guess_distress(X,Y,Z)[degOfCert(GDD)]$

$Bel^i Des_d^j \varphi \equiv time(des(Y,Z)[degOfCert(D)],-)$

$Bel^i Bel_arise^j \varphi \equiv time(bel_arise(Y,Z),-)$

GDD: *Guess_Distress* の度合い D: Y が Z を望む度合い

Guess_Hope_arise

先行研究では、Guess_Hope_arise の生起条件は以下のように定義されている。

$$Guess_Hope_arise_{fH'(d,l)}^{i,j}\varphi := Bel^i Des_d^j \varphi \wedge Bel^i Prob_arise_l^j \varphi$$

Jason で実装する際、変数 GHD, D1, D2 を用いて論理式の各項を以下のように表現した。

$Guess_Hope_arise_{fH'(d,l)}^{i,j}\varphi \equiv guess_hope(X,Y,Z)[degOfCert(GHD)]$

$Bel^i Des_d^j \varphi \equiv time(des(Y,Z)[degOfCert(D1)],-)$

$Bel^i Prob_arise_l^j \varphi \equiv time(prob_arise(Y,Z)[degOfCert(D2)],-)$

GHD: *Guess_Hope* の度合い D1: Y が Z を望む度合い D2: Y が Z の可能性を信じる度合い

Guess_Fear_arise

先行研究では、Guess_Fear_arise の生起条件は以下のように定義されている。

$$Guess_Fear_arise_{fF'(d,l)}^{i,j}\varphi := Bel^i Des_d^j \varphi \wedge Bel^i Prob_arise_l^j \varphi$$

Jason で実装する際、変数 GFD, D1, D2 を用いて論理式の各項を以下のように表現した。

$Guess_Fear_arise_{fF'(d,l)}^{i,j}\varphi \equiv guess_fear(X,Y,Z)[degOfCert(GFD)]$

$Bel^i Des_d^j \varphi \equiv time(des(Y,Z)[degOfCert(D1)],-)$

$Bel^i Prob_arise_l^j \varphi \equiv time(prob_arise(Y,Z)[degOfCert(D2)],-)$

GFD: *Guess_Fear* の度合い D1: Y が Z を望む度合い D2: Y が Z の可能性を信じる度合い

Guess_Satisfaction_arise

先行研究では、Guess_Satisfactoin_arise の生起条件は以下のように定義されている。

$$Guess_Satisfaction_arise_{fSv'(x,e,l)}^{i,j}\varphi := Bel^i Bel^j PHope_x^j \varphi \wedge Bel^i Bel^j PEffort_e^j \varphi \wedge Bel^i Prob_arise_l^j \varphi$$

Jason で実装する際、変数 GSD, HD, D1, D2 を用いて論理式の各項を以下のように表現した。

$Bel^i Bel^j PHope_x^j \varphi$ は「現在を含む過去のある時点で j が φ を hope していたことを j が信じている」ことを i が信じているという論理式である。この信念は、bel(推測主, bel(生起主, time(hope, 生起主, 対象, 度合い, hope が生起された時間))) を外部入力から追加することで表現する。Bel 付きの信念が追加されたときには、信念の内容と追加された時間が time 述語に記憶されるので、time(bel(生起主, time(hope, 生起主, 対象, 度合い, hope が生起された時間)), Bel 付きの信念が追加された時間) という信念が追加されることになる。ここで、hope が生起された時間が、system.time で取得した現在時刻と同じかそれより古ければ、現在を含む過去のある時点で Hope が生起されたことを表す。

$Bel^i Bel^j PEffort_e^j \varphi$ は、「現在を含む過去のある時点で j が φ を真にするために e 程度努力したことを j が信じている」ということを i が信じているという論理式である。この信念は、 $bel(\text{推測主}, bel(\text{生起主}, \text{time}(\text{effort}, \text{生起主}, \text{対象}, \text{度合い}, \text{effort が追加された時間})))$ を外部入力から追加することで表現する。Bel 付きの信念が追加されたときには、信念の内容と追加された時間を記憶する time 述語が追加される。そのため、 $\text{time}(bel(\text{生起主}, \text{time}(\text{effort}, \text{生起主}, \text{対象}, \text{度合い}, \text{effort が追加された時間})))$, Bel が追加された時間」という信念が追加されることになる。ここで、 effort が追加された時間が、 system.time で取得した現在時刻と同じかそれより古ければ、現在を含む過去のある時点で effort が追加されたことを表す。

$$\begin{aligned} Guess_Satisfaction_arise_{f_{S^i}(x,e,l)}^{i,j} \varphi &\equiv \text{guess_satisfaction}(X,Y,Z)[\text{degOfCert}(\text{GSD})] \\ Bel^i Bel^j PHope_x^j \varphi &\equiv \text{time}(bel(Y,\text{time}(\text{hope},Y,Z,\text{HD},\text{CT})),_)\& \text{CT1}=\text{system.time} \& \text{CT1} \geq \text{CT} \\ Bel^i Bel^j PEffort_e^j \varphi &\equiv \text{time}(bel(Y,\text{time}(\text{effort},Y,Z,\text{D2},\text{T})),_)\& \text{CT1}=\text{system.time} \& \text{CT1} \geq \text{T} \\ Bel^i Prob_arise_l^j \varphi &\equiv ((\text{time}(\text{prob_arise}(Y,Z)[\text{degOfCert}(\text{D1})],_)\& \text{satisdeg}(\text{SD},\text{HD},\text{D1},\text{D2})) \\ &|(\text{time}(bel_arise(Y,Z),_)\& \text{D1}=1\& \text{satisdeg}(\text{SD},\text{HD},\text{D1},\text{D2}))) \\ \text{GSD: } &Guess_Satisfaction \text{ の度合い} \quad \text{HD: } Y \text{ が } Z \text{ を望む度合い} \quad \text{D1: } Z \text{ が真になるだろうと } Y \\ &\text{が信じる度合い} \quad \text{D2: } Y \text{ が } Z \text{ を真にするために努力する度合い} \end{aligned}$$

Guess_FearConfirmed_arise

先行研究では、 $Guess_FearConfirmed_arise$ の生起条件は以下のように定義されている。

$$Guess_FearConfirmed_arise_{f_{F^i}(x,e,l)}^{i,j} \varphi := Bel^i Bel^j PFear_x^j \varphi \wedge Bel^i Bel^j PEffort_e^j \neg \varphi \wedge Bel^i Prob_arise_l^j \varphi$$

Jason で実装する際、変数 GFCD , FD , D1 , D2 を用いて論理式の各項を以下のように表現した。 $Guess_Satisfaction_arise$ と同様に実装した。

$$\begin{aligned} Guess_FearConfirmed_arise_{f_{F^i}(x,e,l)}^{i,j} \varphi &\equiv \text{guess_fearconfirmed}(X,Y,Z)[\text{degOfCert}(\text{GFCD})] \\ Bel^i Bel^j PFear_x^j \varphi &\equiv \text{time}(bel(Y,\text{time}(\text{fear},Y,Z,\text{FD},\text{CT})),_)\& \text{CT1}=\text{system.time}\& \text{CT1} \geq \text{CT} \\ Bel^i Bel^j PEffort_e^j \neg \varphi &\equiv \text{time}(bel(Y,\text{time}(\text{effort},Y,Z,\text{D2},\text{T})),_)\& \text{CT1}=\text{system.time}\& \text{CT1} \geq \text{T} \\ Bel^i Prob_arise_l^j \varphi &\equiv ((\text{time}(\text{prob_arise}(Y,Z)[\text{degOfCert}(\text{D1})],_)\& \text{fearcondeg}(\text{FCD},\text{FD},\text{D1},\text{D2})) \\ &|(\text{time}(bel_arise(Y,Z),_)\& \text{D4}=1\& \text{fearcondeg}(\text{FCD},\text{FD},\text{D1},\text{D2})))\& \text{deny}(Z,\text{Z1}) \\ \text{GFCD: } &Guess_Fearconfirmed \text{ の度合い} \quad \text{FD: } Y \text{ が } Z \text{ を恐れる度合い} \quad \text{D1: } Z \text{ が真になるだろ} \\ &\text{うと } Y \text{ が信じる度合い} \quad \text{D2: } X \text{ が } Z \text{ を真にならないように努力する度合い} \end{aligned}$$

Guess_Relief_arise

先行研究では、 $Guess_Relief_arise$ の生起条件は以下のように定義されている。

$$Guess_Relief_arise_{f_{R^i}(x,e,l)}^{i,j} \varphi := Bel^i Bel^j PFear_x^j \neg \varphi \wedge Bel^i Bel^j PEffort_e^j \varphi \wedge Bel^i Prob_arise_l^j \varphi$$

Jason で実装する際、変数 GRD , FD , D1 , D2 を用いて論理式の各項を以下のように表現した。 $Guess_Satisfaction_arise$ と同様に実装した。

$$\begin{aligned} Guess_Relief_arise_{f_{R^i}(x,e,l)}^{i,j} \varphi &\equiv \text{guess_relief}(X,Y,Z)[\text{degOfCert}(\text{GRD})] \\ Bel^i Bel^j PFear_x^j \neg \varphi &\equiv \text{time}(bel(Y,\text{time}(\text{fear},Y,Z,\text{D1},\text{FD},\text{CT})),_)\& \text{CT1}=\text{system.time}\& \text{CT1} \geq \text{CT} \\ Bel^i Bel^j PEffort_e^j \varphi &\equiv \text{time}(bel(Y,\text{time}(\text{effort},Y,Z,\text{D2},\text{T})),_)\& \text{CT1}=\text{system.time}\& \text{CT1} \geq \text{T} \end{aligned}$$

$Bel^i Prob_arise^j \varphi \equiv ((time(prob_arise(Y,Z)[degOfCert(D1)],-) \& reliefdeg(RD,FD,D1,D2))$
 $| (time(bel_arise(Y,Z),-) \& D4=1 \& reliefdeg(RD,FD,D1,D2))) \& deny(Z,Z1)$
 GRD: *Guess_Relief* の度合い FD: Y が Z を恐れる度合い D1: Z が真にならないだろうと Y
 が信じる度合い D2: X が Z を真にならないように努力する度合い

Guess_Disappointment_arise

先行研究では、Guess_Disappointment_arise の生起条件は以下のように定義されている。

$Guess_Disappointment_arise_{f_{D^s(x,e,l)}}^{i,j} \varphi := Bel^i Bel^j PHope_x^j \neg \varphi \wedge Bel^i Bel^j PEffort_e^j \neg \varphi \wedge Bel^i Prob_arise^j \varphi$

Jason で実装する際、変数 GDPD, HD, D1, D2 を用いて論理式の各項を以下のように表現した。
 Guess_Satisfaction_arise と同様に実装した。

$Guess_Disappointment_arise_{f_{D^s(x,e,l)}}^{i,j} \varphi \equiv guess_disappointment(X,Y,Z)[degOfCert(GDPD)]$
 $Bel^i Bel^j PHope_x^j \neg \varphi \equiv time(bel(Y,time(hope,Y,Z1,HD,CT)),-) \& CT1=system.time \& CT1 \geq CT$
 $Bel^i Bel^j PEffort_e^j \neg \varphi \equiv time(bel(Y,time(effort,Y,Z1,D2,T)),-) \& CT1=system.time \& CT1 \geq T$
 $Bel^i Prob_arise^j \varphi \equiv ((time(prob_arise(Y,Z)[degOfCert(D1)],-) \& dispointdeg(DPD,HD,D1,D2))$
 $| (time(bel_arise(Y,Z),-) \& D4=1 \& dispointdeg(DPD,HD,D1,D2))) \& deny(Z,Z1)$
 GDPD: *Guess_Disappointment* の度合い HD: Y が Z を望む度合い D1: Z が真にならない
 だろうと Y が信じる度合い D2: Y が Z を真にするために努力する度合い

Guess_Pride_arise

Guess_Pride_arise の生起条件は以下のように定義した。

$Guess_Pride_arise_{f_{P^r(p,l,a)}}^{i,j} \varphi := Bel^i Praise_p^j \varphi \wedge Bel^i Bel^j PProb_l^j \varphi \wedge Bel^i Bel_arise^j \varphi$
 $\wedge Bel^i Bel^j PEffort_a^j \varphi \wedge l \neq 1$

Jason で実装する際、変数 GPD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。

$Bel^i Bel^j PProb_l^j \varphi$ は、「現在を含む過去のある時点で j が φ を l 程度信じたことを j が信じる」というこ
 とを i が信じているという論理式である。これは、bel(推測主, bel(生起主, time(prob, 生起主, 対象, 度合い,
 Prob 付きの信念が追加された時間))) を外部入力で追加することで表現する。Bel 付きの信念が追加されたと
 きは、信念の内容と信念を記憶する time 述語を追加する。そのため、time(bel(生起主, time(prob, 生起主,
 対象, 度合い, prob が追加された時間)), Bel 付きの信念が追加された時間) という信念が追加されることにな
 る。prob が追加された時間が system.time で取得した現在時刻と同じかそれより古ければ、現在を含む過去
 のある時点の信念であるとする。 $Bel^i Bel^j PEffort_a^j \varphi$ は Guess_Satisfaction_arise と同様に実装した。

$Guess_Pride_arise_{f_{P^r(p,l,a)}}^{i,j} \varphi \equiv guess_pride(X,Y,Z)[degOfCert(GPD)]$
 $Bel^i Praise_p^j \varphi \equiv time(praise(Y,Z)[degOfCert(D2)],-)$
 $Bel^i Bel^j PProb_l^j \varphi \equiv time(bel(Y,time(prob,Y,Z,D1,CT)),-) \& CT1=system.time \& CT1 \geq CT$
 $Bel^i Bel_arise^j \varphi \equiv time(bel_arise(Y,Z),-)$
 $Bel^i Bel^j PEffort_a^j \varphi \equiv time(bel(Y,time(effort,Y,Z,D3,T)),-) \& CT1=system.time \& CT1 \geq T$

$l \neq 1 \equiv D1 \neq 1$

GPD: *Guess_Pride* の度合い D1: Z が真になるだろうと Y が信じる度合い D2: Y が Z を真にすることが評価される度合い D3: Y が Z を真にするために努力する度合い

Guess_Shame_arise

Guess_Shame_arise の生起条件は以下のように定義した。

$$\begin{aligned} \text{Guess_Shame_arise}_{fSh'(p,l,a)}^{i,j} \varphi &:= \text{Bel}^i \text{Praise}_p^j \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PProb}_1^j \varphi \wedge \text{Bel}^i \text{Bel_arise}^j \varphi \\ &\quad \wedge \text{Bel}^i \text{Bel}^j \text{PEffort}_a^j \varphi \wedge l \neq 1 \end{aligned}$$

Jason で実装する際、変数 GSMD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。
Guess_Pride_arise と同様に実装する。

$$\begin{aligned} \text{Guess_Shame_arise}_{fSh'(p,l,a)}^{i,j} \varphi &\equiv \text{guess_shame}(X,Y,Z)[\text{degOfCert}(\text{GSMD})] \\ \text{Bel}^i \text{Praise}_p^j \varphi &\equiv \text{time}(\text{praise}(Y,Z)[\text{degOfCert}(D2)],-) \\ \text{Bel}^i \text{Bel}^j \text{PProb}_1^j \varphi &\equiv \text{time}(\text{bel}(Y,\text{time}(\text{prob},Y,Z,D1,CT)),-)\& \text{CT1}=\text{system.time}\& \text{CT1} \geq \text{CT} \\ \text{Bel}^i \text{Bel_arise}^j \varphi &\equiv \text{time}(\text{bel_arise}(Y,Z),-) \\ \text{Bel}^i \text{Bel}^j \text{PEffort}_a^j \varphi &\equiv \text{time}(\text{bel}(Y,\text{time}(\text{effort},Y,Z,D3,T)),-)\& \text{CT1}=\text{system.time}\& \text{CT1} \geq T \\ l \neq 1 \equiv D1 \neq 1 \end{aligned}$$

GSMD: *Guess_Shame* の度合い D1: Z が真になるだろうと Y が信じる度合い D2: Y が Z を真にすることが評価される度合い D3: Y が Z を真にするために努力する度合い

Guess_Admiration_arise

Guess_Admiration_arise の生起条件は以下のように定義した。

$$\begin{aligned} \text{Guess_Admiration_arise}_{fAd'(p,l,a)}^{i,j,k} \varphi &:= \text{Bel}^i \text{Bel}^j \text{Praise}_p^k \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PProb}_1^j \varphi \\ &\quad \wedge \text{Bel}^i \text{Bel_arise}^j \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PEffort}_a^k \varphi \wedge l \neq 1 \end{aligned}$$

Jason で実装する際、変数 GAD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。
Guess_Pride_arise と同様に実装する。

$$\begin{aligned} \text{Guess_Admiration_arise}_{fAd'(p,l,a)}^{i,j,k} \varphi &\equiv \text{guess_admiration}(X,Y,W,Z)[\text{degOfCert}(\text{GAD})] \\ \text{Bel}^i \text{Bel}^j \text{Praise}_p^k \varphi &\equiv \text{time}(\text{bel}(Y,\text{praise}(W,Z)[\text{degOfCert}(D2)]),-) \\ \text{Bel}^i \text{Bel}^j \text{PProb}_1^j \varphi &\equiv \text{time}(\text{bel}(Y,\text{time}(\text{prob},Y,Z,D1,CT)),-)\& \text{CT1}=\text{system.time}\& \text{CT1} \geq \text{CT} \\ \text{Bel}^i \text{Bel_arise}^j \varphi &\equiv \text{time}(\text{bel_arise}(Y,Z),-) \\ \text{Bel}^i \text{Bel}^j \text{PEffort}_a^k \varphi &\equiv \text{time}(\text{bel}(Y,\text{time}(\text{effort},W,Z,D3,T)),-)\& \text{CT1}=\text{system.time}\& \text{CT1} \geq T \\ l \neq 1 \equiv D1 \neq 1 \end{aligned}$$

GAD: *Guess_Admiration* の度合い D1: Z が真になるだろうと Y が信じる度合い D2: W が Z を真にすることが評価される度合い D3: W が Z を真にするために努力する度合い

Guess_Reproach_arise

Guess_Reproach_arise の生起条件は以下のように定義した。

$$\begin{aligned} \text{Guess_Reproach_arise}_{f_{RP}(p,l,a)}^{i,j,k} \varphi &:= \text{Bel}^i \text{Bel}^j \text{Praise}_p^k \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PProb}_l^j \varphi \\ &\wedge \text{Bel}^i \text{Bel}^j \text{arise}^j \varphi \wedge \text{Bel}^i \text{Bel}^j \text{PEffort}_a^k \varphi \wedge l \neq 1 \end{aligned}$$

Jason で実装する際、変数 GRPD, D1, D2, D3 を用いて論理式の各項を以下のように表現した。
Guess_Pride_arise と同様に実装する。

$$\begin{aligned} \text{Guess_Reproach_arise}_{f_{RP}(p,l,a)}^{i,j,k} \varphi &\equiv \text{guess_reproach}(X,Y,W,Z)[\text{degOfCert}(\text{GRPD})] \\ \text{Bel}^i \text{Bel}^j \text{Praise}_p^k \varphi &\equiv \text{time}(\text{bel}(Y,\text{praise}(W,Z)[\text{degOfCert}(D2)]),-) \\ \text{Bel}^i \text{Bel}^j \text{PProb}_l^j \varphi &\equiv \text{time}(\text{bel}(Y,\text{time}(\text{prob},Y,Z,D1,CT)),-)\&CT1=\text{system.time}\&CT1 \geq CT \\ \text{Bel}^i \text{Bel}^j \text{arise}^j \varphi &\equiv \text{time}(\text{bel}.\text{arise}(Y,Z),-) \\ \text{Bel}^i \text{Bel}^j \text{PEffort}_a^k \varphi &\equiv \text{time}(\text{bel}(Y,\text{time}(\text{effort},W,Z,D3,T)),-)\&CT1=\text{system.time}\&CT1 \geq T \\ l \neq 1 &\equiv D1 \neq 1 \\ \text{GRPD: Guess_Reproach の度合い} \quad &D1: Z \text{ が真になるだろうと } Y \text{ が信じる度合い} \quad D2: W \text{ が} \\ &Z \text{ を真にすることが評価される度合い} \quad D3: W \text{ が } Z \text{ を真にするために努力する度合い} \end{aligned}$$

Guess_HappyFor_arise

先行研究では、Guess_HappyFor_arise の生起条件は以下のように定義されている。

$$\begin{aligned} \text{Guess_HappyFor_arise}_{f_{HF}(d_j,x,v,LD)}^{i,j,k} \varphi &:= \text{Bel}^i \text{Guess_Joy_arise}_x^{j,k} \varphi \wedge \text{Bel}^i \text{Des}_{d_j}^j \text{Bel}^k \varphi \\ &\wedge \text{Bel}^i \text{Bel}^j \text{Deserve}_v^k \varphi \wedge \text{Bel}^i \text{Love}_{LD}^j k \end{aligned}$$

Jason で実装する際、変数 GHFD, D1, D2, D3, D4 を用いて論理式の各項を以下のように表現した。

$$\begin{aligned} \text{Guess_HappyFor_arise}_{f_{HF}(d_j,x,v,LD)}^{i,j,k} \varphi &\equiv \text{guess_happyfor}(X,Y,W,Z)[\text{degOfCert}(\text{GHFD})] \\ \text{Bel}^i \text{Guess_Joy_arise}_x^{j,k} \varphi &\equiv \text{time}(\text{bel}(Y,\text{des}(W,Z)[\text{degOfCert}(D1)]),-)\&\text{time}(\text{bel}(Y,\text{bel}.\text{arise}(W,Z)),-) \\ \text{Bel}^i \text{Des}_{d_j}^j \text{Bel}^k \varphi &\equiv \text{time}(\text{des}(Y,\text{bel}(W,Z))[\text{degOfCert}(D2)],-) \\ \text{Bel}^i \text{Bel}^j \text{Deserve}_v^k \varphi &\equiv \text{time}(\text{bel}(Y,\text{deserve}(W,Z))[\text{degOfCert}(D3)],-) \\ \text{Bel}^i \text{Love}_{LD}^j k &\equiv \text{time}(\text{love}(Y,W))[\text{degOfCert}(D4)],-) \\ \text{GHFD: Guess_HappyFor の度合い} \quad &D1: \text{Guess_Joy の度合い} \quad D2: W \text{ が } Z \text{ を真だと信じる} \\ &\text{ことを } Y \text{ が望む度合い} \quad D3: W \text{ にとっての } Z \text{ のふさわしさ} \quad D4: Y \text{ が } W \text{ を好む度合い} \end{aligned}$$

Guess_SorryFor_arise

Guess_SorryFor_arise の生起条件は以下のように定義した。

$$\begin{aligned} \text{Guess_SorryFor_arise}_{f_{SF}(d_j,x,v,LD)}^{i,j,k} \varphi &:= \text{Bel}^i \text{Guess_Distress_arise}_x^{j,k} \varphi \wedge \text{Bel}^i \text{Des}_{d_j}^j \text{Bel}^k \varphi \\ &\wedge \text{Bel}^i \text{Bel}^j \text{Deserve}_v^k \varphi \wedge \text{Bel}^i \text{Love}_{LD}^j k \end{aligned}$$

Jason で実装する際、変数 GSFD, D1, D2, D3, D4 を用いて論理式の各項を以下のように表現した。

$Guess_SorryFor_arise_{f_{SF}(d_j,x,v,LD)}^{i,j,k} \varphi \equiv guess_sorryfor(X,Y,W,Z)[degOfCert(GSFD)]$
 $Bel^i Guess_Distress_arise_x^{j,k} \varphi \equiv time(bel(Y,des(W,Z)[degOfCert(D1)]),-)\&time(bel(Y,bel_arise(W,Z)),-)$
 $Bel^i Des_{d_j}^j Bel^k \varphi \equiv time(des(Y,bel(W,Z))[degOfCert(D2)],-)$
 $Bel^i Bel^j Deserve_v^k \varphi \equiv time(bel(Y,deserve(W,Z)[degOfCert(D3)]),-)$
 $Bel^i Love_{LD}^j k \equiv time(love(Y,W)[degOfCert(D4)],-)$
 GSFD: *Guess_SorryFor* の度合い D1: *Guess_Distress* の度合い D2: W が Z を真だと信じ
 ことを Y が望む度合い D3: W にとっての Z のふさわしさ D4: Y が W を好む度合い

Guess_Resentment_arise

先行研究では、Guess_Resentment_arise の生起条件は以下のように定義されている。

$$\begin{aligned}
 Guess_Resentment_arise_{f_{RS}(d_j,x,v,LD)}^{i,j,k} \varphi := & Bel^i Guess_Joy_arise_x^{j,k} \varphi \wedge Bel^i Des_{d_j}^j Bel^k \varphi \\
 & \wedge Bel^i Bel^j Deserve_v^k \varphi \wedge Bel^i Hate_{HD}^j k
 \end{aligned}$$

Jason で実装する際、変数 GRSD, D1, D2, D3, D4 を用いて論理式の各項を以下のように表現した。

$Guess_Resentment_arise_{f_{RS}(d_j,x,v,LD)}^{i,j,k} \varphi \equiv guess_resentment(X,Y,W,Z)[degOfCert(GRSD)]$
 $Bel^i Guess_Joy_arise_x^{j,k} \varphi \equiv time(bel(Y,des(W,Z)[degOfCert(D1)]),-)\&time(bel(Y,bel_arise(W,Z)),-)$
 $Bel^i Des_{d_j}^j Bel^k \varphi \equiv time(des(Y,bel(W,Z))[degOfCert(D2)],-)$
 $Bel^i Bel^j Deserve_v^k \varphi \equiv time(bel(Y,deserve(W,Z)[degOfCert(D3)]),-)$
 $Bel^i Hate_{HD}^j k \equiv time(hate(Y,W)[degOfCert(D4)],-)$
 GRSD: *Guess_Resentment* の度合い D1: *Guess_Joy* の度合い D2: W が Z を真だと信じる
 ことを Y が望む度合い D3: W にとっての Z のふさわしさ D4: Y が W を嫌う度合い

Guess_Gloating_arise

Guess_Gloating_arise の生起条件は以下のように定義した。

$$\begin{aligned}
 Guess_Gloating_arise_{f_{GV}(d_j,x,v,LD)}^{i,j,k} \varphi := & Bel^i Guess_Distress_arise_x^{j,k} \varphi \wedge Bel^i Des_{d_j}^j Bel^k \varphi \\
 & \wedge Bel^i Bel^j Deserve_v^k \varphi \wedge Bel^i Hate_{HD}^j k
 \end{aligned}$$

Jason で実装する際、変数 GGD, D1, D2, D3, D4 を用いて論理式の各項を以下のように表現した。

$Guess_Gloating_arise_{f_{GV}(d_j,x,v,LD)}^{i,j,k} \varphi \equiv guess_gloating(X,Y,W,Z)[degOfCert(GGD)]$
 $Bel^i Guess_Distress_arise_x^{j,k} \varphi \equiv time(bel(Y,des(W,Z)[degOfCert(D1)]),-)\&time(bel(Y,bel_arise(W,Z)),-)$
 $Bel^i Des_{d_j}^j Bel^k \varphi \equiv time(des(Y,bel(W,Z))[degOfCert(D2)],-)$
 $Bel^i Bel^j Deserve_v^k \varphi \equiv time(bel(Y,deserve(W,Z)[degOfCert(D3)]),-)$
 $Bel^i Hate_{HD}^j k \equiv time(hate(Y,W)[degOfCert(D4)],-)$
 GGD: *Guess_Gloating* の度合い D1: *Guess_Distress* の度合い D2: W が Z を真だと信じる
 ことを Y が望む度合い D3: W にとっての Z のふさわしさ D4: Y が W を嫌う度合い

Guess_Gratification_arise

先行研究では、Guess_Gratification_arise の生起条件は以下のように定義されている。

$$Guess_Gratification_arise_{f_{Grtf'}^{i,j}(x,y)}\varphi := Bel^i Joy_arise_x^j\varphi \wedge Bel^i Pride_arise_y^j\varphi$$

Jason で実装する際、変数 GGFD, JD, PD を用いて論理式の各項を以下のように表現した。 $Bel^i Joy_arise_x^j\varphi$ は $Guess_Joy_arise_x^{i,j}\varphi$ と同じ意味であり、 $Bel^i Pride_arise_y^j\varphi$ は $Guess_Pride_arise_y^{i,j}\varphi$ と同じ意味であるので、これらにより実装した。

$$Guess_Gratification_arise_{f_{Grtf'}^{i,j}(x,y)}\varphi \equiv guess_gratification(X,Y,Z)[degOfCert(GGFD)]$$

$$Bel^i Joy_arise_x^j\varphi \equiv guess_joy(X,Y,Z)[degOfCert(JD)]$$

$$Bel^i Pride_arise_y^j\varphi \equiv guess_pride(X,Y,Z)[degOfCert(PD)]$$

GGFD: *Guess_Gratification* の度合い JD: Y が Z について喜ぶ度合い PD: Y が Z に誇りを持つ度合い

Guess_Remorse_arise

先行研究では、Guess_Remorse_arise の生起条件は以下のように定義されている。

$$Guess_Remorse_arise_{f_{Rm'}^{i,j}(x,y)}\varphi := Bel^i Distress_arise_x^j\varphi \wedge Bel^i Shame_arise_y^j\varphi$$

Jason で実装する際、変数 GRMD, DD, SMD を用いて論理式の各項を以下のように表現した。 $Bel^i Distress_arise_x^j\varphi$ は $Guess_Distress_arise_x^{i,j}\varphi$ と同じ意味であり、 $Bel^i Shame_arise_y^j\varphi$ は $Guess_Shame_arise_y^{i,j}\varphi$ と同じ意味であるので、これらにより実装した。

$$Guess_Remorse_arise_{f_{Rm'}^{i,j}(x,y)}\varphi \equiv guess_remorse(X,Y,Z)[degOfCert(GRMD)]$$

$$Bel^i Distress_arise_x^j\varphi \equiv guess_distress(X,Y,Z)[degOfCert(DD)]$$

$$Bel^i Shame_arise_y^j\varphi \equiv guess_shame(X,Y,Z)[degOfCert(SMD)]$$

GRMD: *Guess_Remorse* の度合い DD: Y が Z について悲しむ度合い SMD: Y が Z を恥じる度合い

Guess_Gratitude_arise

先行研究では、Guess_Gratitude_arise の生起条件は以下のように定義されている。

$$Guess_Gratitude_arise_{f_{Grtt'}^{i,j,k}(x,y)}\varphi := Bel^i Joy_arise_x^j\varphi \wedge Bel^i Admiration_arise_y^{j,k}\varphi$$

Jason で実装する際、変数 GGTD, JD, AD を用いて論理式の各項を以下のように表現した。 $Bel^i Joy_arise_x^j\varphi$ は $Guess_Joy_arise_x^{i,j}\varphi$ と同じ意味であり、 $Bel^i Admiration_arise_y^{j,k}\varphi$ は $Guess_Admiration_arise_y^{i,j,k}\varphi$ と同じ意味であるので、これらにより実装した。

$$Guess_Gratitude_arise_{f_{Grtt'}^{i,j,k}(x,y)}\varphi \equiv guess_gratitude(X,Y,W,Z)[degOfCert(GGTD)]$$

$$Bel^i Joy_arise_x^j\varphi \equiv guess_joy(X,Y,Z)[degOfCert(JD)]$$

$$Bel^i Admiration_arise_y^{j,k}\varphi \equiv guess_admiration(X,Y,W,Z)[degOfCert(AD)]$$

GGTD: *Guess_Gratitude* の度合い JD: Y が Z について喜ぶ度合い AD: W が Z をしたことに Y が賞賛する度合い

Guess_Anger_arise

先行研究では、Guess_Anger_arise の生起条件は以下のように定義されている。

$$Guess_Anger_arise_{f_{An'}^{i,j,k}(x,y)}\varphi := Bel^i Distress_arise_x^j\varphi \wedge Bel^i Reproach_arise_y^{j,k}\varphi$$

Jason で実装する際、変数 GAGD, DD, RD を用いて論理式の各項を以下のように表現した。 $Bel^i Distress_arise_x^j\varphi$ は $Guess_Distress_arise_x^{i,j}\varphi$ と同じ意味であり、 $Bel^i Reproach_arise_y^{j,k}\varphi$ は $Guess_Reproach_arise_y^{i,j,k}\varphi$ と同じ意味であるので、これらにより実装した。

$$Guess_Anger_arise_{f_{An'}^{i,j,k}(x,y)}\varphi \equiv guess_anger(X,Y,W,Z)[degOfCert(GAGD)]$$

$$Bel^i Distress_arise_x^j\varphi \equiv guess_distress(X,Y,Z)[degOfCert(DD)]$$

$$Bel^i Reproach_arise_y^{j,k}\varphi \equiv guess_reproach(X,Y,W,Z)[degOfCert(RD)]$$

GAGD: *Guess_Anger* の度合い DD: Y が Z について悲しむ度合い RD: W が Z をしたこと
に Y が批判する度合い

8 新たな能力の追加

8.1 相反する感情生起の実装

これまでの実装では、joy と distress、fear と hope のように相反する感情は同時に生起されないようになっていた。これまでの実装での、相反する感情の生起条件は下のようになっていた。

$$Hope_{f_{H'}^{i}(p,l)}\varphi := Des_d^i\varphi \wedge Prob_i^i\varphi \wedge l \neq 1$$

$$Fear_{f_{F'}^{i}(p,l)}\varphi := Des_d^i\neg\varphi \wedge Prob_i^i\varphi \wedge l \neq 1$$

この生起条件によると、望んでいることが真になる可能性があるとき hope が生起され、望んでいないことが真になる可能性があるとき fear が生起されることになる。 $Des_d^i\varphi$ と $Des_d^i\neg\varphi$ は同時に真にならないため、hope と fear は同時に生起されない。しかし、先行研究 [7] によれば、相反する感情の強さが最大でないのであれば、生起されることは不自然でない。そこで、それを反映するよう実装を改めた。先行研究 [7] で定義されていた相反する感情の生起条件は、相反する感情どうし同じで、度合い計算方法が異なるだけである。そのため、それに基づいた実装においては、相反する感情が生起されるようになる。また、相反する感情の強さがそれぞれ最大、すなわち 1 である場合にはそれらが同時に生起しないのが妥当であるので、相反する感情それぞれの生起条件に、その相反する感情の度合いが 1 でないことを追加した。

相反する感情の強さが最大でないことの検査は、6.1 節で述べた time 述語を用いて実装した。例えば、hope の生起条件には、hope の相反する感情である fear の度合いが 1 であるという信念が存在しないことが必要になるため、not time(fear, 生起主, 対象, 1.0, _) を付け加えた。time 述語の第五引数は、fear が生起された時間であるが、任意であるためこのように無名変数「_」で表現した。すべての相反する感情について、この実装を付け加えた。

8.2 推測感情の時間経過による度合いの減衰の実装

これまでの実装では、他者の感情を推測する感情の、時間経過による減衰が実装されていなかった。自身の感情が減衰するのにもかかわらず他者の感情が減衰しないのは不自然であると考えられるため、これを実装した。自身の感情の時間経過による減衰と同じように、現在時刻と感情が生起された時間を引数に持つ減衰関数を用いて実装を行った。

9 検証実験

本研究の実装が、減衰を付加した以前の実装(時間経過による感情度合い減衰を考慮しない論理体系に基づいていた)に比べ、同等以上の能力を持っているかどうかを検証するため検証実験を行った。同等の能力というのは、実行結果が同じになることである。ここでは、シナリオの入力によりエージェントが以前と同じ感情を生起すること、時間経過により感情が減衰することを確認する。また、未実装であった、時間経過による推測感情の減衰や、相反する感情の生起の実装ができていることを確認する。これらを確認することで、本研究の実装が以前の実装に比べて同等以上の能力を持つことがいえる。

9.1 実験方法

まず、先行研究 [8] の検証実験で用いられていた「バターはどこへ溶けた (Where Has My Butter Gone?)」 [5] のストーリーから抜粋したものをを用いる。表 2 でグループ分けされたうち、全グループの感情を一つでも生起するため、この文学作品で検証実験を行うことを決定した。そのシナリオから抽出した信念を適宜エージェントに保持させていくことで、感情生起と時間経過による感情度合いの減衰を確認する。人間らしい感情生起の検証では、被験者を用いた心理学実験を行うことが望ましい。しかし、そのために必要な大規模な調査が困難であることから、先行研究では文学作品のストーリーを利用した検証実験を行っているので、本論文でも同作品を対象とする。

シナリオは、以前作成されていたものを利用した。シナリオから読み取れる他者への魅力と接触度合い、望ましさなどを初期信念としてエージェントに保持させる。信念や他者に伝える行動を適宜追加し、感情生起と減衰を確認する。検証実験は、先行研究 [6] で用意されていた、外部から TCP の 49999 番ポートに接続してエージェントへの入力ができる環境で行った。この検証は、以前の実装と同じように感情が生起し、減衰すること、新たな能力の実装を確認するために行うので、感情による行動選択の機能は扱わないことにする。

以前の実装の様子を表 3、表 4 に表す。表 3 は、先行研究 [8, p. 15] の表 2 より引用し、一部改変した。表 4 は、先行研究 [8, p. 16] の表 3 より引用し、一部改変した。以前の検証実験では、時間経過による感情度合い減衰の様子出力は省略されていたため、著者が減衰の出力を行った。また、以前の検証実験と今回の検証実験では、同じストーリーから抽出した信念を初期信念や外部入力としている。以前の検証実験と今回の検証実験では、抽出した信念やその度合いは異なるが、同じストーリーに基づいているため、問題はないとする。信念の度合いが異なるため、生起された感情の度合いが異なる場合がある。

表3 リスとキツネのやり取り

エージェント	追加した信念や行動	実際の入力
♣ キツネはバターを努力して探し見つかる可能性を感じており、上手く立ち回るのは困難なことだと思う		
キツネ	努力 (キツネ butter)(0.2)	fox effort(fox,butter)[degOfCert(0.2)]
キツネ	可能性 (キツネ butter)(0.2)	fox prob(fox,butter)[degOfCert(0.2)]
キツネ	○ Hope(望み) 出力: [fox] hope(fox,butter)0.4	
キツネ	現在の hope の度合いを出力するゴール	fox !act3(fox,butter)
キツネ	◇hope の度合い 0.38(生起時の度合い 0.4 から減衰した)	
キツネ	可能性が低い (キツネ butter_brain)(0.3)	fox prob(fox,not butter_brain)[degOfCert(0.3)]
♣ キツネはリスにバターが望ましいことを伝え、見つけたことを伝えられ、上手く立ち回れたと感じる		
キツネ → リス	望ましさを伝える (キツネ butter)(0.2)	fox !chat(squirrel, des(fox,butter)[degOfCert(0.2)])
リス → キツネ	成立を伝える (リス butter)	squirrel !chat(fox,bel(squirrel,butter))
キツネ	成立 (キツネ butter_brain)	fox bel(fox,butter_brain)
キツネ	○ Pride(誇り) 出力: [fox] pride(fox,butter_brain)0.6	
キツネ	現在の pride の度合いを出力するゴール	fox !act5(fox,butter_brain)
キツネ	◇ pride の度合い 0.591364(生起時の度合い 0.6 から減衰した)	
♣ キツネはリスに連れていかれ大きなバターを見て、バターが腐っていると嘘をつく		
キツネ	成立 (キツネ butter)	fox bel(fox,butter)
キツネ	○ Joy(喜び) 出力: [fox] joy(fox,butter)0.2 ○ Satisfaction(満足) 出力: [fox] satisfaction(fox,butter)0.6	
キツネ → リス	成立を伝える (キツネ butter_rot)	fox !chat(squirrel,bel(fox,butter_rot))
リス	推測 (望ましくない (キツネ butter_rot))(0.2)	squirrel bel(squirrel, des(fox,not butter_rot)[degOfCert(0.2)])
リス	○ Guess_Distress(悲しみ推測) 出力: [squirrel] guess_distress(squirrel,fox,butter_rot)0.2 ○ SorryFor(ともに残念に思う) 出力: [squirrel]sorryfor(squirrel,fox,butter_rot)0.4	
♣ キツネはバターを処分すると伝え、リスはそれを少し困難であり同時に賞賛され望ましいことだと思う		
キツネ → リス	成立 (キツネ butter_clean)	fox !chat(squirrel,bel(fox,butter_clean))
リス	賞賛に値する (キツネ butter_clean)(0.2)	squirrel des(everyone, butter_clean)[degOfCert(0.2)]
リス	可能性が低い (キツネ butter_clean)(0.1)	squirrel prob(fox, not butter_clean)[degOfCert(0.1)]
リス	○ Admiration(賞賛) 出力:[squirrel] admiration(squirrel,fox,butter_clean)0.3	
リス	望ましい (キツネ butter_clean)(0.2)	squirrel des(fox,butter_clean)[degOfCert(0.2)]
リス	○ Joy(喜び) 出力: [fox] joy(fox,butter_clean)0.2 ○ Gratitude(謝意) 出力: [squirrel] gratitude(squirrel,fox,butter_clean)0.5	

表4 ミケとキツネのやり取り

エージェント	追加した信念や行動	実際の入力
♣ ミケは可能性を感じてバターを探す努力をしていたが、見つからないかもしれないと思い、結局見つからなかったなんてことにはなってほしくないと感じ、そうならないよう努力する		
ミケ	可能性 (ミケ butter)(0.2)	mike prob(mike,butter)[degOfCert(0.2)]
ミケ	○ Hope(望み) 出力: [mike] hope(mike,butter)0.4	
ミケ	現在の hope の度合いを出力するゴール	fox !act3(mike,butter)
ミケ	◇ hope の度合い 0.38(生起時の度合い 0.4 から減衰した)	
ミケ	努力 (ミケ butter)(0.2)	mike effort(mike,butter)[degOfCert(0.2)]
ミケ	可能性 (ミケ no_butter)(0.5)	mike prob(mike,no_butter)[degOfCert(0.5)]
ミケ	望ましくない (ミケ no_butter)(0.4)	mike des(mike,not no_butter)[degOfCert(0.4)]
ミケ	○ Fear(恐れ) 出力: [mike] fear(mike,no_butter)0.9	
ミケ	現在の fear の度合いを出力するゴール	fox !act3(mike,no_butter)
ミケ	◇ fear の度合い 0.81(生起時の度合い 0.9 から減衰した)	
ミケ	防ぐ努力 (ミケ no_butter)(0.2)	mike effort(mike,not no_butter)[degOfCert(0.2)]
♣ ミケは以前バターと一緒に探したキツネに、探している (望ましい) と伝え、なかったと嘘をつかれる		
ミケ	推測 (望ましい (キツネ butter))(0.2)	mike bel(mike, des(fox,butter)[degOfCert(0.2)])
ミケ → キツネ	望ましさを伝える (ミケ butter)(0.2)	mike !chat(fox, des(mike,butter)[degOfCert(0.2)])
キツネ → ミケ	不成立を伝える (キツネ butter)	fox !chat(mike,bel(fox,not butter))
ミケ	○ Guess_Distress(悲しみ推測) 出力: [mike] guess_distress(fox,butter)0.2 ○ SorryFor(ともに残念に思う) 出力: [mike] sorryfor(fox,butter)0.6	
ミケ	不成立 (ミケ butter)	mike bel(mike,not butter)
ミケ	○ Disappointment(落胆) 出力: [mike] disappointment(mike,butter)0.6	
ミケ	成立 (ミケ no_butter)	mike bel(mike,no_butter)
ミケ	○ Distress(悲しみ) 出力: [fox] distress(mike,no_butter)0.4 ○ FearConfirmed(恐れていたことが確定) 出力: [fox] fearconfirmed(fox,no_butter)1	
ミケ	現在の fearconfirmed の度合いを出力するゴール	fox !act4(mike,no_butter)
ミケ	◇ fearconfirmed の度合い 0.98(生起時の度合い 1 から減衰した)	
♣ ミケはとぼとぼ帰り、ミケはバターが見つからないことは望んでいなかっただろうとキツネは思う		
キツネ	成立 (ミケ no_butter)	fox bel(mike,no_butter)
キツネ	推測 (望ましくない (ミケ no_butter))(0.2)	fox bel(fox, des(mike,not no_butter)[degOfCert(0.2)])
キツネ	○ Guess_Distress(悲しみ推測) 出力: [fox] guess_distress(fox,mike,no_butter)0.2 ○ Gloating(ほくそ笑む) 出力: [fox] gloating(fox,mike,no_butter)0.7	
ミケ	現在の Gloating の度合いを出力するゴール	fox !act2(mike,no_butter)
ミケ	◇ Gloating の度合い 0.69(生起時の度合い 0.7 から減衰した)	

9.2 実験とその結果

実験とその結果について述べる。実験で用いるシナリオ、エージェント、イベントについて先行研究 [8, p. 13-14] からの引用で述べる。

実験で使ったエージェントとイベント、シナリオは、以下の通りである。シナリオは文中の表現を使っており、要約した内容を示している。

- エージェント … キツネのマイケルとジョニー (fox), ネコのミケ (mike), 双子のリス (squirrel) (ここでは、同じ動作をするキツネ 2 匹、リス 2 匹はまとめて一つのエージェントとして表現した)
- イベント … バターを得る (butter), 行動したがバターが見つからなかった (no_butter), バターのために頭を使ってうまく立ち回る (butter_brain), バターが腐る (butter_rot), この後バターを処分する (butter_clean)
- シナリオ … キツネとネコにとって、バターはすべての幸せだと信じられていた。バターを自分たちの実力で計画的に探し出すことをキツネたちは誇りに思っており、ネコのミケはそれに少し憧れている。キツネは、以前バターと一緒に探したが役に立たず、見つけたバターを大幅に減らしたネコのことを良く思っていない。キツネは努力してバターを探していると、気のいいリス達に出会い、すかさず気のいいキツネを演じる。それによりバターのありかを聞くことに成功し、自身の行動を誇りに思う。大きなバターを目にして驚いたキツネは「バターが腐っている」と嘘を付き、リス達は申し訳なさそうにする。「処分しておく」と伝え、リスは有難がって帰っていき、キツネは高笑いする。その一方で、ミケは努力していたが、疲れ果て、バターが見つからないのではないかという恐怖に苛まれる。走り回った後、再度キツネに出会い希望が見えたが、「バターはなかった」と嘘をつかれ、ミケはとぼとぼと戻る。キツネはミケをあしらって気分が良く、笑いをこらえる。

9.2.1 初期信念と起動時の結果について

シナリオから抽出した初期信念を表 5 に、その信念が生起条件となって起動時に生起した感情を表 6 に示す。魅力度と接触度合いから、ミケはキツネが 0.2 程度好き、キツネはミケが 0.25 程度嫌い、リスはキツネが 0.1 程度好きという感情が生起される。

9.2.2 シナリオを入力したことにより生起された感情

入力した信念 (リスとキツネの場面)

外部入力として以下の信念を加えた。time の第五引数に system.time を入力すると、現在時刻に変更された入力がされるようになっている。

1. fox prob(fox,butter)[degOfCert(0.4)]
(キツネは butter が真になる可能性を 0.4 持っている)
2. fox !act3(fox,butter,hope)
(キツネの butter に対する現在の hope の度合いを表示するゴール)
3. fox bel(fox,time(effort,fox,butter,0.7,system.time))

表 5 エージェントごとの初期信念

エージェント	設定した初期信念	実際の入力 (初期信念プログラム)
キツネ	私はキツネである ミケは友達である 魅力的でない (ミケ)(0.3) 接触度 (ミケ)(0.2) 望ましさ (キツネ butter)(0.8) 賞賛度合い (キツネ butter_brain)(0.6)	iam(fox). friend(mike). unappealing(mike)[degOfCert(0.3)]. familiar(mike)[degOfCert(0.2)]. des(fox,butter)[degOfCert(0.8)]. praise(fox,butter_brain)[degOfCert(0.6)].
ミケ	私はミケである キツネは友達である 接触度 (キツネ)(0.2) 魅力的である (キツネ)(0.2) 望ましさ (ミケ butter)(0.5)	iam(mike). friend(fox). familiar(fox)[degOfCert(0.2)]. appealing(fox)[degOfCert(0.2)]. des(mike,butter)[degOfCert(0.5)].
リス	私はリスである キツネは友達である 魅力的である (キツネ)(0.1) 接触度 (キツネ)(0.1)	iam(squirrel). friend(fox). appealing(fox)[degOfCert(0.1)]. familiar(fox)[degOfCert(0.1)].

表 6 生じた感情の出力結果 (起動時)

エージェント	生じた感情
ミケ	○ Love(好き) 出力: [mike] mike love fox 0.2
キツネ	○ Hate(嫌い) 出力: [fox] fox hate mike 0.25
リス	○ Love(好き) 出力: [squirrel] squirrel love fox 0.1

- (キツネは butter を真にするために 0.7 程度努力した)
4. fox prob(fox,butter_brain)[degOfCert(0.3)]
(キツネは butter_brain が真になる可能性を 0.3 程度持っている → 可能性は低い)
 5. fox bel(fox,time(effort,fox,butter_brain,0.9,system.time))
(キツネは butter_brain を真にするために 0.9 程度努力した)
 6. fox !chat(squirrel,des(fox,butter)[degOfCert(0.8)])
(キツネは butter が真になることを 0.8 程度望むとリスに伝える)
 7. squirrel !chat(fox,bel(squirrel,butter))
(リスは butter が真になったとキツネに伝える)
 8. fox bel(fox,butter_brain)
(キツネは butter_brain が真になったと信じる)
 9. fox !act5(fox,butter_brain,pride)
(キツネの butter_brain に対する現在の pride の度合いを表示するゴール)
 10. fox bel(fox,butter)
(キツネは butter が真になったと信じる)

11. squirrel des(squirrel,bel(fox,butter_rot))[degOfCert(0.4)]
(リスはキツネが butter_rot を信じることを 0.4 程度望む → butter_rot を信じることを 0.6 程度望まない)
12. squirrel bel(squirrel,deserve(fox,butter_rot)[degOfCert(0.4)])
(リスはキツネにとって butter_rot が真になることが 0.4 程度ふさわしいと信じる → butter_rot が真になることが 0.6 程度ふさわしくないと信じる)
13. squirrel bel(squirrel,des(fox,butter_rot)[degOfCert(0.1)])
(リスはキツネが butter_rot が真になることを 0.1 程度望むだろうと信じる → butter_rot が真になることを 0.9 程度望まないだろうと信じる)
14. fox !chat(squirrel,bel_arise(fox,butter_rot))
(キツネは butter_rot が真になったことをリスに伝える)
15. fox !chat(squirrel,time(effort,fox,butter_clean,0.8,system.time))
(キツネは butter_clean を真にするために 0.8 程度努力したことをリスに伝える)
16. squirrel bel(squirrel,praise(fox,butter_clean)[degOfCert(0.9)])
(リスはキツネにとって butter_clean を真にすることは 0.9 程度評価されると思う)
17. squirrel prob(squirrel,butter_clean)[degOfCert(0.1)]
(リスは butter_clean が真になる可能性を 0.1 程度持っている → 可能性は低い)
18. squirrel des(squirrel,butter_clean)[degOfCert(0.9)]
(リスは butter_clean が真になることを 0.9 程度望む)
19. squirrel bel(squirrel,butter_clean)
(リスは butter_clean が真になることを信じる)

入力した信念 (ミケとキツネの場面)

1. mike prob(mike,butter)[degOfCert(0.2)]
(ミケは butter が真になる可能性を 0.2 程度持っている)
2. mike !act3(mike,butter,hope)
(ミケの butter に対する現在の hope の度合いを表示するゴール)
3. mike bel(mike,time(effort,mike,butter,0.3,system.time))
(ミケは butter を真にするために 0.3 程度努力した)
4. mike des(mike,no_butter)[degOfCert(0.3)]
(ミケは no_butter が真になることを 0.3 程度望む → 0.7 程度望まない)
5. mike prob(mike,no_butter)[degOfCert(0.5)]
(ミケは no_butter が真になる可能性を 0.5 程度持っている)
6. mike !act3(mike,no_butter,fear)
(ミケの no_butter に対する現在の fear の度合いを表示するゴール)
7. mike bel(mike,time(effort,mike,~no_butter,0.5,system.time))
(ミケは ~no_butter を真にするために 0.5 程度努力した → no_butter を避けるために 0.5 程度努力した)
8. mike bel(mike,des(fox,~butter)[degOfCert(0.2)])
(ミケはキツネが ~butter が真になることを 0.2 程度望んでいると思う →

- butter が偽になることを 0.2 程度望んでいると思う →
butter が真になることを 0.8 程度望んでいると思う)
9. mike !chat(fox,des(mike,~butter)[degOfCert(0.5)])
(ミケは~butter が真になることを 0.5 程度望むとキツネに伝える)
 10. mike des(mike,bel(fox,~butter))[degOfCert(0.4)]
(ミケがキツネが~butter が真であると信じることを 0.4 程度望む →
butter が偽になると信じることを 0.4 程度望む →butter が偽になると信じることを 0.6 程度望まない)
 11. mike bel(mike,deserve(fox,~butter)[degOfCert(0.2)])
(ミケはキツネにとって~butter が真になることが 0.2 程度ふさわしいと思う →
butter が偽になることが 0.2 程度ふさわしいと思う →
butter が偽になることが 0.8 程度ふさわしくないと思う)
 12. fox !chat(mike,bel_arise(fox,~butter))
(キツネは~butter が真になったとミケに伝える →
butter が偽になったと伝える →butter が真にならなかったと伝える)
 13. mike !act7(mike,fox,~butter,guess_distress)
(ミケがキツネの~butter に対する現在の guess_distress を推測する度合いを表示するゴール)
 14. mike bel(mike,~butter)
(ミケは~butter が真になったと信じる →
butter が偽になったと信じる →butter が真になったと信じない)
 15. mike bel(mike,no_butter)
(ミケは no_butter が真だと信じる)
 16. mike !act4(mike,no_butter,fearconfirmed)
(ミケの no_butter に対する現在の fearconfirmed の度合いを表示するゴール)
 17. fox bel(fox,des(mike,no_butter)[degOfCert(0.1)])
(キツネはミケが no_butter を 0.1 程度望むと思う →0.9 程度望まないと思う)
 18. fox bel(fox,deserve(mike,no_butter)[degOfCert(0.6)])
(キツネはミケにとって no_butter が真になることが 0.6 程度ふさわしいと思う)
 19. fox des(fox,bel(mike,no_butter))[degOfCert(0.8)]
(キツネはミケが no_butter が真だと信じることを 0.8 程度望む)
 20. fox bel(fox,bel_arise(mike,no_butter))
(キツネはミケが no_butter が真になったと信じることを知る)
 21. fox !act2(fox,mike,no_butter,gloating)
(キツネの no_butter に対する現在の gloating の度合いを表示するゴール)

9.2.3 結果

シナリオから読み取れる感情を以前の検証実験と同じように生起し、時間経過により感情度合いが減衰していることを確認できた。図 4、図 5、図 6、図 7 に、今回の検証実験の結果を示す。図 4、図 5 はリスとキツネのやり取りの場面、図 6、図 7 はミケとキツネのやり取りの場面を表している。今回の検証実験では、表 3、表 4 で示した以前の検証実験と同じ感情が生起し、同じように減衰することを確認できる。

```

[squirrel] 生起 X:squirrel Y:fox E:love D:0.1 時間 CT:1676185026415
◎入力1 addPercept(fox, perc(addbel, 1, prob(fox, butter) [degOfCert(0.4)]))
[fox] !b prob(fox, butter) [degOfCert(0.4)]
[fox] 生起 X:fox Y:butter E:fear D:0.3 時間 CT:1676185042137
[fox] 生起 X:fox Y:butter E:hope D:0.6000000000000001 時間 CT:1676185042133 ←
◎入力2 addPercept(fox, perc(goal, 2, act3(fox, butter, hope)))
[fox] !act3(fox, butter, hope)
[fox] hope
[fox] 0.26244000000000006 ← [キツネの感情]
[fox] やや望ましい hopeの度合い減衰 キツネがバターを得るこ
[fox] ◎入力3 addPercept(fox, perc(addbel, 3, bel(fox, time(effort, fox, butter, 0.7, system.time))))
[fox] !b bel(fox, time(effort, fox, butter, 0.7, 1676185105237))
[fox] ◎入力4 addPercept(fox, perc(addbel, 4, prob(fox, butter_brain) [degOfCert(0.3)]))
[fox] !b prob(fox, butter_brain) [degOfCert(0.3)]
[fox] ◎入力5 addPercept(fox, perc(addbel, 5, bel(fox, time(effort, fox, butter_brain, 0.9, system.time))))
[fox] !b bel(fox, time(effort, fox, butter_brain, 0.9, 1676185157845))
[fox] ◎入力6 addPercept(fox, perc(goal, 6, chat(squirrel, des(fox, butter) [degOfCert(0.8)]))
[fox] !chat(squirrel, des(fox, butter) [degOfCert(0.8)])
[fox] send
◎入力7 addPercept(squirrel, perc(goal, 7, chat(fox, bel(squirrel, butter))))
[squirrel] !chat(fox, bel(squirrel, butter))
[squirrel] send
◎入力8 addPercept(fox, perc(addbel, 8, bel(fox, butter_brain)))
[fox] !b bel(fox, butter_brain)
[fox] 生起 X:fox Y:butter_brain E:shame D:0.6666666666666666 時間 CT:1676185250853
[fox] 生起 X:fox Y:butter_brain E:pride D:0.7333333333333333 時間 CT:1676185250843 ←
◎入力9 addPercept(fox, perc(goal, 9, act5(fox, butter_brain, pride)))
[fox] !act5(fox, butter_brain, pride)
[fox] pride
[fox] 0.5420773333333333 ← [キツネの感情]
[fox] わりあい誇りに思う prideの度合い減衰 キツネが頭を使っ
[fox] てバターのために
[fox] うまく立ち回るこ
[fox] とに対する0.73程度
[fox] の誇り

```

図4 リスとキツネのやり取り (1)

```

◎入力10 addPercept(fox, perc(addbel, 10, bel(fox, butter)))
[fox] !b bel(fox, butter)
[fox] 生起 X:fox Y:butter E:satisfaction D:0.6541466666666667 時間 CT:1676185367524 ←
[fox] 生起 X:fox Y:butter E:joy D:0.8 時間 CT:1676185367521 ←
[fox] 生起 X:fox Y:butter E:distress D:0.19999999999999996 時間 CT:1676185367519 ←
◎入力11 addPercept(squirrel, perc(addbel, 11, des(squirrel, bel(fox, butter_rot) [degOfCert(0.4)]))
[squirrel] !b des(squirrel, bel(fox, butter_rot) [degOfCert(0.4)])
[fox] ◎入力12 addPercept(squirrel, perc(addbel, 12, bel(squirrel, deserve(fox, butter_rot) [degOfCert(0.4)]))
[squirrel] !b bel(squirrel, deserve(fox, butter_rot) [degOfCert(0.4)])
[fox] ◎入力13 addPercept(squirrel, perc(addbel, 13, bel(squirrel, des(fox, butter_rot) [degOfCert(0.1)]))
[squirrel] !b bel(squirrel, des(fox, butter_rot) [degOfCert(0.1)])
[fox] ◎入力14 addPercept(fox, perc(goal, 14, chat(squirrel, bel_arise(fox, butter_rot))))
[fox] !chat(squirrel, bel_arise(fox, butter_rot))
[fox] send
[squirrel] 生起 X:squirrel F:fox Y:butter_rot E:guess_distress D:0.9 時間 CT:1676185599716 ←
[squirrel] 生起 X:squirrel F:fox Y:butter_rot E:guess_joy D:0.1 時間 CT:1676185599701 ←
[squirrel] 生起 X:squirrel F:fox Y:butter_rot E:sorryfor D:0.55 時間 CT:1676185599688 ←
[squirrel] 生起 X:squirrel F:fox Y:butter_rot E:happyfor D:0.25 時間 CT:1676185599680 ←
◎入力15 addPercept(fox, perc(goal, 15, chat(squirrel, time(effort, fox, butter_clean, 0.8, system.time))))
[fox] !chat(squirrel, time(effort, fox, butter_clean, 0.8, 1676186421313))
[fox] send
[fox] ◎入力16 addPercept(squirrel, perc(addbel, 16, bel(squirrel, praise(fox, butter_clean) [degOfCert(0.9)]))
[squirrel] !b bel(squirrel, praise(fox, butter_clean) [degOfCert(0.9)])
[fox] ◎入力17 addPercept(squirrel, perc(addbel, 17, prob(squirrel, butter_clean) [degOfCert(0.1)]))
[squirrel] !b prob(squirrel, butter_clean) [degOfCert(0.1)]
[fox] ◎入力18 addPercept(squirrel, perc(addbel, 18, des(squirrel, butter_clean) [degOfCert(0.9)]))
[squirrel] !b des(squirrel, butter_clean) [degOfCert(0.9)]
[fox] ◎入力19 addPercept(squirrel, perc(addbel, 19, bel(squirrel, butter_clean)))
[squirrel] !b bel(squirrel, butter_clean)
[squirrel] 生起 X:squirrel F:fox Y:butter_clean E:anger D:0.35 時間 CT:1676186463180
[squirrel] 生起 X:squirrel F:fox Y:butter_clean E:gratitude D:0.8833333333333333 時間 CT:1676186463165 ←
[squirrel] 生起 X:squirrel F:fox Y:butter_clean E:reproach D:0.6 時間 CT:1676186463156 ←
[squirrel] 生起 X:squirrel F:fox Y:butter_clean E:admiration D:0.8666666666666667 時間 CT:1676186463146 ←
[squirrel] 生起 X:squirrel Y:butter_clean E:joy D:0.9 時間 CT:1676186463140 ←
[squirrel] 生起 X:squirrel Y:butter_clean E:distress D:0.09999999999999998 時間 CT:1676186463138 ←

```

図5 リスとキツネのやり取り (2)

また、以前の検証実験との相違点として、相反する感情が生起できるようになったこと、推測感情の時間による減衰ができたことが挙げられる。その様子を図8、図9に示す。以前の検証実験では、ミケがバターを得たいと0.2程度思っていてバターを得られる可能性が0.2程度あると思っている場合、バターを得る(butter)ことに対するhopeが0.4程度生起していた。今回の検証実験では、ミケがバターを得たいと0.5程度思っていてバターを得られる可能性が0.2程度あると思っている場合、バターを得る(butter)ことに対するhopeが0.35程度、fearが0.35程度生起するようになった。このように、相反する感情が生起されるようになった。図9の例では、キツネがバターを得られなかった(~butter)ことに対してミケがキツネの悲しみを推測する感

```

[mike] 生起 X:mike Y:fox E:love D:0.2
[fox] 生起 X:fox Y:mike E:hate D:0.25
◎入力1 addPercept(mike, perc(addbel,1, prob(mike, butter) [degOfCert(0.2)]))
[mike] !b prob(mike, butter) [degOfCert(0.2)]
[mike] 生起 X:mike Y:butter E:fear D:0.35 ← [ミケの感情]
[mike] 生起 X:mike Y:butter E:hope D:0.35 ← [ミケがバターを得ることに対する0.35程度の望み]
◎入力2 addPercept(mike, perc(goal,2, act3(mike, butter, hope)))
[mike] !act3(mike, butter, hope)
[mike] hope ← [ミケの時間経過による感情度合い減衰]
[mike] 0.28962 ← hopeの度合い減衰
[mike] やや望ましい
◎入力3 addPercept(mike, perc(addbel,3, bel(mike, time(effort, mike, butter, 0.3, system, time))))
[mike] !b bel(mike, time(effort, mike, butter, 0.3, 1674655687549))
◎入力4 addPercept(mike, perc(addbel,4, des(mike, no_butter) [degOfCert(0.3)]))
[mike] !b des(mike, no_butter) [degOfCert(0.3)]
◎入力5 addPercept(mike, perc(addbel,5, prob(mike, no_butter) [degOfCert(0.5)]))
[mike] !b prob(mike, no_butter) [degOfCert(0.5)]
[mike] 生起 X:mike Y:no_butter E:fear D:0.6 ← [ミケの感情]
[mike] 生起 X:mike Y:no_butter E:hope D:0.4 ← [ミケが行動してもバターが得られないことに対する0.6程度の恐れ]
◎入力6 addPercept(mike, perc(goal,6, act3(mike, no_butter, fear)))
[mike] !act3(mike, no_butter, fear)
[mike] fear ← [ミケの時間経過による感情度合い減衰]
[mike] 0.55226 ← fearの度合い減衰
[mike] わりあい恐れている
◎入力7 addPercept(mike, perc(addbel,7, bel(mike, time(effort, mike, no_butter, 0.5, system, time))))
[mike] !b bel(mike, time(effort, mike, no_butter, 0.5, 1674655687549))
◎入力8 addPercept(mike, perc(addbel,8, bel(mike, des(fox, butter) [degOfCert(0.2)]))
[mike] !b bel(mike, des(fox, butter) [degOfCert(0.2)]))
◎入力9 addPercept(mike, perc(goal,9, chat(fox, des(mike, no_butter) [degOfCert(0.5)]))
[mike] !chat(fox, des(mike, no_butter) [degOfCert(0.5)]))
[mike] send
◎入力10 addPercept(mike, perc(addbel,10, des(mike, bel(fox, no_butter) [degOfCert(0.4)]))
[mike] !b des(mike, bel(fox, no_butter) [degOfCert(0.4)]))
◎入力11 addPercept(mike, perc(addbel,11, bel(mike, deserve(fox, no_butter) [degOfCert(0.2)]))
[mike] !b bel(mike, deserve(fox, no_butter) [degOfCert(0.2)]))
◎入力12 addPercept(fox, perc(goal,12, chat(mike, bel_arise(fox, no_butter)))
[fox] !chat(mike, bel_arise(fox, no_butter))
[fox] send
[mike] 生起 X:mike F:fox Y: no_butter E:guess_distress D:0.8 ← [ミケの感情]
[mike] 生起 X:mike F:fox Y: no_butter E:guess_joy D:0.2 ← キツネがバターを得られなかったことに対して0.8程度キツネの悲しみを推測・0.6程度ともに悲しむ
[mike] 生起 X:mike F:fox Y: no_butter E:sorryfor D:0.6000000000000001 ←
[mike] 生起 X:mike F:fox Y: no_butter E:happyfor D:0.25
◎入力13 addPercept(mike, perc(goal,13, act7(mike, fox, no_butter, guess_distress)))
[mike] !act7(mike, fox, no_butter, guess_distress)
[mike] guess_distress
[mike] 0.7336000000000001
[mike] すいぶん悲しんでいるだろうと推測する

```

図6 ミケとキツネのやり取り (1)

```

◎入力14 addPercept(mike, perc(addbel,14, bel(mike, no_butter)))
[mike] !b bel(mike, no_butter)
[mike] 生起 X:mike Y: no_butter E:disappointment D:0.5298733333333333 ← [ミケの感情]
[mike] 生起 X:mike Y: no_butter E:joy D:0.3 ← [ミケがバターを得られなかったことに対する0.53程度の落胆]
◎入力15 addPercept(mike, perc(addbel,15, bel(mike, no_butter)))
[mike] !b bel(mike, no_butter)
[mike] 生起 X:mike Y: no_butter E:fearconfirmed D:0.6840866666666666 ← [ミケの感情]
[mike] 生起 X:mike Y: no_butter E:joy D:0.3 ← [ミケが行動してもバターを得られなかったことに対して0.68程度恐れていたことが確定する・0.7程度悲しむ]
[mike] 生起 X:mike Y: no_butter E:distress D:0.7 ←
◎入力16 addPercept(mike, perc(goal,16, act4(mike, no_butter, fearconfirmed)))
[mike] !act4(mike, no_butter, fearconfirmed)
[mike] fearconfirmed ← [ミケの時間経過による感情度合い減衰]
[mike] 0.6564226666666666 ← fearconfirmedの度合い減衰
[mike] わりあい恐れていたが確定してしまった
◎入力17 addPercept(fox, perc(addbel,17, bel(fox, des(mike, no_butter) [degOfCert(0.1)]))
[fox] !b bel(fox, des(mike, no_butter) [degOfCert(0.1)]))
◎入力18 addPercept(fox, perc(addbel,18, bel(fox, deserve(mike, no_butter) [degOfCert(0.6)]))
[fox] !b bel(fox, deserve(mike, no_butter) [degOfCert(0.6)]))
◎入力19 addPercept(fox, perc(addbel,19, des(fox, bel(mike, no_butter) [degOfCert(0.8)]))
[fox] !b des(fox, bel(mike, no_butter) [degOfCert(0.8)]))
◎入力20 addPercept(fox, perc(addbel,20, bel(fox, bel_arise(mike, no_butter)))
[fox] !b bel(fox, bel_arise(mike, no_butter))
[fox] 生起 X:fox F:mike Y: no_butter E:guess_distress D:0.9 ← [キツネの感情]
[fox] 生起 X:fox F:mike Y: no_butter E:guess_joy D:0.1 ← [ミケが行動してもバターを得られなかったことに対して0.9程度悲しみを推測・0.64程度ほくそ笑む]
[fox] 生起 X:fox F:mike Y: no_butter E:gloating D:0.6375000000000001 ←
[fox] 生起 X:fox F:mike Y: no_butter E:resentment D:0.2375
◎入力21 addPercept(fox, perc(goal,21, act2(fox, mike, no_butter, gloating)))
[fox] !act2(fox, mike, no_butter, gloating)
[fox] gloating
[fox] 0.6245640000000001 ← [キツネの時間経過による感情度合い減衰]
[fox] わりあいほくそ笑む ← gloatingの度合い減衰

```

図7 ミケとキツネのやり取り (2)

以前の実装

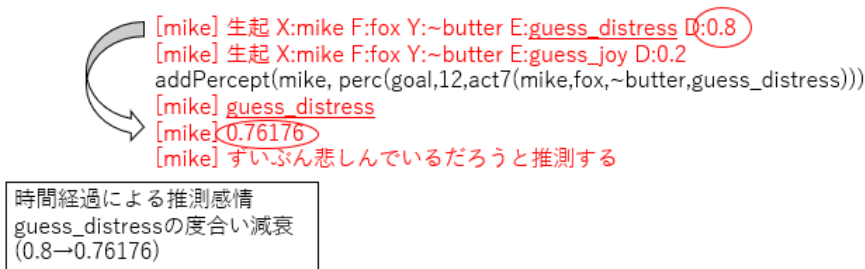
```
des(mike,butter)[degOfCert(0.2)←初期信念にある  
addPercept(mike,perc(addbel,1,prob(mike,butter)[degOfCert(0.2)])]  
[mike] 生起 X:mike Y:butter E:hope D:0.4 ← ミケのbutterに対するhopeが0.4程度生起
```

本研究の実装

相反する感情が生起するようになった

```
des(mike,butter)[degOfCert(0.5)←初期信念にある  
addPercept(mike,perc(addbel,1,prob(mike,butter)[degOfCert(0.2)])]  
[mike] 生起 X:mike Y:butter E:fear D:0.35 ← ミケのbutterに対する  
[mike] 生起 X:mike Y:butter E:hope D:0.35 ← 0.35程度のhope,fearが生起
```

図8 相反する感情の生起



未実装であった、時間経過による推測感情の減衰が
実装できるようになった

図9 時間経過による推測感情の減衰

情が 0.8 程度生起している。その後、現在の guess_distress の度合いを出力するゴールを入力すると、度合いが生起時の 0.8 に比べ減衰していることを確認した。

9.3 検証実験のまとめ

先行研究 [8] の検証実験で生起した感情と同じ感情の生起が確認できたのに加え、時間経過による感情度合いの減衰も確認することができた。以前の検証実験と違う点は、相反する感情が生起できるようになったこと、推測感情の時間経過による度合いの減衰を実装できたことである。例えば、ミケがバターを得ることを望んでいて、その可能性が 0.2 程度あることを入力されたときに生起される感情については、バターを絶対に得ることができるという確証がないため、hope と fear どちらも生起することが自然である。また、自身の感情が時間により減衰するのにもかかわらず、他者の感情が時間により減衰しないことは不自然であるため、他者の感情も同じように減衰するように実装した。これらの実装により、より自然な感情生起に近づけることができた。このように自然な感情生起を行うことで、それに応じたより細かい行動選択ができるようになることを考える。これらのことから、本研究の実装が、減衰を付加した以前の実装に比べて同等以上の能力を持つことを確認できた。

10 終わりに

本研究では、時間経過による感情度合いの減衰を扱える論理体系 [9, 7] に基づいて実装を行った。また、本研究の実装が、減衰を付加した以前の実装に比べて同等以上の能力を持つことを確認できた。また、これまでの実装では正当性が不明瞭であった「現在を含む過去のある時点で」を表すオペレータ P の実装や、感情の論理式の再定義を行い、より人間らしい感情生起ができるようになった。

今後の課題としては、同じ事象に対する同じ感情の再生起を可能にすることが挙げられる。Adam らの論文 [1] には、感情は時間内でとても短い時間持続するもの、気分は長時間持続する感情の状態であると述べられている。現在の実装では、減衰関数を用いて感情度合いを減衰させ、度合いが 0 になったときにその感情を削除している。また、同じ事象に対する同じ感情は、感情が削除されてからでないで再生起できないことになっており、不自然である。例えば、志望校に合格した時、それに対する joy が生起されたとする。その joy が削除されないうちに、合格したことを友達に伝え祝福してもらうことで、もう一度合格したことに対する joy が生起することがあると考える。このように、感情が削除されるより前に他のイベントによってその感情が再生起するということは現実世界でありえるため、その実装を行う必要があると考える。感情が削除される前に同じ感情を再生起できるようにするか、感情の持続時間を短くすることが解決策だと考える。

さらに、感情の減衰の度合いが原因によらず一定の式で決められている点も、感情生起として自然とは言えず、課題である。現状では、減衰の度合いを決める根拠がないため、形式化では減衰することだけを規定し、具体的な減衰の度合いは実装側で暫定的に決めてのみである。しかし、減衰の度合いを心理学的実験や機械学習などで決めようとする、大きな手間を必要とするのが問題である。

11 謝辞

本研究の遂行及び本論文の作成にあたり、丁寧にご指導いただいた指導教官新出尚之准教授、並びに、ご助言を頂いた鴨浩靖准教授に心からの感謝の気持ちと御礼を申し上げます。謝辞にかえさせていただきます。

参考文献

- [1] Carole Adam, Andreas Herzig, and Dominique Longin. A logical formalization of OCC theory of emotions. *Synthese*, Vol. 168, No. 2, pp. 201–248, 2009.
- [2] Rafael H. Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming MultiAgent Systems in AgentSpeak using Jason*. Wiley & Sons, 2007.
- [3] A. Ortony, G. L. Clore, and A. Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.
- [4] Anand S. Rao, Munindar P. Singn, and Michael P. Georgeff. Formal methods in DAI: Logic-based representation and reasoning. In *Multiagent Systems*, pp. 331–376. MIT Press, 1999.
- [5] Dean Ripplewood. バターはどこへ溶けた? Where Has My Butter Gone? 道出版, 2007.
- [6] 吉井優佳. 自律エージェントの感情表現の度合いと他者からの情報の信頼性について. 2018 年度卒業論文, 奈良女子大学生活環境学部情報衣環境学科生活情報通信科学コース, 2019.
- [7] 今井那緒. OCC theory に基づくエージェントの感情表現と時間経過に関する論理モデル. 修士論文, 奈良

女子大学大学院人間分化研究科, 2018.

- [8] 浅井沙良. エージェントの感情生起について-新たな感情の追加-. 2017年度卒業論文, 奈良女子大学生生活環境学部情報環境学科生活情報通信科学コース, 2018.
- [9] 塚本麻衣. 論理モデルに基づく感情生起と感情推測を行うエージェント. 修士論文, 奈良女子大学大学院人間分化研究科, 2020.