情報処理学会研究報告
IPSJ SIG Technical Report

動的議論システムの意味論的考察

南部　優[†1]　高橋和子[†1]

本発表では，動的議論システムの意味論について考察する．これまでの議論システムは，議論への参加エージェントの知識ベースは不変で accept される論証や議論の勝敗は静的に計算できるような仕組みになっていた．それに対して，著者らはエージェントが個別の知識ベースを持ち，議論の実行によって提示された論証によってこの知識ベースが変化していくような動的議論システムを提案した．そこでは実行ごとに accept される論証を考慮する必要があり，これまでの意味論を適用するのは不適切である．本発表では，この動的議論システムに与える新しい意味論を提案する．

A Study on Semantics for Dynamic Argumentation Systems

Yu Nambu[†1] and Kazuko Takahashi[†1]

This presentation discusses semantics for dynamic argumentation systems. Most argumenattion systems proposed so far are constructed so that agents' knowledge bases are invariant, a set of acceptable arguments and the winner of the argumentation can be derived from them. On the other hand, we proposed a dynamic argumentation system in which each agent has its own knowledge base that chages by the disclosed arguments according as the argumentation proceeds. In this sytem, we have to consider acceptable arguments for each execution of an argumentation. We propose new semantics for the dynamic argumentation system.

1. はじめに

対話についてはこれまで多くの分野で様々な観点からの研究が行われている．議論 (argumentation) は対話の一種だが，知識ベースから主張と根拠をもとにした論証 (argument) を導出し，さらに論証同士の間に反論の関係がある，という意味で形式論理にうまく適合するものである．Dung が議論と論理プログラミングや非単調推論との関係を定式化して以来[7]，議論は人工知能関係の多くの研究者の興味を引き出すことになった[3),11)]．また，議論システムは，マルチエージェント環境での意思決定支援[1)] や裁判における主張の証明，品質保証 (assurance) の支援など実際面への応用も期待されている．

議論システムは一般に各エージェントが発言可能な論証とそれらの間の攻撃関係からなる議論フレームワークで定義され，これらは議論の進行にかかわらず不変なものとして設定されてきた．しかし，実際の議論はエージェント間の情報のやりとりであり，動的環境すなわちエージェントの知識が変化していくことを想定する方が自然である．著者らは，議論の動的な側面に着目し，動的議論システムを提案した[9),10)]．動的議論システムにおいては「議論の実行」という概念が伴う．そこでは，各エージェントが固有の知識をもち，議論の進行にともない相手の論証が提示されることによって自分に新しい知識が加わり，その結果新しい反論が生まれるという現象がある．我々は，このような現象を「脅威」と呼び，動的議論システムに特有のものとみなしている．動的議論システムでは脅威を扱うことによって，自分の不用意な発言が自分の首をしめるような現実に頻繁に起こる現象を説明することができた．

議論システムの意味は一般に各エージェントが発言可能な論証およびそれらの間の攻撃関係から成る議論フレームワークから計算される extension として与えられる[7]．extension は与えられた議論フレームワークにおいて同時に accept されることを許すような論証の集合であり，一般に矛盾のない集合である．Dung は acceptance の強さによっていくつかの extension を定義した．これを拡張したものとして，prudent semantics[6]．ideal semantics[8], semi-stable semantics[5] などいくつかの意味論が提案され，Boroni らはこれらの強さを懐疑性 (skepticism) という観点から比較している[2]．

一方，動的議論では実行順序によって提示される論証の集合が異なるため，accept されるものが異なり，議論の勝敗も異なる．さらに，議論の進行とともに新たな論証が出現し，結果としてそれらの攻撃関係も新たに加わるため，これまでの意味論をそのまま適用するのは不適切である．Cayrol らは論証や攻撃関係が増えた場合の extension の変化について考察しているが[4]，増える理由やタイミングについては考慮していない．

本発表では，動的議論システムの意味論として，実行順序ごとに extension を与え，これらの集合を議論フレームワークに対する extension と定義した動的 extension を提案する．

†1 関西学院大学大学院理工学研究科
School of Science & Technology, Kwansei Gakuin University

与えられた議論フレームワークから論証をノード，攻撃関係をエッジとする初期議論木を作成する．初期議論木の枝を任意の順番に選んで実行し，実行によって得られる実行木をもとにその実行に対する extension を計算する．本発表ではその計算方法について述べ，初期議論木の形と得られた extension の関係や動的 extension のもつ意味について考察する．

また，議論の実行手続きとして，既に提案している APKC2 以外の方法について述べ，現実の議論の模倣と，実行結果得られる extension の観点からそれらを比較する．

本発表の構成は以下のとおりである． 2 節で基本概念を説明し動的議論システムについて述べる． 3 節で動的議論システムの意味を与える． 4 節で議論の実行手続きについて検討する．最後に，5 節で結論を述べる．

2. 動 的 議 論

2.1 動的議論フレームワーク

動的議論システムは，提案者 (P) と反対者 (C) が各々個別の共通する要素を含む知識ベースを持つ．与えられた各々の知識ベースを基に P,C の論証の集合と論証間の攻撃関係の集合が導出される．本発表では論証とその間の攻撃関係は与えられたものとして，動的議論を抽象的なレベルで取り扱うものとする．

定義 1 (議論フレームワーク) 議論フレームワーク $AF = \langle Arg_P, Arg_C, Atts \rangle$ は次のように定義される．Arg_P, Arg_C はそれぞれ P と C の持つ論証の集合である．$Atts$ は $Arg_P \cup Arg_C$ の上で定義される二項関係で攻撃関係と呼ばれる．$(A, B) \in Atts$ に対して，$A \in Arg_P, B \in Arg_C$ または $A \in Arg_C, B \in Arg_P$ が成り立つ．また，任意の論証 A, B の組に対して，$(A, B), (B, A)$ の両方が同時に $Atts$ に含まれることはない．

定義 2 (議論木) φ を提案，P と C は φ の提案者と反対者，AF を議論フレームワークとする．この時，以下の条件を満たす有限の有向木を AF の議論木とよぶ．

- 根ノードは φ に対する P の論証に相当する．
- 各ノードは $Arg_P \cup Arg_C$ に含まれる論証に相当する．
- 各ノード N からノード M へのエッジは論証 N から論証 M への攻撃に相当する．
- P と C の論証は各枝[⋆1]において交互に出現する．
- 各ノードは単一の親ノードを持つ．
- 異なる枝に同じ論証が出現する場合がある．
- 各枝においてループは存在しない．

定義 3 (枝の勝敗) もし枝 D の葉ノードが P の論証であるならば，D において P は勝利する．そうでなければ，D において P は敗北する．

定義 4 (候補部分木) 候補部分木は本来の議論木から C のノードについては子ノードを一つだけ，P のノードについては子ノードをすべて含むような部分木である．

定義 5 (解部分木) 解部分木は議論木のすべての枝において P が勝利しているような候補部分木である．

多くの議論システムでは，議論の勝敗は各枝を個別に処理して決定される．しかし，動的議論システムでは，一つの枝のすべての論証が処理された後，他の枝が処理される．この場合，ある枝の論証が別の枝へと影響を及ぼす．これによって新たな論証が生成され，議論の勝敗を覆すことがある．これが動的議論システムのもっとも大きな特徴である．

2.2 議論の実行

議論の実行は，枝の選択，コミットメントストアとエージェントの議論履歴の更新，議論木の修正の過程から成る．コミットメントストアは，今までに提示されたすべての論証に含まれるすべての論理式の集合である．

与えられた議論フレームワークの議論木を構成する．以後，これを初期議論木とよぶ．議論は初期議論木の枝を選択することで開始される．実行は枝にそって進み，葉ノードにたどり着いたとき，停止する．その時コミットメントストアは更新され，エージェントは自らの知識ベースに加えてコミットメントストアを用いて新たな論証を作り出せる．そのため，論証の集合と攻撃関係の集合の要素数は各枝の実行に従って増加する．新しい論証が生成されたならば，議論木に新しいノードが追加される．そして次に別の枝が選ばれる．実行が進むにつれて，実行されたノードはマークされ，マークされていないノードを含む枝が選ばれ実行される．停止した枝はマークされていない新しいノードが加えられると再開可能となる．枝の選択において，発言者の順番は保持される．これは，ある枝が (P か C どちらかの) 発言者で停止したときに，次に選ばれる枝はもう一方の発言者のノードから始まるべきであることを意味している．

議論の実行の形式的な定義を述べる．

定義 6 (実行可能ノード) 現在が P/C の順番で，ある枝 D=$[M_1,\ldots,M_n]$ における枝 M_i $(1 \leq i \leq n)$ が存在する時，もし $M_1,\ldots,M_{i-1}$ がマーク済みで，$M_i,\ldots,M_n$ がマークされていない，かつ M_i が P/C の論証ならば，ノード M_i は実行可能であるという．

定義 7 (停止・再開) 枝 D のすべてのノードが実行されたならば，D の実行は停止され

⋆1 枝とは，根ノードからある葉ノードまでの一本のパスである．

たという．停止された枝 D について，もし議論木の修正によって実行可能なノードが追加され，D が選択されたならば，D の実行は再開されたという．

定義 8 (脅威) $\langle Arg_P, Arg_C, Atts\rangle$ に対して A,A' をともに Arg_P または Arg_C に属する論証とする．もし A の実行によって A' に対して攻撃する論証が一つ以上生成されるならば，A は A' に対する脅威であるといい， $threat(A,A')$ と表記する．また，A,A' をそれぞれ脅威元，脅威先という．

直観的に，脅威は自分の論証と矛盾した論証を実行したために議論の相手に有利な情報を与えるような現象である．同一枝内の論証が脅威になる場合もある．

知識ベースの変更を伴う議論の手続き ($APKC2$)

$AF = \langle Arg_P, Arg_C, Atts\rangle$ を議論フレームワーク，φ を提案とする．

[ステップ 1 (初期化)]

$turn = P$ とする．すべてマークされていない AF の φ についての初期議論木を構築する．

[ステップ 2 (議論の実行)]

if 実行できる枝がない
  if $turn$=P, **then** P の敗北で終了．
  else $turn$=C, **then** P の勝利で終了．
else 枝を選び，実行可能なノードから葉ノードまで実行する．

[ステップ 3 (木の修正)]

$threat(A, A')$ を満たす A, A' が Arg_P,Arg_C に含まれているとき
if A がマークされている
  then Arg_P または Arg_C に新しい論証 B を加え，それぞれに，新しい攻撃関係 (B, A') を $Atts$ に加える．
φ における AF の更新によって議論木を再構築.
if ノード N とノード M が同一であり， N がマークされかつ，M がマークされていない
  then M をマークする．
ステップ 2 へ．

APKC2 ではお互いのエージェントが交互に論証を提示し，反論できないエージェントがその実行における敗者となる．論証を提示する順番を交互にするという条件は，現実の議論の進行を模倣するものである．

定義 9 (実行の勝敗) APKC2 のある実行が P の勝利/敗北で停止したとき，P はその実行において勝利/敗北するという．

定義 10 (実行木) 議論フレームワーク AF において，APKC2 のある実行 $exec$ の結果として最終的に得られる木の部分木で，マークされたノードとその間のエッジから構成される木を $exec$ に対する実行木という．

定義 11 (連続候補部分木) 候補部分木 CT において，もし新しい論証の追加によって一つ以上の候補部分木が生成されたとき，それらを CT の連続候補部分木という．

定義 12 (動的解部分木) 初期議論木において候補部分木を CT とする．CT のどのような枝の実行順においても，もし APKC2 が P の勝利で停止もしくは，P で勝利するような CT の連続候補部分木が存在するとき，CT は動的解部分木であるという．

定義 13 (議論における動的勝利) もし議論木が動的解部分木をもつならば，そのとき P は動的勝利であるという．そうでなければ P は動的敗北であるという．

T_{init} と T_{final} をそれぞれ AF の φ についての初期議論木と APKC2 を実行した結果得られる最終的な議論木とする．もし Arg_P と Arg_C に脅威が存在しない場合，マーク/未マークを無視すると，どのような実行 e に対しても T_e は T_{init} に含まれ $T_{final} = T_{init}$ となる．

3. 意　味　論

3.1 動的 extension

Dung の extension の定義に準じて，動的 extension を定義する[7]．

定義 14 (conflict-free，admissible) 議論フレームワーク $AF = \langle Arg_P, Arg_C, Atts\rangle$ に対して， $A \in Arg_P \cup Arg_C$, $\mathcal{S} \subseteq Arg_P \cup Arg_C$ とする．
(1) $(A, B) \in Atts$ となるような A，B が $\mathcal{S}$ の要素として含まれないとき，$\mathcal{S}$ は $conflict\text{-}free$ である．
(2) $\mathcal{S}$ が A を攻撃するすべて論証を攻撃するとき，$\mathcal{S}$ は A を防御する．
(3) $\mathcal{S}$ が $conflict\text{-}free$ かつ $\mathcal{S}$ のすべての要素に対して防御するとき， $\mathcal{S}$ は $admissible$ である．

定義 15 (preferred extension) $\mathcal{E} \subseteq Arg_P \cup Arg_C$ とする．$\mathcal{E}$ が $\subseteq$ に関して極大な

admissible 集合 のとき，$\mathcal{E}$ は *preferred extension* という．

議論フレームワーク AF において，T_e を実行 e に対する実行木とする．Arg'_P と Arg'_C をそれぞれ T_e に出現する P と C の論証の集合とし，$Atts'$ をそれらの論証間の攻撃関係の集合とする．すると，T_e は議論フレームワーク $AF_e = \langle Arg'_P, Arg'_C, Atts' \rangle$ の φ についての議論木に相当する．このような AF_e を e に対する議論フレームーワークとよぶ．

定義 16 (動的 extension) 議論フレームワーク AF とその実行 $exec$ について，AF_{exec} を $exec$ の議論フレームワークとする．このとき AF_{exec} の *preferred extension* を AF の $exec$ に対する動的 extension という．AF のすべての $exec$ に対する動的 extension の集合を AF に対する動的 extension という．

3.2 脅威の及ぼす影響

AF を議論フレームワークとする．

脅威が議論の勝敗に影響を与えることは明らかだが，どのような場合にどのような影響を与えるかは明解でない．以下では，AF の初期議論木の形ごとに脅威の存在しない場合と存在する場合の動的 extension を比較しながら，脅威の及ぼす影響について述べる．

なお，ここでは説明の簡単化のため，脅威元と脅威先が異なる候補部分木に属さないとする．

AF が P から P への脅威を含む場合について説明する．

Pr，Pd をそれぞれ脅威元，脅威先とする．また C' を脅威によって加えられる新しいノードとする．簡単化のために議論木内に存在する枝を D_1,D_2 のみとし，D_1 が P_r，D_2 が P_d をそれぞれ含むと仮定する．脅威が存在しないとは，$threat(P_r, P_d)$ ではなく，どのような実行をしても C' は出現しないことを意味する．

(Case 1) すべての葉ノードが P の場合

ここでは

(i) C' が葉ノード

(ii) C' が新しい枝

の二つの場合が考えられる．

まず図 1(1)(i) の脅威が存在しない場合を考える．左の枝 D_1 からの実行を $exec_1$，右の枝 D_2 からの実行を $exec_2$ とする．

$exec_1$ の場合，まず根ノード P_0 から始まり，C_1，P_r をマークする．D_1 には実行可能なノードがこれ以上存在しないので，次に D_2 へと移動し C_2，P_d をマークする．ここで D_2 にも他の枝にも実行可能なノードが存在しないので実行が終了する．このとき $exec_1$ に対する動的 extension $\mathcal{E}_{exec_1}$ は $\mathcal{E}_{exec_1} = \{P_0, P_r, P_d\}$ となる．

$exec_2$ の場合，$exec_1$ とは逆の順番に，D_2 の実行可能なノード P_0, C_2, P_d をマークしたのち， D_1 の実行可能なノード C_1, P_r をマークし実行を終了する．このとき動的 extension $\mathcal{E}_{exec_2} = \{P_0, P_r, P_d\}$ となり，$\mathcal{E}_{exec_1}$ と一致する．

よって，この AF に対する動的 extension $\mathcal{E}$ は $\mathcal{E} = \{\mathcal{E}_{exec_1}\}$ である．

どちらの枝から実行しても P のノードを実行したのちに終了しているので，この議論木は動的解部分木を持ち，P は動的勝利である．

次に図 1(1)(i) の脅威が存在する場合を考える．左の枝 D_1 からの実行を $exec'_1$，右の枝 D_2 からの実行を $exec'_2$ とする．

$exec'_1$ の場合，根ノード P_0 から始まり C_1,P_r をマークする．D_1 には実行可能なノードがこれ以上存在しないので，次に木の修正を行う．C_1, C_2 はそれぞれ脅威元，脅威先であるので，C' が D_2 の葉ノードとして生成される．次に D_2 に移動し実行可能な C_2, P_d, C' をマークし，実行可能なノードがなくなるので終了する．このとき $turn = P$ なので $exec'_1$ では P の敗北となる．このときの $exec'_1$ に対する動的 extension は $\mathcal{E}_{exec'_1} = \{C', C_2, P_r\}$ である．

$exec'_2$ の場合，D_2 で実行可能な P_0, C_2, P_d を順番にマークした後，木の修正を行うが，この時点での変更はない．その後，D_1 へと移動し C_1, P_r をマークした後，木の修正を行う．ここで脅威によって新しく実行可能なノード C' が追加される．次に，D_2 を再開し C' をマークした後，実行を終了する．このときも P の敗北となる．$exec'_2$ に対する動的 extension は $exec'_1$ の動的 extension と一致する．

よって，この AF に対する動的 extension $\mathcal{E}'$ は $\mathcal{E}' = \{\mathcal{E}_{exec'_1}\}$ となる．

$exec_1, exec_2$ ともに P の敗北となるので，この議論木は動的解部分木を持たず，P は動的敗北となる．

以上により (i) においては，脅威が存在する場合としない場合では生成される動的 extension が異なり勝敗にも影響を与えている．

図 1(1)(ii) の場合も (i) と同様に，どちらの枝から実行しても動的 extension が一致するので脅威が存在しない場合の AF に対する動的 extension は $\mathcal{E} = \{\{P_0, P_r, P_d, P_1\}\}$ となり，この場合も P が動的に勝利である．脅威が存在する場合の AF に対する動的 extension は $\mathcal{E}' = \{\{P_r, P_1, C_2, C'\}\}$ となり，P は動的敗北である．

(Case 2) すべての葉ノードが C の場合

図 1(2) の脅威が存在しない場合を考える．左の枝 D_1 からの実行を $exec_1$，右の枝 D_2 か

らの実行を $exec_2$ とする．

$exec_1$ の場合，P_0, C_1, P_r, C_3 を順番にマークする．ここで D_1 の葉ノードが C_3 で終わっている．発言は順番に従うので次に P の実行可能なノードをマークしなければならないが，議論木中に存在しないので実行はここで終了する．

$exec_1$ に対する動的 extension $\mathcal{E}_{exec_1}$ は $\mathcal{E}_{exec_1} = \{C_1, C_3\}$ である．

$exec_2$ の場合，D_2 に存在する実行可能なノード P_0, C_2, P_d, C_4 を順番にマークする．$exec_1$ の場合と同様に，P の実行可能なノードが存在しないので実行が終了する．

$exec_2$ に対する動的 extension $\mathcal{E}_{exec_2}$ は $\mathcal{E}_{exec_2} = \{C_2, C_4\}$ である．

よって，この AF に対する動的 extension $\mathcal{E}$ は $\mathcal{E} = \{\mathcal{E}_{exec_1}, \mathcal{E}_{exec_2}\}$ となる．

次に図 1(2) の脅威が存在する場合を考える．左の枝 D_1 からの実行を $exec'_1$，右の枝 D_2 からの実行を $exec'_2$ とする．$exec'_1$ の場合，D_1 に存在する実行可能なノード P_0, C_1, P_r, C_3 を順番にマークする．ここで実行可能なノードが存在しないので実行がここで終了する．

このとき，脅威の影響によって C' が加わるものの C' が実行されることはないため，$exec'_1$ に対する動的 extension $\mathcal{E}_{exec'_1}$ も脅威がない場合と変わらず $\mathcal{E}_{exec'_1} = \{C_1, C_3\}$ である．

$exec'_2$ の場合，D_2 に存在する実行可能なノード P_0, C_2, P_d, C_4 を順番にマークする．$exec'_1$ の場合と同様に，P の実行可能なノードが存在しないので実行が終了する．

こちらも脅威がない場合と同じで $exec'_2$ に対する動的 extension $\mathcal{E}_{exec'_2}$ は $\mathcal{E}_{exec'_2} = \{C_2, C_4\}$ である．

よって，この AF に対する動的 extension $\mathcal{E}'$ は $\mathcal{E}' = \{\mathcal{E}_{exec'_1}, \mathcal{E}_{exec'_2}\}$ となる．

以上によりこの場合は，脅威によって議論木が変化するが，動的 extension には影響がない．

(Case 3) D_1 の葉ノードが P，D_2 の葉ノードが C の場合

図 1(3) について D_1 から実行された場合，D_2 内の P_d に C' が加えられるので新しい枝 D_3 が生成される．D_1 が実行された後は D_2 と D_3 のどちらか一方が実行されて終了するので，それぞれ異なる動的 extension が得られる．この AF に対する動的 extension $\mathcal{E}$ は $\mathcal{E} = \{P_r, C_2, C_3, C'\}$ である．

以上により，この場合は脅威によって動的 extension の要素数が増える．

(Case 4) D_1 の葉ノードが P，D_2 の葉ノードが C の場合

図 1(4) についてここでは

(i) C' が D_2 の葉ノード

(ii) C' が新しい枝

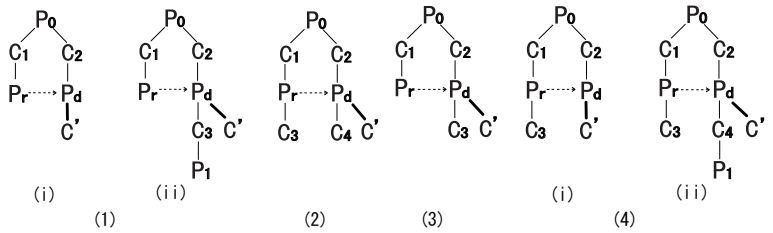

図 1 P から P への脅威

の二つの場合が考えられる．

しかし，いずれの場合もどのような順番で枝を実行しても追加された新しいノード C' を実行することなく終了してしまうので，動的 extension に変化はない．

AF が C から C への脅威を含む場合についても同様の考察ができる．

extension の変化について考えるとき，単純に枝の更新についてだけ考えるのでは十分でない．たとえ脅威によって新しいノードが加えられたとしても，extension にまで常に影響があるとは限らない．これは発言順の保持と現在の枝の実行可能なノードがすべて実行されるまで新しい枝は実行されないことが原因である．

4. 議論の別の実行手続き

APKC2 では議論において交互に論証が提示され，枝において実行可能なノードはすべて実行してから次の枝を選択するような実行手法になっている．議論システムを現実の議論を模倣だと見たときこの妥当性については検討の余地がある．以下はこれ以外の実行手続きについて述べる．

追い打ち型

APKC2 では実行の順番によってはある枝の勝敗と共にその時点で実行が終了してしまうことがある．追い打ち型は，ある枝の実行が停止したときの最後の発話者と同一の者から始まる別の枝を実行可能だとする方法である．

追い打ち型ではいずれの実行もすべてのノードを実行するため実行によって決定される動的 extension は一つに定まる．APKC2 は枝の実行順によって生成される extension が異なるのは，すべての枝を実行する前に途中の枝で停止する場合が存在するからであり，たとえ脅威によってノードが加えられたとしても，すべての枝が必ず実行される追い打ち型では APKC2 のような実行による動的 extension の差異は生じない．

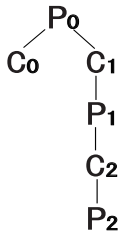

図 2 発言飛躍型の議論木

例 1 図 2 をある AF の初期議論木とする．APKC2 では左の枝から実行すると，P_0, C_0 をマークして P の敗北で終了するが，追い打ち型だと，その後右の枝に移って C_1, P_1, C_2, P_2 をマークして P の勝利で終了する．

追い打ち型の問題点は，順番に関係なくすべての枝を実行してしまうため，脅威が存在しない場合は初期議論木を構築した時点で勝敗が決定してしまうこと，脅威が存在する場合は勝敗が覆る場合もあるが，動的な勝敗判定ができない点である．

現実の議論と比較して，議論で勝利を得るために相手の意見を完全に潰すということを想定すると追い打ち型のような議論の進め方をするのではないかと考えられる．しかし，追い打ち型は**例 1** のような議論木ではあたかも C は自らの不用意な発言によってみすみす勝利を逃しているように見える．

発言飛躍型

発言飛躍型は，順番にはのっとるがある枝を実行中に他の枝へと移動できるものである．すなわち，たとえ今実行している枝にまだ実行可能なノードが存在していたとしても別の枝の実行可能なノードを実行することのできる方法である，APKC2 が一つの枝を最後まで実行する深さ優先探索のような振る舞いをするのに対して，発言飛躍型は枝間を行き来する幅優先探索のように振る舞う．

発言飛躍型は，APKC2 の実行では出現しない動的 extension が存在する．これはある枝を実行中に途中で別の枝に移動した時に動的に敗北する場合である．このとき，この実行に対する動的 extension を部分集合として含むような他の実行に対する動的 extension が存在する．

例 2 図 2 に対して APKC2 で右の枝から開始する実行を $exec_1$ とする．$exec_1$ に対する動的 extension は $\mathcal{E}_{exec_1} = \{C_0, P_1, P_2\}$ である．発言飛躍型だと P_0, C_1, P_1 を実行した後，左の枝へと移動して C_0 を実行し，停止するような実行 $exec_2$ が存在する．$exec_2$ の動的 extension は $\mathcal{E}_{exec_2} = \{C_0, P_1\}$ となり，動的 extension 間の関係は $\mathcal{E}_{exec_2} \subset \mathcal{E}_{exec_1}$ となる．

この方法は，議論の実行を探索問題としてとらえて抽象的な話をするには有効だが，実際の議論ではこのように議論の対象となる点が転々とする場合は非常に考えにくい．

5. おわりに

本発表では，動的議論システムに適した新しい意味論として動的 extension を提案した．そこでは，議論の実行順序ごとに extension を定めた．また，脅威を含む議論の実行によって extension がどのように変化するかについて調べ，動的 extension の特徴について述べた．

今回は単一の候補部分木内の脅威しか想定しなかったが，今後は複数の木にまたがる脅威がある場合の動的意味についても考察したい．

参 考 文 献

1) L.Amgoud, S.Parsons, and N.Maudet: Arguments, dialogue, and negotiation, ECAI2000, pp.338-342, 2000.
2) P.Baroni and M.Giacomin: Comparing Argumentation Semantics with Respect to Skepticism. Symbolic and Quantitative Approaches to Reasoning with Uncertainty, pp.210-221, LNCS4724, 2007.
3) T.Bench-Capon and P.Dunne: Argumentation in artificial intelligence, Artificial Intelligence, 171, pp.619-641, 2007.
4) C.Cayrol, F.D.de St-Cyr, and M-C Lagasquie-Shiex: Revision of an argumentation system. pp.124-134, KR2008, 2008.
5) M.Caminada: Semi-stable semantics. In COMMA2006, pp.121-130, 2006.
6) S.Coste-Marquis, C.Devred and P.Marquis: Prudent semantics for argumentation frameworks. In ICTAI2005, pp.568-572, 2005.
7) P.M.Dung: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence, 77, pp.321-357, 1995.
8) P.M.Dung, P.Mancarella and F.Toni: A dialectic procedure for sceptical, assumption-based argumentation. In COMMA2006, pp.145-156, 2006.
9) K.Okuno and K.Takahashi: Argumentation system with changes of an agent's knowledge base, IJCAI2009, pp.226-232, 2009.
10) K.Okuno and K.Takahashi: Argumentation System Allowing Suspend/Resume of an Argumentation Line ArgMAS2010, pp.145-162, 2010.
11) I.Rahwan, and G.Simari (eds.): *Argumentation in Artificial Intelligence*, Springer, 2009.