人工知能学会研究会資料　SIG-SWO-051-09

モンテカルロ木探索による記述論理の充足可能性判定

Description Logic Satisfiability using Monte Carlo Tree Search

高橋大樹[1]　兼岩憲[1]
Daiki Takahashi[1]　Ken Kaneiwa[1]

[1] 電気通信大学 大学院情報理工学研究科 情報・ネットワーク工学専攻
[1] Department of Computer and Network Engineering,
Graduate School of Informatics and Engineering, The University of Electro-Communications

Abstract: Web 情報をコンピュータで処理・活用するためにセマンティック Web ではメタデータやオントロジーが用いられる．記述論理は Web オントロジー言語 OWL の理論的基盤であり，オントロジーを記述するための言語や，推論を行うための知識ベースを提供する．命題論理の充足可能性問題に対してモンテカルロ木探索を適用した先行研究をもとに，本研究ではモンテカルロ木探索を用いて記述論理の充足可能性を判定する手法を提案する．評価実験では，実装した $\mathcal{ALC}$ 概念の充足可能性判定に対して提案手法が有用であることを示す.

1　はじめに

記述論理は，概念という特徴的対象もしくは名詞的・形容詞的語彙に特化して，その知識表現と推論の方法論を提案する論理体系である [1, 2]．その推論システムの実装により，セマンティック Web[3] のオントロジーに対する推論を可能にする．記述論理の標準的な推論アルゴリズムにはタブロー法があり，停止性・健全性・完全性が保証されている．記述論理は高い表現力を持ちながら，一階述語論理とは異なり推論の決定可能性が保証されている．しかし，判定する概念の表現力や規模によっては推論の効率が悪くなる恐れがある．

一般的に，論理表現では選言が推論過程において分岐を生じさせて計算量を増加させる．さらに，量化子（全称と存在）もまた推論の計算量を増加させる要因となる．したがって，効率的に論理的推論を行うために，論理式を連言標準形や節形式へ変換して導出を単純化する方法が用いられている．Previti らの先行研究 [4] では，連言標準形で表した命題論理式の充足可能性判定に対してモンテカルロ木探索が適用された．モンテカルロ木探索は強化学習に分類される学習法であり，囲碁 AI・スケジューリング・物理シミュレーションなど様々な分野において応用されている [5]．強化学習における手法は一般に確率的アルゴリズムであるため，計算量の減少が期待される．

本研究では，次の 3 つを提案することで強化学習による記述論理 $\mathcal{ALC}$ の推論手法を実現する．

- 連言標準形に準じた記述論理 $\mathcal{ALC}$ の平坦な概念表現
- $\mathcal{ALC}$ 概念の節集合に対する推論アルゴリズム
- モンテカルロ木探索による強化学習の推論戦略

記述論理ではほとんど用いられていない連言標準形の概念表現を新しく定義して，その節集合に対する推論ルールを実現する．さらに，その推論ルールを適用する導出過程をモンテカルロ木探索による状態遷移によって学習する．

本稿の構成は，次のようになっている．第 2 章では準備として，記述論理の構文・意味論，モンテカルロ木探索および先行研究について紹介する．第 3 章では記述論理における連言標準形を定義し，節集合に対する推論アルゴリズムについて述べる．第 4 章では，第 3 章で述べた推論アルゴリズムを強化学習として実装する上での戦略について述べる．第 5 章では提案手法の評価実験を行う．最後に，第 6 章で本稿の結論を述べる．

2　準備

2.1　記述論理

2.1.1　構文

記述論理の概念言語は，構成要素と構文規則の組み合わせによって異なる表現力をもつ言語ファミリーを形成する [1]．以降では概念言語の 1 つである $\mathcal{ALC}$ 言語について扱う．

$\mathcal{ALC}$ 言語は，概念名 A の集合 **CN**，ロール名 R の集合 **RN**，個体名 a の集合 **IN** および論理結合子 $\sqcap$（連

言），⊔（選言），¬（否定）と量化子 ∃（存在），∀（全称）から構成される．また，全てのインスタンスを含む最大概念 ⊤ および何も含まない空概念 ⊥ が **CN** に含まれる．$\mathcal{ALC}$ 概念は概念名 A，ロール名 R および任意の概念 C，D を用いて以下の構文規則によって帰納的に定義される．

$$A \mid \top \mid \bot \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$$

任意の概念とロール名および論理結合子を組み合わせることで，複雑な概念を表現できる．例として,「足をもつ動物」は以下のように表せる．

$$Animal \sqcap \exists hasPart.Leg$$

2.1.2 意味論

記述論理の解釈 $\mathcal{I}$ は，対象領域 $\Delta^{\mathcal{I}}$ と解釈関数 $\cdot^{\mathcal{I}}$ の対 $\left(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\right)$ で構成される．解釈 $\mathcal{I}$ によって対象領域 $\Delta^{\mathcal{I}}$ の要素により，概念名・ロール名・個体名に対して以下のように解釈が与えられる．

$A \in \mathbf{CN}$ に対して，$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$(特に$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}, \bot^{\mathcal{I}} = \emptyset$)
$R \in \mathbf{RN}$ に対して，$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
$\circ \in \mathbf{IN}$ に対して，$\circ^{\mathcal{I}} \in \Delta^{\mathcal{I}}$

また，複雑な $\mathcal{ALC}$ 概念の解釈については以下のように帰納的に定義される．

$$\begin{aligned}
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \left\{x \in \Delta^{\mathcal{I}} \mid \forall y \left[(x,y) \in R^{\mathcal{I}} \to y \in C^{\mathcal{I}}\right]\right\} \\
(\exists R.C)^{\mathcal{I}} &= \left\{x \in \Delta^{\mathcal{I}} \mid \exists y \left[(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\right]\right\}
\end{aligned}$$

ある概念 C について，$C^{\mathcal{I}} \neq \emptyset$ となる解釈 $\mathcal{I}$ が存在するならば，C は充足可能であるという．

2.2 モンテカルロ木探索

モンテカルロ木探索は，強化学習に分類される手法の 1 つである．強化学習では問題を繰り返しシミュレートすることで解法を導く．シミュレーションでは状態 s，行動 $a(s)$，報酬 $r(s,a)$ で構成される環境を考える．時刻 t に環境中の状態 s_t においてある行動 $a(s_t)$ を実行することで報酬 $r(s_t, a(s_t))$ を獲得し，次の状態 s_{t+1} へ遷移する．状態遷移を繰り返すことで報酬などの環境に関する知識を獲得し，各状態における最適な行動 $a^*(s)$ を学習する．

モンテカルロ木探索では状態をノード，行動をエッジとして表現した木構造のグラフを用いて探索を行う．モンテカルロ木探索のアルゴリズムの概要を Algorithm 1[5] に示す．ここで，s は状態，Δ は終端状態で獲得した報酬を表す．

Algorithm 1 モンテカルロ木探索アルゴリズムの概要

function MCTSSEARCH(s_0)
  create root node v_0 with state s_0
  while within computational budget **do**
    $v_l \leftarrow$ TreePolicy(v_0)
    $\Delta \leftarrow$ DefaultPolicy($s(v_l)$)
    Backup(v_l, Δ)
  return a(BestChild($v_0, 0$))

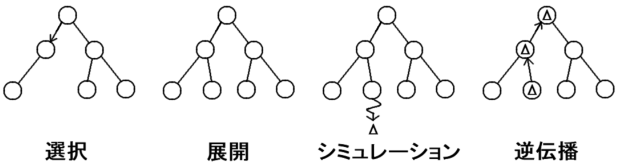

図 1: 各フェーズの模式図

モンテカルロ木探索では選択・展開・シミュレーション・逆伝播の 4 つのフェーズを繰り返し実行する．選択フェーズは TreePolicy に対応し，優先度に従って行動を選択・実行し，状態の遷移を繰り返す．ある状態において未実行の行動がある場合，その行動についてノード 1 つ分だけ展開し，シミュレーションフェーズへ移行する．このフェーズは DefaultPolicy に対応し，終端状態に到達するまで探索木の情報を用いずにシミュレートする．終端状態に到達した時点で報酬 Δ を獲得し，通過してきたパス中の各ノードに Δ を逆伝搬する．

選択フェーズにおける行動選択法である UCT (Upper Confidence Tree) では獲得した報酬の平均値およびノードの探索回数から各行動の優先度を決定する．ノード v から v' に遷移する行動について，UCT による行動優先度は式 (1) で表される．ただし，$R(v')$ はノード v' の報酬の総和，$n(v)$ はノード v の探索回数，c は推定価値の補正に関する係数である．

$$\frac{R(v')}{n(v')} + c\sqrt{\frac{2\ln(n(v))}{n(v')}} \tag{1}$$

式 (1) の第一項で次の状態の価値を推定し，その値が高い状態を優先的に選択する．一方で，学習初期は探索が不十分なため，第二項によって探索回数が少ない状態ほど値を高くすることで，状態価値が補正される．

2.2.1 UCTSAT

Previti ら [4] によって提案された UCTSAT$_{cp}$ および UCTSAT$_{sbs}$ は，命題論理の充足可能性判定にモンテ

カルロ木探索を適用したアルゴリズムである．この研究では，探索木のノードは命題論理式を表し，エッジは変数および代入する真偽値を表す．ノードが表す論理式において，エッジが表す変数に真偽値を代入して簡素化することで次ノードの論理式を得られる．したがって，あるノードにおいて充足可能であると判定されたとき，根ノードからそのノードまでのパスが論理式を満たす変数の値を示す．また，あるノードで矛盾が発生して充足不可能であることが確定したとき，そのノードをマーキング（closed）する．以降そのノードを探索させないため，探索時間を削減できる．

3　記述論理の節集合に対する推論

3.1　記述論理の連言標準形

本節では，連言標準形に準じた新たな $\mathcal{ALC}$ 概念表現を定義する．任意の概念名 A とその否定 $\neg A$ および後に定義する連言標準形 F を値にもつ任意のロール概念 $\exists R.F, \forall R.F$ を概念リテラルと呼び，L で表す．概念リテラルの選言を節と呼び，CL で表す，節の連言を連言標準形と呼び，F で表す，A を概念名，R をロール名，$L_1, \ldots, L_m$ を概念リテラル，$CL_1, \ldots, CL_n$ を節，F を連言標準形とすると，以下のように帰納的に定義できる．

$$
\begin{aligned}
L &= A \mid \neg A \mid \exists R.F \mid \forall R.F \\
CL &= L_1 \sqcup \ldots \sqcup L_m \\
F &= CL_1 \sqcap \ldots \sqcap CL_n
\end{aligned}
$$

概念名 A とその否定 $\neg A$ は，互いに他方の補リテラルである．さらに，F_1 が F_2 の補リテラルならば，$\exists R.F_1$ と $\forall R.F_1$ はそれぞれ $\forall R.F_2$ と $\exists R.F_2$ の補リテラルである．ここで，概念リテラル L の補リテラルを $\overline{L}$ と表す．

任意の $\mathcal{ALC}$ 概念 C は，次の手順により C と同値な連言標準形 ($\mathrm{CNF}(C)$ と表す) に変換できる．すなわち，$C \equiv \mathrm{CNF}\,(C)$ となる．

1. ド・モルガンの法則と二重否定を用いて，否定が概念名のみに現れるように変換する．

$$
\begin{aligned}
\neg\,(C \sqcap D) &\equiv \neg C \sqcup \neg D \\
\neg\,(C \sqcup D) &\equiv \neg C \sqcap \neg D \\
\neg\,(\exists R.C) &\equiv \forall R.\neg C \\
\neg\,(\forall R.C) &\equiv \exists R.\neg C \\
\neg\neg C &\equiv C
\end{aligned}
$$

2. 概念全体および各ロール概念に対して交換法則と分配法則を用いて，選言の中に連言が含まれないように変換する．

$$
\begin{aligned}
C \sqcup D &\equiv D \sqcup C \\
C \sqcap D &\equiv D \sqcap C \\
C \sqcup (D \sqcap E) &\equiv (C \sqcup D) \sqcap (C \sqcup E) \\
\exists R.C &\equiv \exists R.\mathrm{CNF}\,(C) \\
\forall R.C &\equiv \forall R.\mathrm{CNF}\,(C)
\end{aligned}
$$

3. 次の結合法則が成り立つことから，選言あるいは連言が連続する場合は，括弧を省略して $C \sqcup D \sqcup E$ および $C \sqcap D \sqcap E$ に変換する．

$$
\begin{aligned}
(C \sqcup D) \sqcup E &\equiv C \sqcup (D \sqcup E) \\
(C \sqcap D) \sqcap E &\equiv C \sqcap (D \sqcap E)
\end{aligned}
$$

3.2　節集合の推論アルゴリズム

任意の $\mathcal{ALC}$ 概念 C から変換した連言標準形 $\mathrm{CNF}(C) = CL_1 \sqcap \ldots \sqcap CL_n$ を以下の節集合で表す．

$$\{CL_1, \ldots, CL_n\}$$

ここで，各節 $CL_i = L_1 \sqcup \ldots \sqcup L_m$ をリテラル集合 $\{L_1, \ldots, L_n\}$ とする．特に，$|CL| = 1$ のとき単位節といい，$|CL| = 0$ のとき空節という．

連言標準形の概念 $F = \mathrm{CNF}(C)$ について充足可能性を判定するアルゴリズムは次の通りである．各ノードを概念集合 S_i，各エッジを推論ルールの適用とした木構造を導出木という．以降，概念集合 S_i の要素は節集合で表された概念とする．F の導出木は根ノードを $S_0 = \{F\}$ とし，各ノード S_i に推論ルールを適用して得られた結果をその子ノード S_{i+1} とする．推論ルールは各節集合 $F \in S_i$ に対して 1 回ずつ実行され，条件を満たす限り適用される．

(A1) $|CL| \geq 2$ かつ $L \in CL$ が存在するとき，任意の $CL' \in F$ に対して以下を実行する．
- (i) $L \in CL'$ ならば $CL' \to \{L\}$
- (ii) $L \notin CL'$ かつ $\overline{L} \in CL'$ ならば $CL' \to CL' \backslash \{\overline{L}\}$

(A2) $\forall R.F_1 \in CL$ が存在するとき，任意の $CL' \in F$ に対して以下を実行する．
- (i) $\forall R.F_1 \in CL'$ ならば $F \to F \setminus \{CL'\}$
- (ii) $\exists R.F_2 \in CL'$ ならば $\exists R.F_2 \to \exists R.(F_1 \cup F_2)$

(A3) 任意の $CL \in F$ が単位節 $\{A\}, \{\neg A\}$ または $\{\exists R.F'\}$ であり，$\{\exists R.F_1\} \in F$ が存在するとき，以下を実行する．
- (i) $F \to F \setminus \{\{\exists R.F_1\}\}$ および $S_i = S_i \cup \{F_1\}$

導出木のあるノードに対してどのルールも適用できないとき，その概念集合 S_i は完全である．$S_0 = \{F\}$ から導出された完全な概念集合 S_i が存在し，空節も矛盾も含まないとき，F は充足可能であると判定する．

ルール A1 の適用条件は，2 つ以上の概念リテラルを含む節 CL が存在することである．その節 CL から概念リテラル L を選択する．全ての節 CL'（単位節を含む）について，L を含むならば L 以外の概念リテラルを節から除去し，L の補リテラルを含むならば $\overline{L}$ を節から除去する．すなわち，概念リテラル L を選択したとき以下のように変換される．

$$L \sqcup L_1 \sqcup \ldots \sqcup L_m \rightarrow L$$
$$\overline{L} \sqcup L_1 \sqcup \ldots \sqcup L_m \rightarrow L_1 \sqcup \ldots \sqcup L_m$$

ルール A2 の適用条件は，全称ロール概念 $\forall R.F_1$ が存在することである．$\forall R.F_1$ を含む節 CL' を全て除去した後，同じロール名 R を用いた全ての存在ロール概念 $\exists R.F_2$ を，節集合 F_1 と合わせた $\exists R.(F_1 \cup F_2)$ へ変換する．

ルール A3 の適用条件は，全ての節が全称ロール概念以外の単位節であり，ある存在ロール概念の単位節 $\{\exists R.F_1\}$ が含まれることである．この単位節を節集合 F から除去して部分概念 F_1 を概念集合 S_i に追加する．

導出木において，充足不可能性は各ルールの適用前後で親ノードから子ノードへ継承される．すなわち，概念集合 S_i のある要素が充足不可能であるならば，ルール A1，A2，A3 より導出された概念集合 S_{i+1} に充足不可能な要素が存在する．この対偶により，S_{i+1} の全ての概念が充足可能ならば，S_i の全ての概念も充足可能である．完全な概念集合は直ちに充足可能性が決定され，直前のルール適用前の概念集合についても充足可能性が決定される．したがって，根ノードがもつ $S_0 = \{F\}$ の充足可能性もいずれ決定される．充足不可能性の継承の妥当性は以下の通りである．

ルール A1 の適用によって，節内の選言が簡略化される．ここで任意の概念リテラル L, L' について $L \sqsubseteq (L \sqcup L')$ より，$L \sqcup L'$ が充足不可能ならば L も充足不可能である．同様に，$L' \sqsubseteq (\overline{L} \sqcup L')$ より，$\overline{L} \sqcup L'$ が充足不可能ならば L' も充足不可能である．

ルール A2 の適用によって全称ロール概念 $\forall R.F_1$ が除去され，$\exists R.F_2$ が $\exists R.(F_1 \cup F_2)$ へ変換される．ここで $F \sqsubseteq F \backslash \{CL'\}$ が成り立つので，$F \backslash \{CL'\}$ が充足不可能ならば F は充足不可能である．さらに，$\forall R.F_1 \sqcap \exists R.F_2$ が充足不可能のとき，$F_1 \cup F_2$ $(= F_1 \sqcap F_2)$ が充足不可能である．ゆえに，$\exists R.(F_1 \cup F_2)$ $(= \exists R.(F_1 \sqcap F_2))$ も充足不可能である．

ルール A3 の適用によって存在ロール概念の単位節 $\{\exists R.F_1\}$ が削除され，S_i に F_1 が追加される．このとき，$\exists R.F_1$ が充足不可能ならば F_1 も充足不可能である．

推論の停止性（決定可能性）について述べる．概念リテラルの個数は有限であるので，いずれルール A1 を適用できなくなる．ルール A2，A3 は少なくとも外側のロール記号を 1 つ削除するので，有限個のロール数で終了する．したがって有限ステップで終端状態に到達し，概念の充足可能性が決定される．

4　強化学習による推論戦略

4.1　ルールとリテラルの選択

ルール A2 は存在ロール概念を含む節によって新たな概念の追加を発生させる．すなわち，$\forall R.F_1$ と $\exists R.F_2$ の部分概念 F_1 と F_2 との間についてもさらに充足可能性を判定するので，推論の計算量を増大させる．ここで，ルール A1 を優先的に適用して存在ロール概念が削除されれば不要な推論を回避できる．また，ルール A3 はルール A2 の適用に依存するため，優先順序はルール A1→ ルール A2 → ルール A3 となる．

ルール A1 は，選択された概念リテラルの単位節を除いて全て削除するため，再びルール A1 の適用条件を満たすことはない．したがって，ルール A1 の適用回数は「概念名の種類数 + ロール概念の個数」以下となる．ルール A1 における概念リテラルの除去によって概念名の種類数が減少する．したがって，総適用回数の減少のために出頻度が最も多い概念名を選択する．一方で，ロール概念はルール A2，A3 の適用を誘発するので，ルール A1 では単純な概念名を優先的に選択する．

4.2　DLSAT

本節ではモンテカルロ木探索により充足可能性を判定するルール選択の過程を学習する（DLSAT）．DLSAT における探索木では概念集合を状態 $s = S_i$ として表し，ルール･対象概念および選択された要素（ルール A1 なら概念リテラル，ルール A2，A3 ならロール概念）を行動 $a = (\mathrm{A1}, F, L), (\mathrm{A2}, F, \forall R.F_1)$ または $(\mathrm{A3}, F, \exists R.F_1)$ として表す．例えば，図 2 のようにルールが選択される．

ある状態において完全な概念集合が得られたとき，終端状態になる．完全な概念集合が空節または矛盾を含まないならば，充足可能となり探索が終了する．一方，空節または矛盾を含んで充足不可能となったノードは，closed とマーキングして行動選択の候補から除外される．各終端状態 S_n では必ず充足可能性が決定されるため，その報酬 Δ は SAT（充足可能）もしくは UNSAT（充足不可能）のいずれかとなる．終端でない各状態 S_i の報酬は，子ノードの状態 S_{i+1} から報酬 Δ を逆伝搬して得られる．ルール A1 では選言で分岐した子ノー

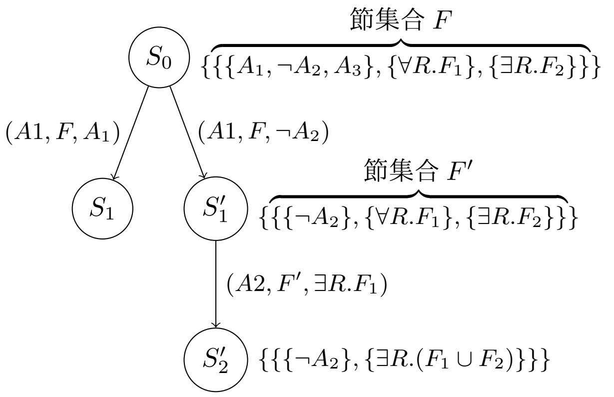

図 2: モンテカルロ木探索によるルール選択

ドの 1 つが充足不可能でも，他の子ノードをチェックしなければまだ親ノードの充足可能性は決定されない．

ルール A1 を実行した後，子ノードの状態 S_{i+1} で報酬 Δ =UNSAT を得た場合を考える．全ての子ノードがマーキングされている場合は，親ノードもマーキングして Δ =UNSAT が逆伝搬される．そうでないとき，子ノードが展開済みならば $\Delta = 0$ を逆伝搬し，子ノードが未展開ならばヒューリスティック報酬を逆伝搬する．行動 $a = (A1, F, L)$ を実行したとすると，ヒューリスティック報酬は以下により算出される．

$$\sum_{CL \in F} \frac{\delta(CL, L)}{|CL|} \tag{2}$$

ここで，$\delta(CL, L)$ はルール A1 における概念リテラル L の選択によって節 CL が簡略化される前と後の差をリテラル数で表す．また，ルール A1 適用後の子ノードで得た報酬が $\Delta \neq$UNSAT のとき，親ノードの報酬は式 (2) に Δ を加算した値とする．

一方，ルール A2，A3 では選言による分岐が発生しないため，子ノードの状態 S_{i+1} で得られた報酬 Δ をそのまま親ノードへ逆伝搬する．

5 実験

本研究の推論方法を評価するために，$\mathcal{ALC}$ 概念の節集合を用いた充足可能性判定の実験を行う．推論アルゴリズムは Python で実装し，実行環境は OS:Windows 10 Home 64bit, CPU:Intel(R) Core(TM) i7-8565U @ 1.80GHz 1.99GHz, 実装 RAM:16.0GB である．

命題論理式のベンチマークセット[1]を用いてテストデータを加工する．命題変数を記述論理の概念名とみなして，量化子とロール名を追加して．$\mathcal{ALC}$ 概念を生成する．テストデータの加工方法は，以下のように論理式内のいくつかの節を量化する．

$$CL_1 \sqcap CL_2 \sqcap CL_3 \rightarrow CL_1 \sqcap \forall R.CL_2 \sqcap CL_3$$

量化する節数は全体の 10%（その内存在量化は 30%）であり，ロール名は 1 種類のみである．対象データ数は，概念名の種類数が 20 個のものが 1000，50 個および 75 個のものがそれぞれ 100 である．

モンテカルロ木探索による推論手法を評価するために，選択フェーズにおいて根ノードから各ルールを適用する 2 つの探索方法と比較する．1 つ目は，各ルールで選択する概念リテラルやロール概念を候補から一様ランダムに決定する．2 つ目は，子ノードが展開済みならば直近に訪れた子ノードへの行動（ルールの適用）を再び選択して深さ優先探索を行う．どちらの手法でも，closed とマーキングされた子ノードは選択から除外される．なお，UCT の式 (1) における係数 c の値は参考論文 [4] と同様に 0 で固定した．

各手法に対する推論結果として，充足可能と判定した時点での探索木のサイズ（エッジの総数）を表 1 に示す．この結果は全データに対する平均値であり，太字は最小値を表す．

手法	概念名の種類数		
	20	50	75
一様ランダム	**6.6**	41.2	93.7
深さ優先探索	7.0	36.2	90.7
DLSAT	7.3	**24.6**	**50.1**

表 1: 節の量化を含む $\mathcal{ALC}$ 概念の充足可能性判定の性能

もう 1 つのテストデータの加工は，いくつかの概念名および節について，単一の概念名または節全体を量化することである．例として以下の加工が施される．

$$CL_1 \sqcap CL_2 \sqcap (A_1 \sqcup A_2) \rightarrow CL_1 \sqcap \forall R.CL_2 \sqcap (\exists Q.A_1 \sqcup A_2)$$

量化する節数は全体の 10%（その内存在量化は 30%）であり，ロール名は 2 種類とし，量化が入れ子になることを許容した．2 つ目のテストデータに対する推論結果を表 2 に示す．ただし，概念名の種類数が 75 のときの深さ優先探索は 6 時間経過してもプログラムが停止しなかったため，timeout とした．

表 1 と表 2 のどちらについても，概念名の種類数が少ないシンプルな概念に対する推論では手法による結果の差はほぼみられなかった．一方で複雑な概念の推論では，DLSAT は 2 つの比較手法よりも小さい探索領域から解を導出できている．したがって，DLSAT はモンテカルロ木探索により獲得した報酬を用いて行動を選択して，効率的に解を導いている．

[1] `https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html`, Uniform Random-3-SAT

手法	概念名の種類数		
	20	50	75
一様ランダム	5.8	62.1	320.3
深さ優先探索	6.0	537.2	timeout.
DLSAT	**5.5**	**34.1**	**286.7**

表 2: 節やリテラルの量化を含む $\mathcal{ALC}$ 概念の充足可能性判定の性能

6 まとめ

本研究ではモンテカルロ木探索を用いた記述論理の充足可能性判定アルゴリズムを提案した．記述論理の $\mathcal{ALC}$ 概念に対して，連言標準形に準じた表現を新しく定義し充足可能性を判定する推論ルールが導入されている．さらに，モンテカルロ木探索により推論過程を選択して強化学習で解く方法を実現している．実験結果より，提案手法は単純な戦略に比べて少ない計算ステップで効率的に判定可能であることを示した．

今後の課題として，報酬の改良による効率化，表現力の高い記述論理への拡張，記述論理用のベンチマークセットへの適用や Web 上に実在する文章から生成される概念表現への応用が考えられる．

参考文献

[1] 兼岩憲: 記述論理と Web オントロジー言語, オーム社. (2009)

[2] Baader, Franz., Calvanese, Diego., McGuinness, Deborah., Nardi, Daniele., Patel-Schneider, Peter.: *The Description Logic Handbook: Theory, Implementation, and Applications*, 2nd Edition. Cambridge University Press, Cambridge (2007)

[3] 兼岩憲: セマンティック Web とリンクトデータ, コロナ社. (2017)

[4] Alessandro, Previti., Raghuram, Ramanujan., Marco, Schaerf., Bart, Selman.: Monte-Carlo Style UCT Search for Boolean Satisfiability, *Congress of the Italian Association for Artificial Intelligence*, Springer, Berlin, Heidelberg, pp. 177–188 (2011)

[5] Browne, Cameron B., Powley, Edward., Whitehouse, Daniel., Lucas, Simon M., Cowling, Peter I., Rohlfshagen, Philipp., Tavener, Stephen., Perez, Diego., Samothrakis, Spyridon., Colton, Simon.: A Survey of Monte Carlo Tree Search Methods, *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, pp. 1–43 (2012)