

DTS の部分体系を用いた定理自動証明器への等号型の導入

理学専攻 情報科学コース 2040646 大洞 日音 (指導教員: 戸次 大介)

1 はじめに

理論言語学の目的の一つは言語現象を説明することであり, 検証可能性の確保が重要であるが, 大きなデータについて理論的予測を手計算で確認するのは非常にコストが高い. そのため, 近年は形式統語論・意味論に基づく統語解析器・意味計算を定理証明器と組み合わせることにより, テキストから推論までのパイプライン処理を自動で行うというアプローチが盛んに研究されている. パイプライン処理の概要を図 1 に示す.

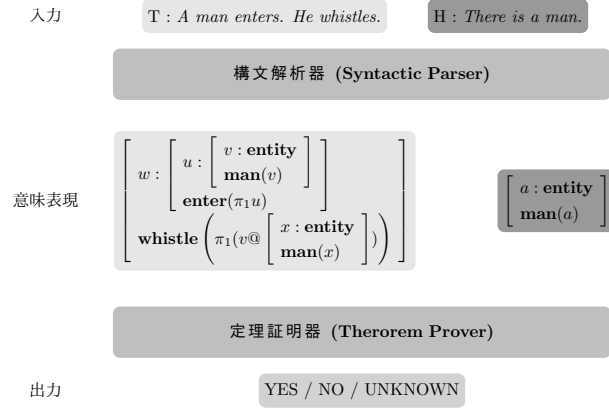


図 1: テキストから推論までを行うパイプラインの構成

形式意味論の分野においては談話表示理論 (Discourse Representation Theory; DRT) のようなモデル論的意味論が主流であるが, 照応現象・前提現象などの一部の言語現象は DRT を代表とする従来の形式意味論の枠組みで構造的に扱うことは難しいとされている. 依存型意味論 (Dependent Type Theory; DTS)[2] は自然言語のための証明論的意味論であり, 照応現象を含む自然言語の意味現象を統一的に扱うことができる [6] とされている. 意味表現として DTS を採用した構文解析器としては lightblue [1] が知られている. 対して DTS の計算に用いることのできる定理証明器は Alligator [4] や Sato2016 [8] が存在するが, これらは自然言語における時間関係や比較の説明に必要とされている (cf.[3], [7]) 自然数型や等号型をカバーしていない.

本研究では lightblue との接続を目指し, Alligator をプログラミング言語 Haskell で再実装した証明器 Alligator_H を改良. 等号型導入を行うことで DTS の等号型を含む部分体系のための定理自動証明器 wani を実装した. これは DTS の意味表示に該当する Haskell のデータ型を入力とした証明探索のための関数を他プログラムから import して実行することや, TPTP [5] 形式の問題を入力としてコマンドラインから実行することが可能である.

2 前向き推論と後ろ向き推論

Alligator では前向き推論と後ろ向き推論の二つの段階に分けて推論を行なう. 前向き推論では前提と結論が与えられたとき前提に規則を適用することで証明木を上から下に構成するのに対して, 後ろ向き推論では示したい命題が真であるためにはどのような命題が真であるかという視点で証明木を下から上に構成する. 前向き推論は前提と規則から導き出せる内容のみ取り扱うため推論の自由度が低い一方, 後ろ向き推論は制限がないため与えられた規則によっては停止性が保証されない.

3 Alligator_H からの変更

Alligator_H は DTS の規則を用いて愚直な深さ優先探索を行うが, この方法は人間が計算で行う照明と比べると非効率な回り道をしてしまうことがある. たとえば人間が手で証明を行う際は, 無意識に一度失敗した計算に類したものを堂々巡りを避けようとする, wani では探索の枝切りという方法で計算時間の短縮を目指した.

3.1 後ろ向き推論部分全体についての変更

Ⓐ 後ろ向き推論の深さ優先探索の途中で証明が見つかった場合は一部の場合を除いて計算を切り上げるようにした. また, Ⓑ 規則の適用を弱い順に優先するように変更した. Ⓐ だけでも計算量を小さくすることはできるが, Ⓑ と組み合わせることで, 最悪計算量の大幅な削減を期待できる.

3.2 (ΣI)/(ΠE) についての変更

依存型についての規則である (ΣI) や (ΠE) でははじめに前件部分について証明探索を行った上で, 見つかった証明項を受けて後件部分についての証明探索を行う. ここで後件に依存されていない前件部分は証明項が複数見つかったとしても 1 つだけ採用するように変更した.

4 等号型の導入

DTS における等号型に関する規則を図 2 に示す.

等号型導入に当たっては等号型除去規則 (=E) の下段として許容する型の制限がないことが大きな壁となる.

本研究では等号型除去規則をそのまま導入するのではなく図 3 に該当する操作を前向き推論に追加した.

$$\frac{\Gamma \vdash_{SF} a : A \quad A' \in \text{eqForward}_2(\Gamma, A)}{\Gamma \vdash_{EF} \text{eqE}(a) : A'}$$

図 3: wani における等号型除去操作

$$\frac{\Gamma \vdash_{SF} a : A \quad \Gamma \vdash_{EF} a : A}{\Gamma \vdash_F a : A} \quad \frac{\Gamma \vdash_F a : A}{\Gamma \vdash_F a : A}$$

図 4: wani における前向き推論

図 3 において \vdash_{SF} は等号型除去操作を行われていな

$$\begin{array}{c}
\frac{\Gamma \vdash A : \text{type} \quad \Gamma \vdash M : A \quad \Gamma \vdash N : A}{\Gamma \vdash M =_A N : \text{type}} \quad (=F) \\
\frac{\Gamma \vdash E : M =_A N \quad \Gamma \vdash P : (x : A) \rightarrow (y : A) \rightarrow (z : x =_A y) \rightarrow \text{type} \quad \Gamma \vdash R : (x : A) \rightarrow Pxx(\text{refl}_A(x))}{\Gamma \vdash \text{idpeel}_E^P(R) : PMNE} \quad (=E) \\
\frac{\Gamma \vdash A : \text{type} \quad \Gamma \vdash M : A}{\Gamma \vdash \text{refl}_A(M) : M =_A M} \quad (=I)
\end{array}$$

図 2: DTS における等号型の導入則, 除去則, 形成規則

$$\begin{array}{c}
\frac{\Gamma \vdash_T c_0 : t_0 \quad \Gamma \vdash_T t_0 : \text{type}}{[u_0 : t_0, u_1 : u_0 =_{t_0} c_0] \Rightarrow u_0 \in \text{eqForward}_1(\Gamma, c_0)} \\
\frac{\vec{A}_1 \Rightarrow A_2 \in \text{eqForward}(\Gamma, A) \quad \vec{B}_1 \Rightarrow B_2 \in \text{eqForward}(\Gamma, B)}{\vec{A}_1, \vec{B}_1 \Rightarrow A_2 =_T B_2 \in \text{eqForward}_2(\Gamma, A =_T B)} \quad \frac{\text{eqForward}(\Gamma, A) = [] \quad \vec{B}_1 \Rightarrow B_2 \in \text{eqForward}(\Gamma, B)}{\vec{B}_1 \Rightarrow A =_T B_2 \in \text{eqForward}_2(\Gamma, A =_T B)}
\end{array}$$

図 5: eqForward_1 と等号型についての eqForward_2 の定義

い前向き推論, \vdash_{EF} は等号型除去操作を行った前向き推論, \vdash_F は前向き推論で示された内容であるということを表す。図 4 のとおり, 等号型除去操作を行う前の結果も行った後の結果もどちらも前向き推論の結果として扱う。

等号型除去操作の内部では wani の採用する Arrowterm について構造的に再帰する関数 eqForward を用いる。 eqForward は, 命題部分の形にかかわらず適用される eqForward_1 と命題の形によって動きの異なる eqForward_2 から構成される。

eqForward_2 の一部と eqForward_1 を図 5 に示す。

5 評価

5.1 計算時間短縮のための変更の評価

TPTP ライブラリの SYN ドメインで fof 形式で記述された問題を用いて加えた枝刈りについての評価を行った。深さ制限 9, 制限時間 60 秒での Alligator_H と wani についての評価の結果を表 1 に示す。

	Actual: YES	Actual: UNK
predicted: YES	44	0
predicted: UNK	227	83
	Actual: YES	Actual: UNK
predicted: YES	76	0
predicted: UNK	195	83

表 1: Alligator_H(上) と wani(下) の結果

wani では 32 問の問題を正しく回答できるようになっており, また Alligator_H が正解している問題は wani も正解していることを確認した。この結果から今回の範囲の問題については加えた制限が悪影響を及ぼしておらず, さらに時間または深さの制限によって探索を打ち切るケースが減ったということが言える。

5.2 等号型についての拡張の評価

等号型を含む簡単な問題について wani が適切な証明木を返すことを確認した。

$$\begin{array}{l}
\text{en} : \text{type}, x : \text{en}, y : \text{en}, z : \text{en}, \\
v_1 : x =_{\text{en}} y, v_2 : y =_{\text{en}} z \vdash ? : x =_{\text{en}} z
\end{array}$$

図 6: 等号型を含む問題例

図 6 に示す問題について wani で証明を行うと深さ制限 3 では 4503 ステップの推論を深さ制限 4 では 146939 ステップの推論を深さ制限 5 では 3984162 ステップの推論を行っていた。

これは人間が同じ問題を解いたときに辿る過程を考えると非常に多いため, 更なる制限を設けることでより効率的な計算を行うことができると考えられる。

6 おわりに

本研究では, DTS の等号型を含む部分体系のための証明探索アルゴリズムについて考察し, その実装を行った。今後は証明探索の更なる効率化や自然数型についての拡張を行いたい。また, 日本語 CCG パーザ lightblue[1] と組み合わせることによって, パイプラインを構築し, DTS に基づく検証可能な意味計算システムを実現することが今後の課題である。

参考文献

- [1] Daisuke Bekki and Ai Kawazoe. Implementing variable vectors in a ccg parser. In *LACL2016, Nancy, France, C.Retore and S.Pogodalla (Eds)*, pp. 52–67, Heiderburg, 2016.
- [2] Daisuke Bekki and Koji Mineshima. *Context-Passing and Underspecification in Dependent Type Semantics*, pp. 11–41. Springer International Publishing, Cham, 2017.
- [3] Izumi Haruta, Koji Mineshima, and Daisuke Bekki. A ccg-based compositional semantics and inference system for comparatives. *CoRR*, Vol. abs/1910.00930, pp. 47–56, 2019.
- [4] Paul Piwek. The alligator theorem prover for dependent type systems: Description and proof sample. In *Proceedings of the Inference in Computational Semantics Workshop*, pp. 157–163, 2006.
- [5] Geoff Sutcliffe. The tptp problem library and associated infrastructure. *J. Autom. Reason.*, Vol. 59, No. 4, pp. 483–502, December 2017.
- [6] Yukiko Yana, Koji Mineshima, and Daisuke Bekki. Variable handling and compositionality: Comparing drt and dts. *Journal of Logic, Language and Information*, Vol. 28, pp. 261–285, 05 2019.
- [7] 大西舞子, 谷中瞳, 峯島宏次, 戸次大介. CCG と定理証明器による自然言語の時間関係認識. 人工知能学会全国大会論文集, Vol. JSAI2020, pp. 1E3GS904–1E3GS904, 2020.
- [8] 佐藤未歩. 依存型意味論の証明探索とその実装. 修士論文, お茶の水女子大学大学院, 2016.