

## 発表概要

明示的な  $\alpha$  変換を用いた  $\alpha$  単一化山口 文彦<sup>†1</sup> 齋藤 博昭<sup>†1</sup>

単一化は、述語を持つ論理系の上で自動推論を行う際の基本操作である。従来、構文的等価性という、最も強い等価性に基づいた単一化が多く用いられてきたが、より弱い等価性に基づいた単一化を用いることで、系の表現力の向上や、推論ステップ数の減少が期待できる。そのような単一化を言語処理系に用いた例として、Urban らの  $\lambda$  Prolog がある。 $\lambda$  Prolog では、項の表現として、変数を束縛することを許し、束縛変数の名前の違いを無視する  $\alpha$  等価性に基づいた単一化を用いている。しかし、彼らの用いた単一化アルゴリズム Nominal Unification では、項  $F, F', G$  について、 $F$  と  $G$  が単一化可能で  $F$  と  $F'$  が  $\alpha$  等価であるとき、 $F'$  と  $G$  が単一化不能になる場合がある。本発表では、単一化の操作として、 $\alpha$  変換を明示的に返すような単一化アルゴリズムを提案する。また、提案アルゴリズムが、 $F$  と  $G$  を単一化するならば、 $F$  と  $\alpha$  等価なすべての項  $F'$  について、 $F'$  と  $G$  を単一化することを示す。

 $\alpha$  Unification with Explicit  $\alpha$  ConversionFUMIHIKO YAMAGUCHI<sup>†1</sup> and HIROAKI SAITO<sup>†1</sup>

Unification is the basic operation of automated reasoning on some predicate logic. Popular unification algorithms are based on the strongest equality i.e. syntactical equality. Unification based on weaker equality will empower the expression of the system and will decrease the step of reasoning.  $\lambda$  Prolog is an instance of such reasoning applied to programming language. In  $\lambda$  Prolog, bound variable can be expressed and unification method named nominal unification, which is based on  $\alpha$  equality, is used. However, nominal unification has following problem: when terms  $F$  and  $G$  are unifiable,  $F$  and  $F'$  are  $\alpha$  equivalent, but  $F'$  and  $G$  are not always unifiable. In this presentation, an unification algorithm, which returns  $\alpha$  conversions explicitly, is proposed. And it will be shown that any term  $F'$  which is  $\alpha$  equivalent to  $F$  is also unifiable with  $G$ , under the condition that  $F$  and  $G$  are unifiable.

<sup>†1</sup> 慶應義塾大学理工学部情報工学科

Department of Information and Computer Science, Keio University