

# Abstracting Effect Systems for Algebraic Effect Handlers

吉岡拓真\*1 関山太郎\*2 五十嵐淳\*1

\*1: 京都大学 \*2: 国立情報学研究所

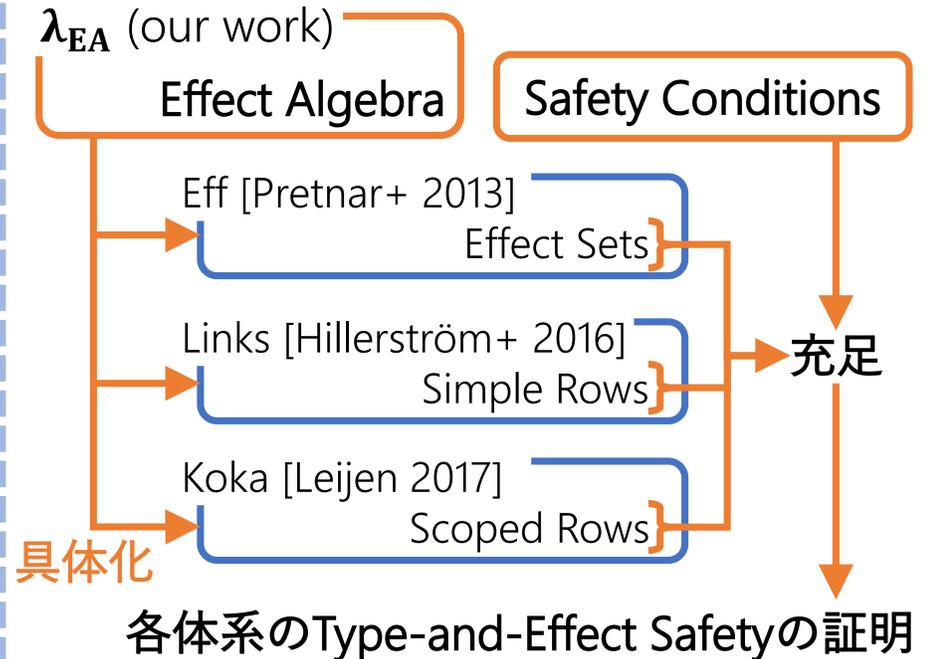
## 動機と目的

Effect Handlerのための既存のEffect SystemにはEffectの表現方法とその操作において様々な方式があり、それらの間の関係は不明であった

➡ 目標: 共通点と相違点を形式化

## 本研究の貢献

- Effectの表現・操作をEffect Algebraとして抽象化
- Effect Algebraをパラメータに持つ体系  $\lambda_{EA}$
- Type-and-Effect SafetyのためにEffect Algebraに要求される性質である、Safety Conditionsの同定



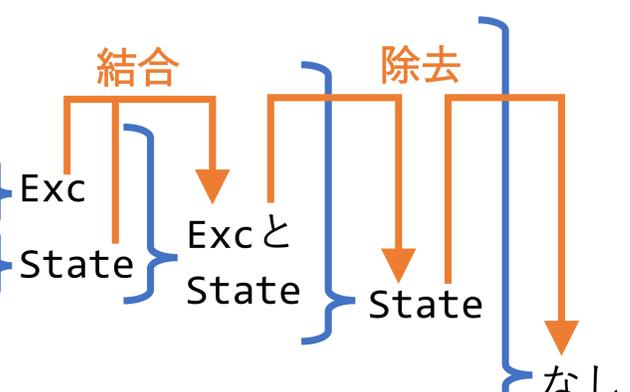
## Algebraic Effect Handler、Effect System、Effect Safety

- Algebraic Effect Handlerは副作用の発生を表すoperationとその挙動を定めるhandlerからなる
  - 未ハンドルのoperationは実行時エラーを引き起こす (cf. 未捕捉の例外)
- Effect Systemは全てのoperationが適切にハンドルされること (Effect Safety) を静的に保証

例: メモリセルの数値を  $x$  で割りその結果を書き込むプログラム ( $x = 0$ ならば例外を発生)

```

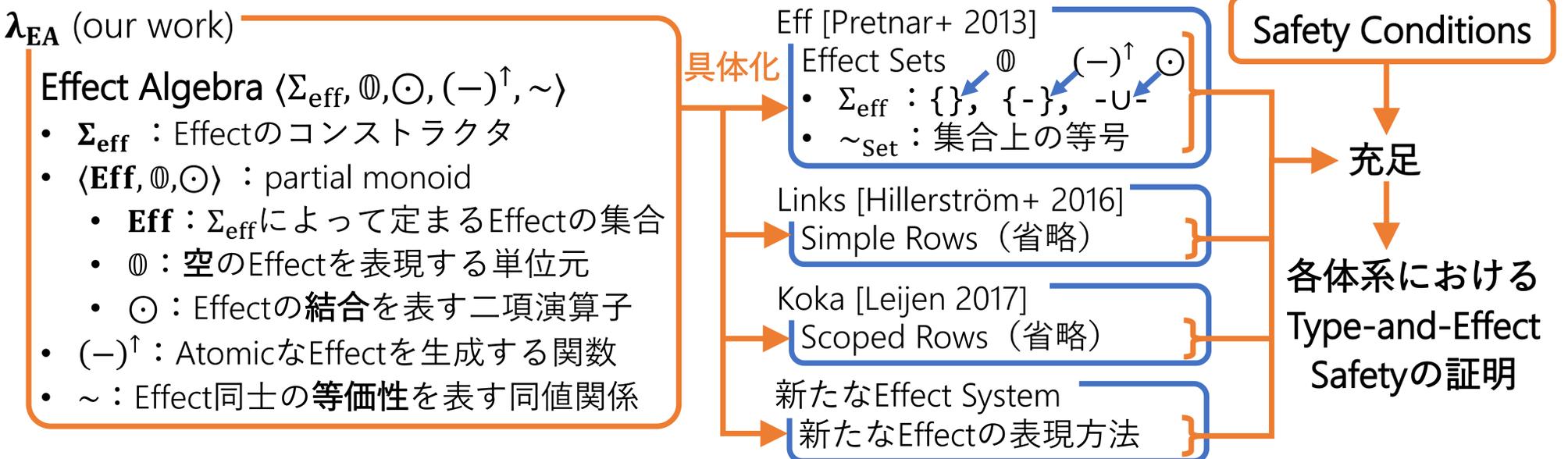
handle <State>
  handle <Exc>
    if x = 0
      then raise ()
      else set (get () / x)
  with { raise -> ... }
with { get -> ..., set -> ... }
    
```



- 未ハンドルのoperationの情報をEffect Systemは追跡する
- Set表現:  $\{\text{Exc}, \text{State}\}, \{\text{State}\}, \{\}$
- Row表現:  $\langle \text{Exc} | \langle \text{State} | \langle \rangle \rangle \rangle, \langle \text{State} | \langle \rangle \rangle, \langle \rangle$

## Effect Systemの抽象化とSafetyの証明: Effect Algebra、 $\lambda_{EA}$ 、Safety Conditions

Theorem (Type-and-Effect Safety) 与えられたEffect AlgebraがSafety Conditionsを満たすならば、そのEffect Algebraによる  $\lambda_{EA}$  のインスタンスはType-and-Effect Safeである



## $\lambda_{EA}$ で取り扱える他の設定

- Shallow Handlers [Kammer+ 2013]
- Lift Coercions [Biernacki+ 2018]
- Type-Erasure Semantics [Biernacki+ 2019]

## 今後の課題

- Local Effects [Biernacki+ 2019]
- Lexical Scoped Handlers [Zhang+ 2019, Biernacki+ 2020]
- Effectの構造に依存しない型推論やCPS変換