

漸進的型付け: 安全性と柔軟性を両立するプログラミング言語

どんな研究？

ソフトウェアを記述するためのプログラミング言語の多くにはソフトウェアの誤りを検出する方法として型システムが備わっています。型システムはソフトウェアの記述方法を制限する代わりに誤りを網羅的に検出できる静的型システムと、誤りの未検出を許容する代わりに柔軟なソフトウェア記述ができる動的型システムに分類でき、本研究はこれら二種類の型システムの「いいところ取り」をする漸進的型システムというものについて研究しています。

何がわかる？

静的型システムと動的型システムの両方の長所を取り入れることで高信頼性と高生産性を両立したプログラミング言語の実現につながる事が期待されます。

また、この「静的検査と動的検査の融合」というアイデアを応用することで、モデル検査やプログラム論理といった型システム以外の検証手法をより使いやすく、より広範囲に適用できるようになることが期待できます。

研究背景

静的型検査

- ↑ 誤り検査能力
- ↓ 柔軟性

高信頼だが
変更が難しい



動的型検査

- ↑ 柔軟性
- ↓ 誤り検査能力

変更は容易だが
信頼性の担保が
難しい

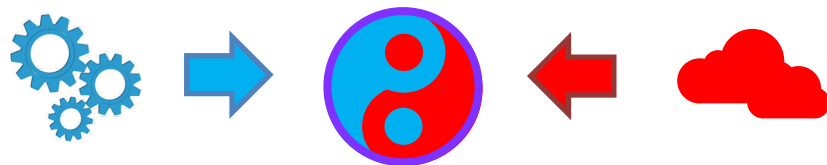


どちらを選ぼう？

- どちらを選ぶか決めるのは一般に難しい
- 開発状況とともに答も変わる

研究内容

静的型検査・動的型検査を融合した漸進的型検査



- ↑ 誤り検査能力・柔軟性
(開発者が必要に応じて自由に選択)

変更が容易で高信頼な
ソフトウェア開発が可能に

