研究スタッフ

教授：小林直樹　　准教授：住井英二郎

助教：寺内多智弘

研究目的

- **ソフトウェアが意図通りに動作すること，誤りを含んでいないことを実行前に機械的に(検証ソフトウェアによって)解析・検証**
  - 実行結果が常に正しいか？
  - 異常終了を起こさないか？
  - デッドロック状態などの危険な挙動を示さないか？
  - ▶ 検証の正しさを理論的に保証

    cf. テスト実行によるデバックでは誤りがないことは保証できない！
- **理論的な解析に基づく，プログラムの自動変換**
  - メモリ効率の良いプログラムへの変換
  - 異なるプログラミング言語で書かれたプログラムへの変換

主な研究テーマ

1. 高階関数型プログラムのモデル検査

- **世界初の高階モデル検査器 TRecS を実現!!**
  - ▶ML，OCamlなどの高レベル言語のプログラム検証が可能に!

モデル検査手法	扱えるプログラム
有限状態モデル検査	単純なループだけを持つプログラム
プッシュダウンモデル検査	一階の再帰関数を持つプログラム
我々の手法	高階関数を持つ高レベルプログラム

  - ▶様々な仕様の検証が可能に!
    - ファイルやロックなどの計算資源が正しく使用されるか？
    - エラー状態に到達しうるか?
    - XML変換プログラムが仕様に合ったXML文書を出力するか？

例：次のようなプログラムが、実行時に正しくファイルにアクセスすることを自動検証

```
let rec g x =                    /* 引数xとしてファイルを取る関数gを定義  */
  if * then close(x)             /* ファイルxを閉じる                     */
  else read(x); g(x) in          /* ファイルxを読み、関数gを再帰的に呼ぶ  */
g(open[read* ;close] "foo")      /* 読み込み専用ファイルを開き関数gに渡す */
```

2. 並行プログラムのレース解析

- マルチスレッドプログラムのレースを自動的に検出！

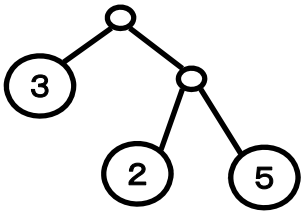
```
int c = 0;
void *f(void *) { c++; }
void main(void) {
    pthread_thread_t t;
    pthread_create(&t, NULL, &f, NULL);
    c++;
    printf("%d¥n", c);   //レースの為、出力が非決定！
}
```

- 線形計画法を用い、高速にレースを検出！！

3. XML処理プログラムの自動変換

目的：プログラマに負担をかけずにメモリ効率のよいプログラムを得られるようにしたい.

- XML処理プログラムの方法は２通り
  - 木構造処理 ： 入力データをメモリに木構造として展開してから処理



長所：プログラムがわかりやすい
短所：メモリ効率が悪い

  - ストリーム処理 ： 入力データをそのまま先頭から読みながら処理

node	3	node	2	5

長所：メモリ効率がいい
短所：プログラムがわかりにくい

木構造処理プログラムからストリーム処理プログラムへ自動変換することで、両方の長所を生かせる！！