# 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　　助教：西澤弘毅

# 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

    c.f. 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. 並行プログラムの通信，同期の解析

検証する性質：**デッドロック，ライブロック**

- ▶ 成功すべき通信が成功する前にプログラムが終了しないか？
- ▶ 成功すべき通信がいずれ成功するか？

  e.g. サーバーへ出した通信がいずれ受理されるか？

対象言語：Concurrent ML，Java などの並行プログラム言語

通信チャネルcから xを受け取り，通信チャネルdへ x+1を送信するプロセス

2つのプロセスを並行に実行

通信チャネルcへ値１を送信し，通信チャネルdから値を受け取りyに代入，その値yをプリントサーバーへ送信するプロセス

○ c?(x).d!(x+1) | c!(1).d?(y).print!(y)

× c?(x).d!(x+1) | d?(y).c!(1).print!(y)

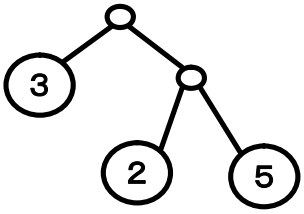
デッドロックしてしまう！

- ■ ×のようなプログラムを機械的に検出して実行を未然に防ぐ

## 2. 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！！

## 3. 計算資源の使用法解析

検証する性質：計算資源(ファイル, ロック等)が, その仕様に従って使われているか？

対象言語　　：ML, OCamlなどの関数型言語, Java仮想機械言語

```
let x = new[read* ;close]( )  in  /* 読み込み専用ファイルの生成 */
let y = new[write*;close]( )  in  /* 書き込み専用ファイルの生成 */
try
    if read(x) then write(y)  /* 読み込みが成功したらyに書き込み */
               else raise     /* 読み込みが失敗したら例外を発生   */
with
   close(x);close(y)          /* 例外発生時にはファイルを閉じる */
```

- 上記のプログラムは仕様通りにファイルにアクセスしていない
  - ファイル x, y が closeされない可能性がある(ファイルの仕様に違反)
- このようなプログラムを実行前に機械的に検出し，計算資源の誤った使用を防ぐ