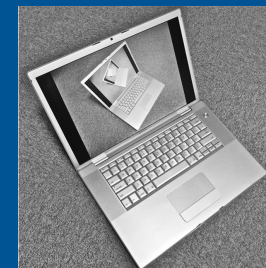
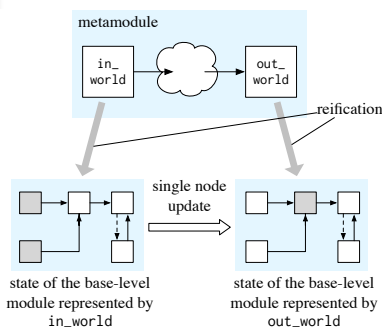
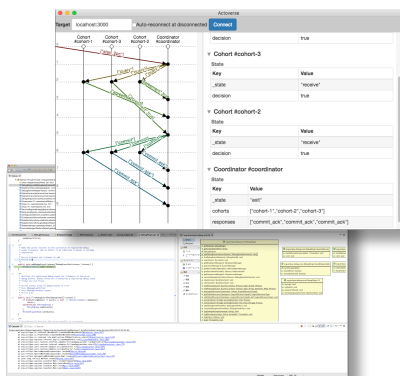
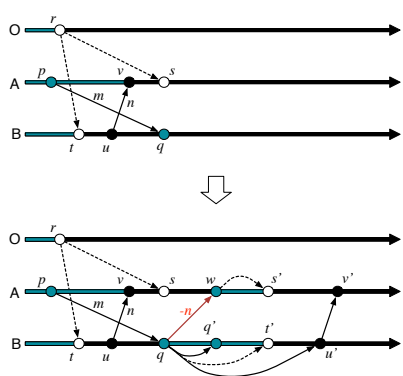


# メタプログラミングと関数プログラミング



コンピュータのプログラムは記号の列であり、それ自身プログラムで扱うデータとなり得ます。プログラムを処理対象とするプログラム、つまりプログラムの生成・変換・実行・解析・検証などを行うプログラムをメタプログラムと呼びます。さらに、プログラム中でそのプログラム自身やその実行機構に関する処理を行う特別なメタプログラムも可能です。これは我々が自分自身の身体や動作あるいは思考について考えることと似ています。そのような、自己の構造や動作を観察し変更するような計算を自己反映計算（リフレクション）と呼びます。この考え方を応用することで、ソフトウェアの拡張性や適応性、柔軟性を飛躍的に向上させることができます。さらに、自己の動作を監視する機能をソフトウェアに与えることで、安全性や信頼性を高めることもできます。渡部研究室はメタプログラミングや自己反映計算研究のパイオニアのひとつであり、基礎から応用にかけて様々な研究を行っています。

プログラムを数学的な関数として記述しようという考え方を関数プログラミングと呼びます。この考え方にもとづいて書かれたプログラムは、参照透過性をはじめとする良い性質を持ち、数学的・形式的な議論の対象にしやすいことから、関数プログラミングはプログラミングおよびプログラミング言語に関する研究の基盤となっています。渡部研究室では、関数プログラミングの発展形のひとつである関数リアクティブプログラミングの考え方をういた、組み込みシステム向けの新しいプログラミング言語の研究・開発を行っています。これにより、副作用のない式を書くだけでさまざまな組み込みシステムの動作を容易に記述できるようになります。その他、渡部研究室では、可視化によるプログラム理解支援や並行プログラムのデバッグ支援、様々なヒューマンエラーに対する形式手法にもとづく頑健性の分析手法の確立などの研究を行っています。



```
Theorem message_persistence:
  ∀ c c' l m (n: nat),
  n == count_mem m
  (in_flight_messages c) →
  c ~(-1) ~ c' →
  if l == Receive (to m) (from m)
    (content m) then
  count_mem m
  (in_flight_messages c')
  == n-1
  else if l == Send (from m)
    (to m)
    (content m)
  then
  count_mem m
  (inflight_messages c')
  == n+1
  else
  count_mem m
  (in_flight_messagec c') == n.
```



情報理工学院・情報工学系  
**渡部研究室** (西8号館E棟8F)  
 Programming Systems Group, Tokyo Tech.



[www.psg.c.titech.ac.jp](http://www.psg.c.titech.ac.jp)