

やさしい仕様記述による通信プログラム自動生成系の開発

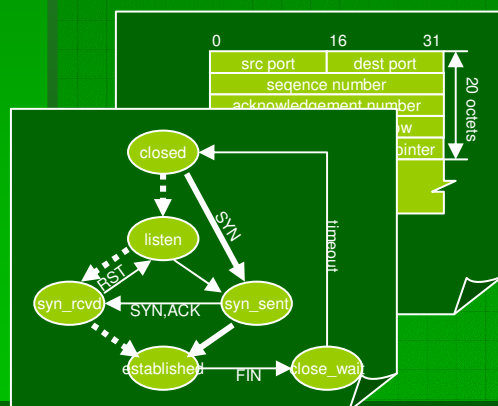
服部健太, 数馬洋一 (株式会社 システム計画研究所)

Protocol code generator

based on **R**egular **E**xpressions and **CCS**

簡単な仕様記述から
通信プロトコルの処理を行うC言語のコードを
自動的に生成する通信プロトコルコンパイラ
Preccsを開発

複雑な通信プロトコルの仕様も



簡単に記述

```
/* オプション形式の定義 */
Option ::= (
  tag : octet,
  len : octet,
  data : octet[len]
);

/* メインプロセス定義 */
Main() ::= Init();
/* 応答待ちプロセス定義 */
WaitReply(sPkt:DhcpReq) ::=
  timer!10 -> Init()
| sockin?msg:octet* ->
  ( match msg with
    | DhcpAck -> Bound(sPkt)
    | DhcpNak -> Init()
    | _ -> Error(msg)
    , Init()
  );
```

Preccs
コンパイラ

プログラムも自動生成

```
/* extended NFA */
nfa_t nfa_table[] = {
  /* [0] */ { NFA_MATCH, 2, 0, 1, -1 },
  /* [1] */ { NFA_EPSLN, 0, 0, 2, -1 },
  /* [2] */ { NFA_GETVS, 0, 0, 7, -1 },
  /* [3] */ { NFA_PSHVS, 0, 0, 0, -1 },
  /* static data */
  char_data_8CF9F8[] = "x01";
  value blk_1_val_8cf9f8 = { { VTYPE_BLK, 0 }, 1, _data_8CF9F8 };
  char_data_8CFA20[] = "x01";
  value blk_1_val_8cfa20 = { { VTYPE_BLK, 0 }, 1, _data_8CFA20 };
  char_data_8CFA48[] = "x06";
  /* state machine */
  void disp_proc(proc_t *proc) {
    next:
    switch (proc->state) {
      case 0: /* InitProc */
        {
          proc_t *np = make_proc(1, 0);
          proc_setrdy(np, 1);
          free_proc(proc); break;
        }
      case 1: /* Main */
        {
          proc_t *np = make_proc(2, 1);
          proc_setrdy(np, 2);
          free_proc(proc); break;
        }
      case 2: /* DhcpInit */
        {
          proc->data[0] = (void *)&_val_8cfaa0;
        }
    }
  }
```

Preccsを使えば...

- プログラマを苦役から解放
- 実装時のつまらないバグを撲滅
- 生産性の向上

Preccsを用いてDHCPクライアントを実装した結果
Cと比較して記述量は約3分の1に削減でき
実行時間の増加は約25%に収まった

公開ページ <http://preccs.isp.jp>