

2018/08/27

佐藤周行

プログラミング言語処理系論 (2018) 講評

1. 表彰

佐藤の成績のつけ方はとても甘いのですが、それではインセンティブが働かないこともあり、優秀なレポートを提出した人を講評時に表彰することで差をつけています。今回は、**value numbering** でプログラムをきちんと書いた 4 人と、**ECMAScript** の仕様にトライした 1 人に優秀賞を進呈します。

優秀賞： 井原央翔君 宮坂幸雄君 渋谷陽人君 中村朋生君 樗木悠亮君

履歴書に書いて威張れるかどうかはよくわかりませんが、適当にご自慢ください。

2. 全体講評と解説

今回の授業レポートは、解答に偏りがみられました。問題と解答数の表を下にあげておきます。

問題の種類	(問題)	解答
言語の仕様、実現	1-7	1 1人 1' 1人 7 1人
VM の動作	8-10	9 7人
最適化	11-14	11 8人 13 1人

問題 9 (論文をまとめる) と問題 11 (**value numbering**) が圧倒的に人気でしたが、どうい理由なのか解析しきれいていません。問題 9 の対象論文は 2013 年のもので、少々古かったのは申し訳なく思っています。

言語の仕様、実現

この講義では、言語仕様を読み込むことが最初のテーマになるわけですが、従来の読みやすい **Java** や **Ruby** に加えて問題 1' で **Ecmascript** の言語仕様を読み込むことを課題として出しました。一度読んでみればわかるのですが、「まだ整理しきれいていない」「ステークホルダーが多すぎて、にっちもさっちもいかない」と言語仕様はこんな惨状を呈するという典型かもしれません。トライした人がいたのは称賛に値します。鬼のような改訂が続いていますから引続きウォッチするとよいと思います。問題 3 は、**LALR** の復習問題、問題 4 は措いておいて、問題 5 は **Perl** の **source tree** の解析です。**Perl**、言語としての寿命を

迎えつつありますね。若い人は知らなくてよろしい。

問題 7 で、言語デザインに挑戦した人がいました。コード生成のところで JVM を使う仕様にしたので **calling convention** の設計の経験が積み重なったようですが、その代わり、タイプチェックのルーチンをきちんと書いてきました。できることからでよいので、このような中くらいのプロジェクトを経験しておくとういと思います。

VM の動作

VM というとは JVM が一昔前は標準で、それは今でもそうなのですが、Python がのってきて、比較検討しやすくなりました。問題 8 はその意味で誰かトライしてほしいのですが… 問題 9 は動的言語を JVM 上に実現して大丈夫か、という観察です。結果はそれほどポジティブではありません。今もそうです（私の観察が古びていなければ）。問題 8 あたりをこなしたうえで **vmgen** あたりでさくさくっと VM が書ける（問題 10）ようになるとよいと思います。

最適化

最適化は問題 11 の基本ブロック内のものと、問題 12 以降のいわゆる大域的な最適化に分類され、後者は言語の意味論の観点からも興味深いものです。問題 11、12 は、プログラムを書けるかどうかのポイントです。アルゴリズムを設計したらプログラムを書きましょう。問題 13 は、SSA を知っている問題 11 の拡張 (**global value numbering**) として解けます。SSA、説明しなかったのは申し訳ありませんでした。問題 14 は GCC の最適化ルーチンのソースを読めという問題でした。「教科書（理論）通り」にコードが書かれていることを観察できます。