プログラムの意味論演習システム

京都大学 大学院情報学研究科
五十嵐 淳

体験アカウント 絶賛配布中!!!

主な特徴

Web ベース

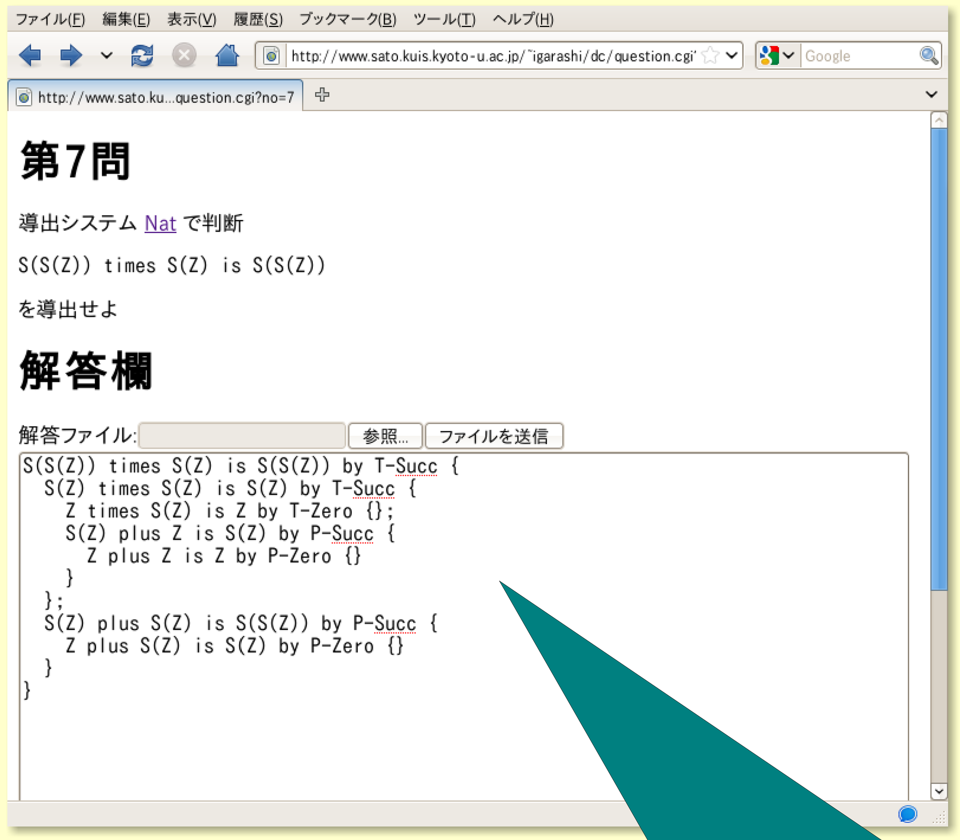
型判断・評価/簡約関係などの**導出の記述演習**

形式**体系記述**用の**専用言語**

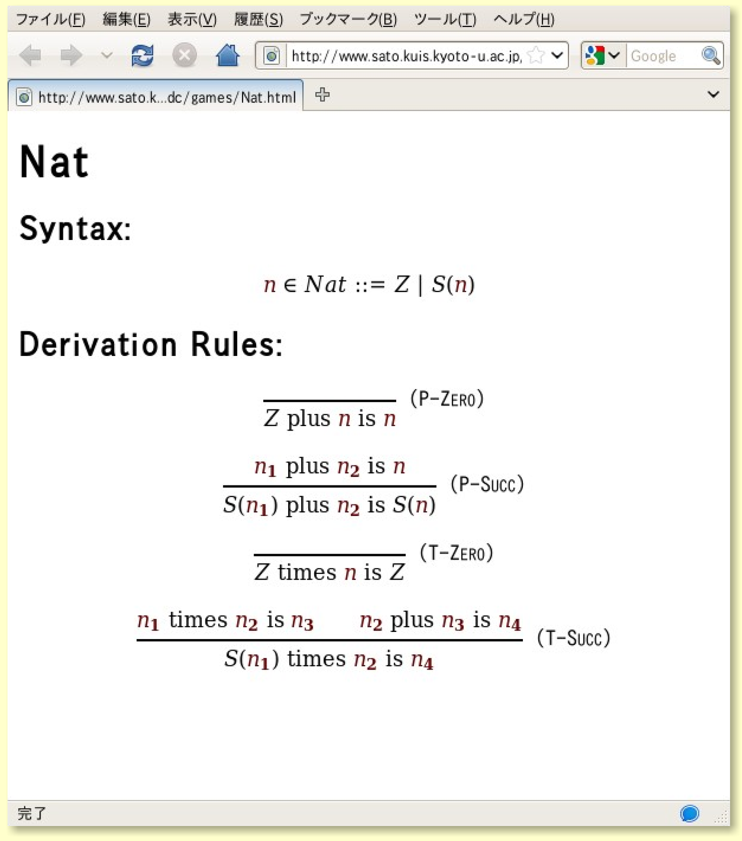
導出検査(および生成)器の**自動生成**

演習システム外観

問題と解答フォーム



体系定義



$$\dfrac{Z * S(Z) = Z \qquad \dfrac{Z + Z = Z}{S(Z) + Z = S(Z)}}{S(Z) * S(Z) = S(Z)} \qquad \dfrac{Z + S(Z) = S(Z)}{S(Z) + S(Z) = S(S(Z))}$$
$$S(S(Z)) * S(Z) = S(S(Z))$$

講義(M1向け)内容:

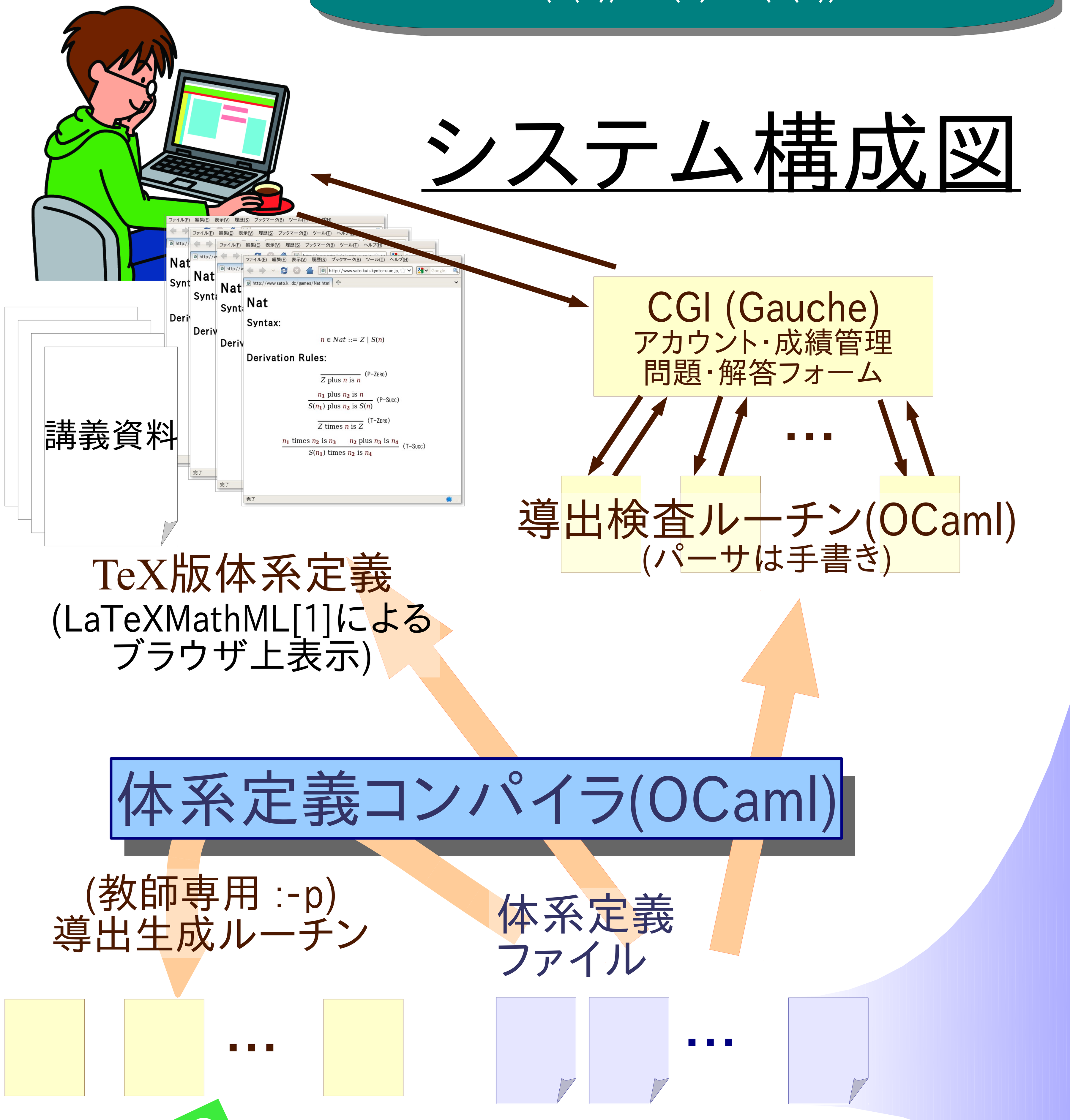
- 自然数の加算・乗算・比較
- コアMLの評価意味論
  - 整数、真偽値、let、(再帰)関数、リスト、パターンマッチング
  - 継続、参照
- de Bruijn index表記への変換
- コアMLの単相・多相型システム
- 型推論アルゴリズム
- 型健全性などのメタ定理

}全て演習システムに実装済

2009年度の戦績

- 全150問出題
- 履修者29名(うち合格者22名)
- 全問解答 2名、100問以上解答 17名
- 解答用プログラムを書いた学生 2名
  - ただし全問正解はしていない ;-)

システム構成図



TODO

- Eliom による Ocsigen web server [2] への組込み
- より伝統的な形式システム定義記法(c.f. Ott [3])
- マルチバイト文字サポート

[1] D. R. Woodall. LaTeXMathML: translating LaTeX math notation dynamically to Presentation MathML.
`http://www.maths.nottingham.ac.uk/personal/drw/lm.html`
[2] V. Balat, J. Vouillon, B. Yakobowski. Experience Report: Ocsigen, a Web Programming Framework. In *Proc. of ICFP2009.*
[3] P. Swell, F. Z. Nardelli, S. Owens, G. Peskine. Ott: Effective Tool Support for the Working Semanticist. In *Proc. of ICFP2007.*

形式体系の定義例: EvalML1 (単純な式の評価)

```
[Syntax]
i in int
b in bool
v in Value ::= i | b
e in Exp ::= i | b | BinOp(op, e, e) | If(e, e, e)
op in Prim ::= Plus | Minus | Mult | Lt

[Judgments]
EvalTo(e; v);
AppBOp(op, v, v; v);

[Rules]
E-Int:   EvalTo(i, i) :- ;
E-Bool:  Eva...
E-IfT:  ...
E-IfF:  ...
E-Plus:
   EvalTo(BinOp(Plus, e1, e2), i3) :-
      EvalTo(e1, i1), EvalTo(e2, i2),
      AppBOp(Plus, i1, i2, i3);
...
B-Plus:
   AppBOp(Plus, i1, i2, i3) :- `$i3 = $i1 + $i2`;
...
```

OCamlの型

抽象構文定義

$e \Downarrow v$

判断形式定義

「ここの前の引数が決まれば導出が生成できる」

(Prolog風) 推論規則定義

$$\dfrac{e_1 \Downarrow i_1 \qquad e_2 \Downarrow i_2 \qquad i_1 \text{ plus } i_2 \text{ is } i_3}{e_1 + e_2 \Downarrow i_3}\ \text{(E-Plus)}$$

規則の“型検査”

OCamlによる付帯条件記述

タルスキ曰く「『i1とi2の和がi3である』のは I1とI2の和がi3である時である」(やや嘘)