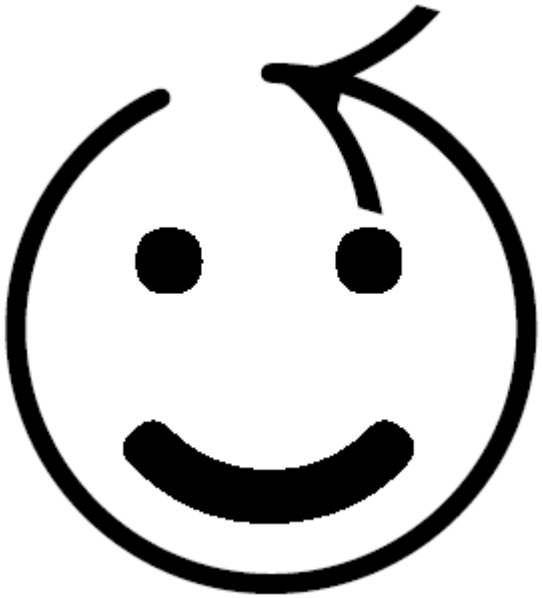
自動定理証明の紹介

～Proof Summit～ 2011-09-25

酒井 政裕

自己紹介

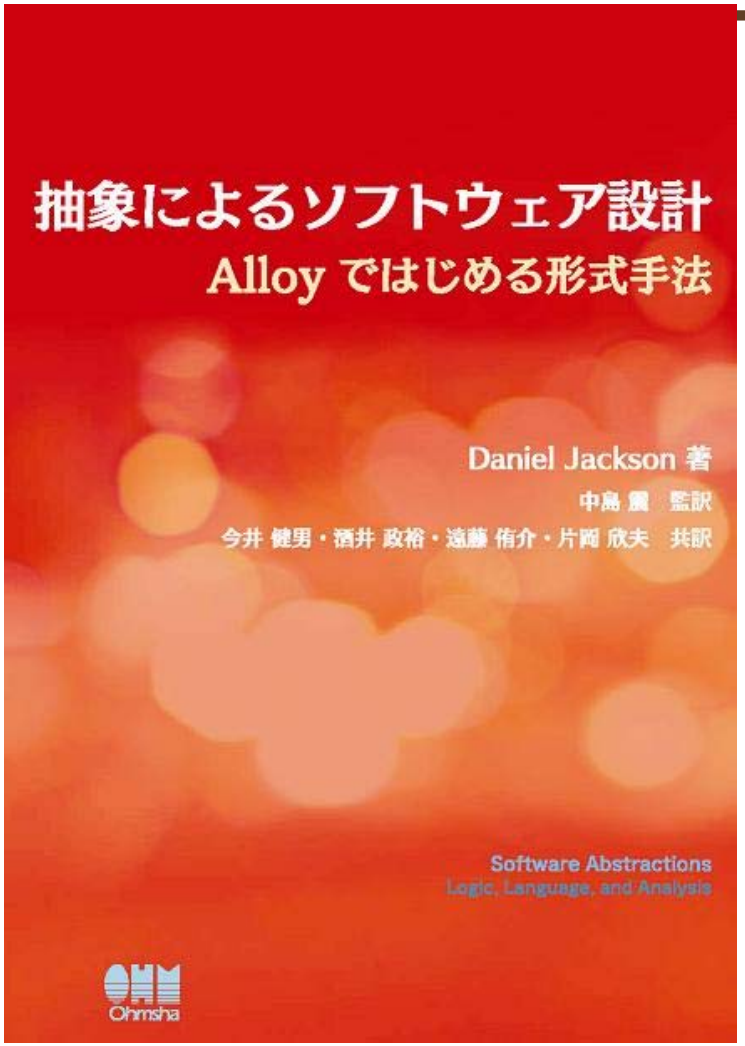
酒井 政裕



- Blog: ヒビルテ
- Twitter: @masahiro_sakai

- Haskeller
- Agda: Agda1のころ遊んでた
- Coq: 最近入門
- 最近、Alloyの本を翻訳

Alloy本



『抽象によるソフトウェア設計
Alloyではじめる形式手法』

- オーム社より
好評発売中！

このLTの趣旨

CoqとかAgdaの話ばかりでツマラン

ので、息抜きにちょっとは違う話を

話したかったこと

- 自動定理証明
  - –Automated Theorem Proving
- モデル発見
- 色々な Decision procedure
- SAT/SMT

CoqやAgdaへのイチャモン

- 自分が考えた証明を形式化し、
  正しさを確認したり、
  リファクタリングしたりには便利

- だけど、何かを証明したいときに、
  本当に支援になるのか?

CoqやAgdaへのイチャモン

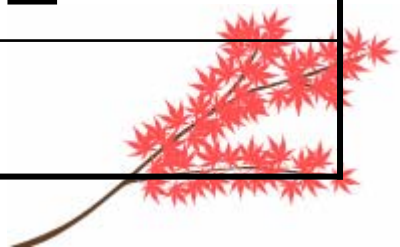
そもそも
機械に分からせるために、
細部まで証明を書くなど

機械の奴隷

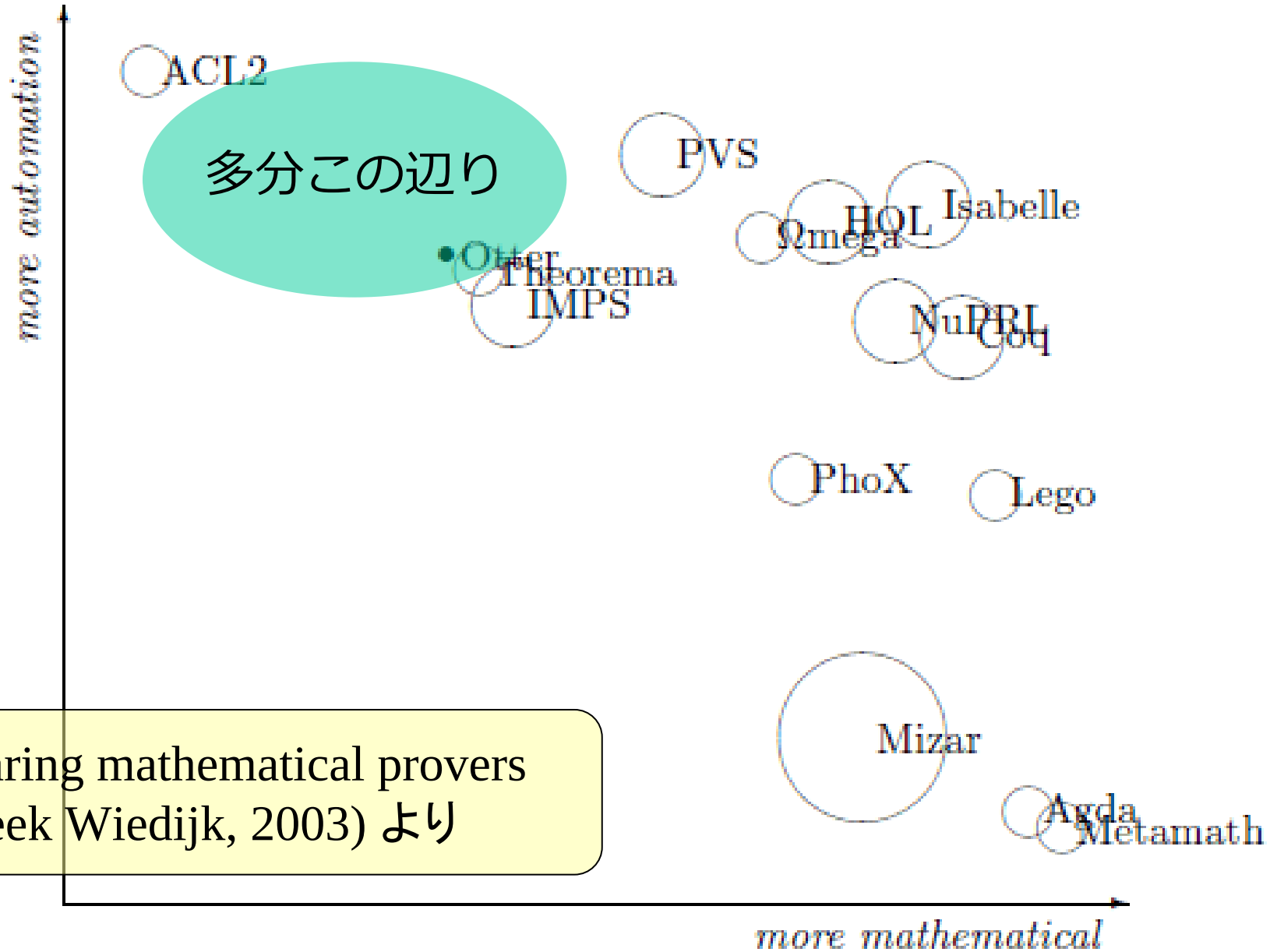
では?

対話的定理証明 vs 自動定理証明

	対話的定理証明	自動定理証明
ツール	Coq, Agda, …	E, SPASS, Otter, …
自動化	人間が証明を書き、それをツールが検査。	ツールが証明を探索。人間はそれをガイド
古典論理?	直観主義論理	古典論理
高階/一階	高階	一階



6 A rather subjective two-dimensional diagram



Comparing mathematical provers
(by Freek Wiedijk, 2003) より

- 数学的な表現力は弱い
- が、その分自動化されている！

自動定理証明器

- Otter
  - 自動定理証明器が有名になるきっかけになった証明器
- SPASS
  - 様相論理などもサポート
- E
  - The E Equational Theorem Prover (eprover)
- 他にも沢山
  - Prover9, Vampire, Waldmeister, …

簡単な例: ソクラテスは死ぬ (SPASS)

```
begin_problem(Sokrates1).

list_of_descriptions.
name({*Sokrates*}).
author({*Christoph Weidenbach*}).
status(unsatisfiable).
description({* Sokrates is mortal and
    since all humans are mortal, he is
    mortal too. *}).
end_of_list.

list_of_symbols.
  functions[(sokrates,0)].
  predicates[(Human,1),(Mortal,1)].
end_of_list.

list_of_formulae(axioms).
formula(Human(sokrates)).
formula(forall([x],implies(Human(x),M
    ortal(x)))).
end_of_list.

list_of_formulae(conjectures).
formula(Mortal(sokrates)).
end_of_list.

end_problem.
```

ヘッダ

公理

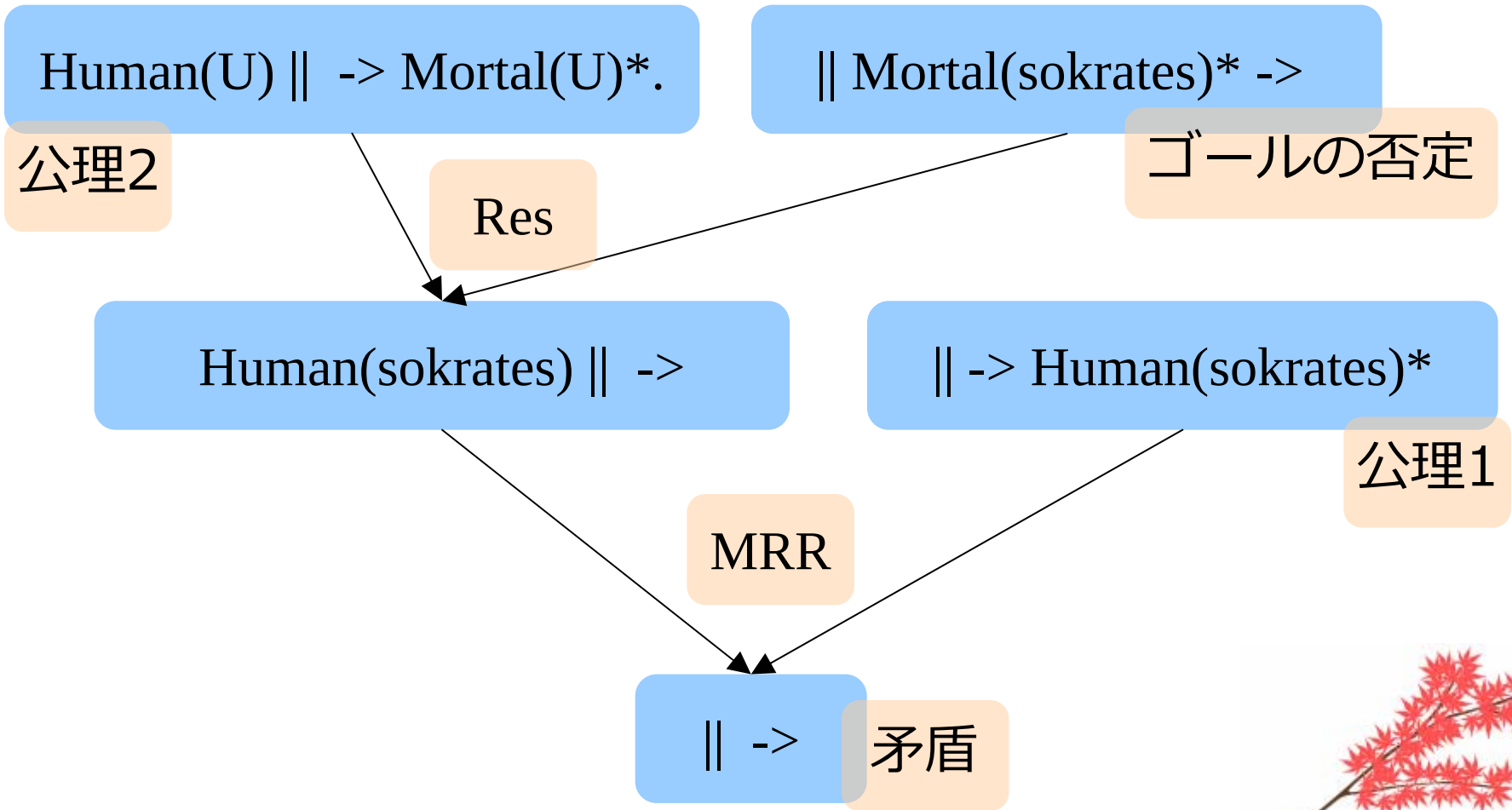
証明したいゴール

関数・述語とアリティの宣言

簡単な例: ソクラテスは死ぬ (SPASS)

```
% SPASS -DocProof Socrate.dfg
(中略)
Here is a proof with depth 1, length 5 :
1[0:Inp] ||  -> Human(sokrates)*.
2[0:Inp] || Mortal(sokrates)* -> .
3[0:Inp] Human(U) ||  -> Mortal(U)*.
4[0:Res:3.1,2.0] Human(sokrates) ||  -> .
5[0:MRR:4.0,1.0] ||  -> .
Formulae used in the proof : axiom0
   conjecture0 axiom1
```

簡単な例: ソクラテスは死ぬ (SPASS)



表現力の弱さ

- 有限個の公理のみ & 一階
- たとえば、数学的帰納法は公理化不能:

$$P(n) \land (\forall n.\ P(n) \rightarrow P(s(n))) \rightarrow \forall n.\ P(n)$$

- 各述語Pに対して公理がひとつ必要
- 高階なら ∀P：N→Prop. ～ として、
  単一の公理で公理化できる

例: 自然数の加算の結合性 (E / TPTP)

```
fof( plus_z, axiom, ![X] : plus(X,z)=X ).
fof( plus_s, axiom, ![X,Y] : plus(X,s(Y))=s(plus(X,Y)) ).

fof( plusAssocP_def, axiom, ![X,Y,Z] : (
plusAssocP(X,Y,Z) <=>
    (plus(plus(X,Y),Z)=plus(X,plus(Y,Z)))
) ).
fof( plusAssocP_ind, axiom, ![X,Y] : (
(plusAssocP(X,Y,z) &
  (![Z] : plusAssocP(X,Y,Z) => plusAssocP(X,Y,s(Z))))
  => ![Z] : plusAssocP(X,Y,Z)
) ).

fof( plus_assoc, conjecture, ![X,Y,Z] : plus(plus(X,Y),Z) =
    plus(X,plus(Y,Z)) ).
```

! は ∀

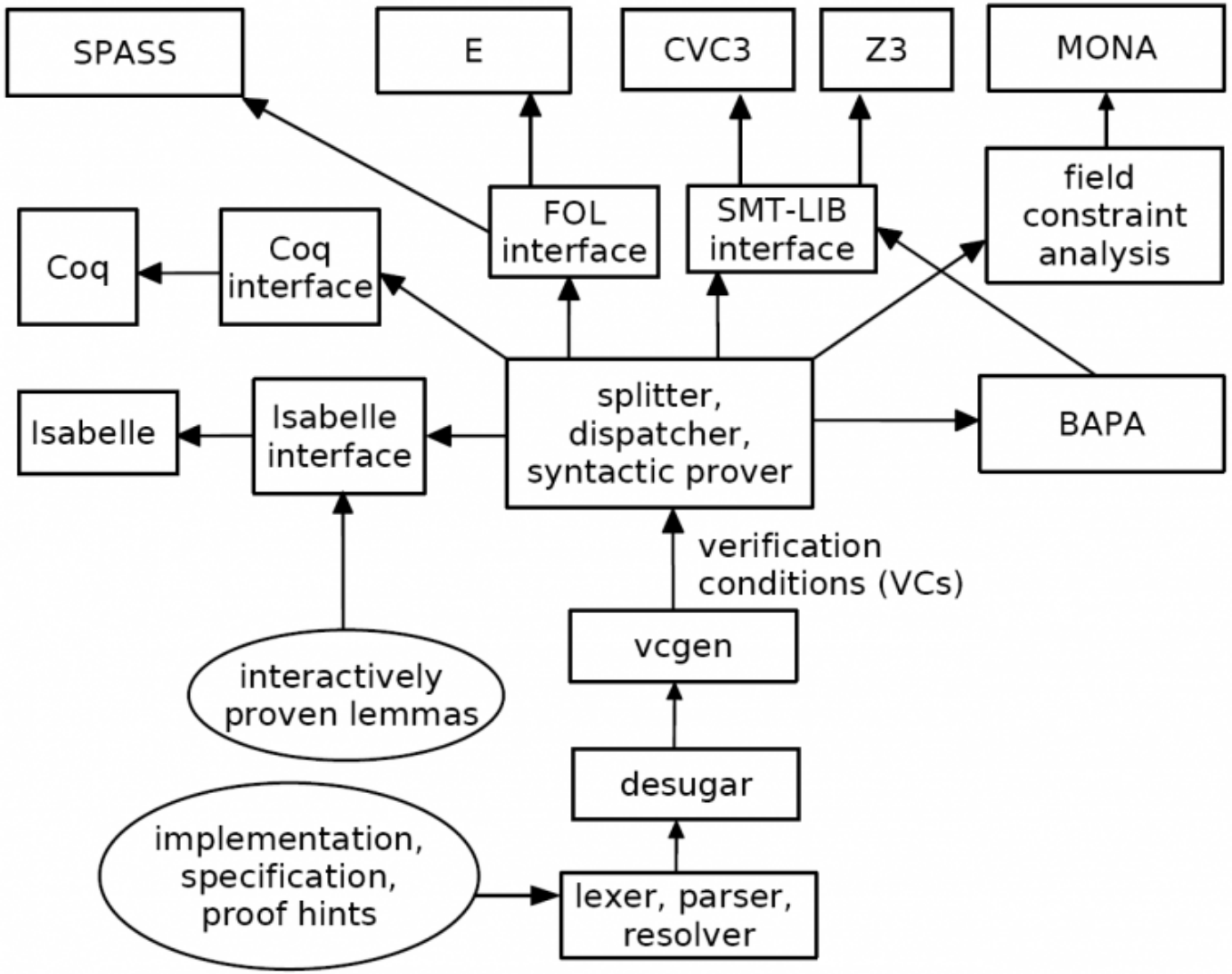
帰納法の
インスタンスを
明示的に公理に

```
eprover -l 2 --tptp3-format nat1.tptp
```

TPTP

- Thousands of Problems for Theorem Provers
  - http://www.cs.miami.edu/~tptp/
- ATP向けの諸々を提供
  - 問題セット
  - 入力ファイルフォーマット
  - 各種ユーティリティ
    - 証明を見やすく表示したり、他のツール用に変換したりといったツールもあった気がするが、忘れた
- SATLIB, SMTLIB, MIPLIB等のATP版

応用: プログラム検証Jahob



チャレンジ
Seven Trees 問題

- 関数記号: +, ×, T
- 公理: 半環の公理 + $\{T = 1 + T^2\}$
- ゴール: $T = T^7$

- あなたの使っている定理証明系は、
  こういうのを解くのを支援してくれる?

自動定理証明器に解かせてみた (1)

```
fof( coprod_comm, axiom, ![X,Y] : coprod(X,Y) =
    coprod(Y,X) ).
fof( coprod_assoc, axiom, ![X,Y,Z] : coprod(X,coprod(Y,Z))
    = coprod(coprod(X,Y),Z) ).
fof( prod_comm, axiom, ![X,Y] : prod(X,Y) = prod(Y,X) ).
fof( prod_assoc, axiom, ![X,Y,Z] : prod(X,prod(Y,Z)) =
    prod(prod(X,Y),Z) ).
fof( dist, axiom, ![X,Y,Z] : prod(X,coprod(Y,Z)) =
    coprod(prod(X,Y),prod(X,Z)) ).
fof( prod_unit, axiom, ![X] : prod(one, X) = X ).
fof( t, axiom, t = coprod(one, prod(t, t)) ).

fof( seven_trees, conjecture, t =
    prod(t,prod(t,prod(t,prod(t,prod(t,prod(t,t)))))) ).
```

爆発

自動定理証明器に解かせてみた (2)

```
fof( comm, axiom, ![X,Y] : coprod(X,Y) = coprod(Y,X) ).
fof( assoc, axiom, ![X,Y,Z] : coprod(X,coprod(Y,Z)) =
    coprod(coprod(X,Y),Z) ).
fof( t1, axiom, t1 = coprod(t0, t2) ).
fof( t2, axiom, t2 = coprod(t1, t3) ).
fof( t3, axiom, t3 = coprod(t2, t4) ).
fof( t4, axiom, t4 = coprod(t3, t5) ).
fof( t5, axiom, t5 = coprod(t4, t6) ).
fof( t6, axiom, t6 = coprod(t5, t7) ).
fof( t7, axiom, t7 = coprod(t6, t8) ).
fof( t8, axiom, t8 = coprod(t7, t9) ).

fof( seven_trees, conjecture, t1 = t7 ).
```

解けた

チャレンジ: Otterが有名になった例

- Two inverter problem
  - 3入力3出力の組み合わせ回路で、各出力が対応する入力の否定になっているものを作れ
  - ただし、andとorは何個使ってもよいが、notは2個しか使えない
- OtterはPrologのメタ述語のような機能を持っていて、それを利用して解いた
  - (今の一般的な定理証明系では使えない)
- あなたの使っている定理証明系は、こういうのを解くのを支援してくれる?

CASC

- The CADE ATP System Competition
  - http://www.cs.miami.edu/~tptp/CASC/
  - CADE(the Conference on Automated Deduction)で毎年やってるコンペ
- CASC-23 (2011) の勝者:

THF: Satallax 2.1	CNF: E 1.4pre
TFA: SPASS+T 2.2.14	EPR: iProver 0.9
FOF: Vampire 0.6	UEQ: Waldmeister 710
FNT: Paradox 3.0	LTB: Vampire-LTB 1.8