Using external provers in proof assistants

An example of small scale inter-operability

Frédéric Blanqui

Deduc⊢eam

Why using external provers?

- relieve proof assistant users
  - improve productivity
  - may increase expressive power
- relieve proof assistant developers
  - theorem-proving/termination/confluence are undecidable
  - there exist many different techniques and specialized tools

argument against: safety?

solution: certify results

Well known examples

- computer algebra systems
  - Maple (Isabelle/HOL 1995, PVS 2001, Coq 2005)
  - not certified
- SAT/SMT solvers
  - trace verification (Coq 2011, Isabelle/HOL 2011)
  - reflexive tactics (Coq 2008, Isabelle/HOL 2010)
- automated theorem provers
  - Sledgehammer exports a goal to a TPTP prover and Metis reconstructs an Isabelle/HOL scripts from its output (2007)
  - Zenon[Modulo] and iProverModulo generate Coq or Dedukti files

little use in proof assistants with dependent types. . . :-(

A less known example: termination and confluence provers

Since 2009:

- there is a common format for:
  - termination/confluence/complexity problems (XTC) for
    [conditional] [relative] [integer]
    [string|[first|higher-order] term] rewrite systems
    [modulo some equational theory]
  - [non] termination/confluence/complexity certificates (CPF)
- many provers now produce certificates in CPF (AProVE, TTT2, CiME3, Matchbox, CSI, mkbTT, KBCV, CaT, TcT)
- there are [certified] CPF-verifiers (CiME3, Rainbow, CeTA)

  most complete: CeTA, written in Isabelle/HOL (even parsing), proved correct and extracted to standalone Haskell program

  with AProve or TTT2: 70% of TPDB (1522/2132 TRSs)

  see René Thiemann's talk

CPF

the CPF grammar is described by an XML Schema file

a termination certificate is a tree (XML file) where:

- each node is the application of a clearly identified theorem T:
  R terminates if $R_1, \ldots, R_n$ terminate and $C_T(R, R_1, \ldots)$ holds
- subtrees are certificates for the termination of $R_1, \ldots, R_n$

a certificate is valid if, at each node T, $C_T(R, R_1, \ldots)$ holds

CPF example

- theorem $MN(R, I)$: R terminates if I is a monotone polynomial interpretation on $\mathbb{N}$ such that $R \subseteq \geq_I$ and $R \setminus >_I$ terminates
- R can be proved terminating by finding a sequence of polynomial interpretations $I_1, \ldots, I_n$ such that each interpretation removes some rules until none remains

$$R = R_0, \quad R_1 = R_0 \setminus >_{I_1}, \quad \ldots, \quad R_n = R_{n-1} \setminus >_{I_n} = \emptyset$$

- in this case, the certificate is the tree $\mathcal{T}(MN(R_0, I_1), \mathcal{T}(MN(R_1, I_2), \ldots, \mathcal{T}(MN(R_{n-1}, I_n), ()) \ldots))$
- it is valid if, for every k, I_k is a monotone polynomial interpretation on $\mathbb{N}$, $R_{k-1} \subseteq \geq_{I_k}$, $R_k = R_{k-1} \setminus >_{I_k}$, and $R_n = \emptyset$

Conclusion

Why not using external provers
not only for propositional or first-order subgoals
but also for termination and confluence proofs?