22nd International Joint Conference on Artificial Intelligence (IJCAI'11), Barcelona, Spain, July 19-22, 2011
Best Paper Track: International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'11)

Pushing Efficient Evaluation of HEX Programs by Modular Decomposition

Thomas Eiter[1] Michael Fink[1] Giovambattista Ianni[2] Thomas Krennwallner[1] Peter Schüller[1]

In a Nutshell

- HEX-Programs
  - knowledge representation language
  - extends Answer Set Programming (ASP)
  - comprise external atoms for external knowledge/computation
- Evaluation Efficiency
  - original model calculation algorithm has several drawbacks
  - realizing applications in HEX called for efficiency improvements
- Theoretical Contributions
  - generalized program splitting theorem
    $\Rightarrow$ program fragments may overlap, branch, and join
    $\Rightarrow$ constraints shared among fragments
  - program decomposition via a directed acyclic graph
    $\Rightarrow$ nodes are program fragments, edges represent dependencies
  - sound and complete model enumeration using this graph
- Effect of Improvements
  - independent program fragments are evaluated independently
  - early elimination of models
  - considerably less redundant computations in relevant scenarios
  - parallel evaluation possible

  $\Rightarrow$ Improved time and space efficiency!
- Open Source Implementation `dlvhex` (supports `dlv` and `clingo`):
  `http://www.kr.tuwien.ac.at/research/systems/dlvhex/`

HEX

External Atom $\&g[\vec{x}](\vec{y})$

- input list $\vec{x} = x_1, \ldots, x_n$ and output list $\vec{y} = y_1, \ldots, y_m$ of terms
- Model: $I \models \&g[\vec{x}](\vec{y})$ iff oracle function $f_{\&g}(I, \vec{x}, \vec{y})$ is true
- Computational view: $I \models \&g[\vec{x}](\vec{y})$ for all $\vec{y} \in F_{\&g}(I, \vec{x})$
- External computation depends on interpretation
  $\Rightarrow$ external atoms cannot be evaluated during grounding process!

Example (choose a plan and a usage, determine resource needs):

$$\left\{\begin{array}{rl} choose(a,c,d) \leftarrow & choose(b,e,f) \leftarrow \\ plan(a) \vee plan(b) \leftarrow & \\ need(p,C) \leftarrow & \&res[plan](C) \\ use(X) \vee use(Y) \leftarrow & plan(P), choose(P,X,Y) \\ need(u,C) \leftarrow & \&res[use](C) \\ \leftarrow & need(\_, money) \end{array}\right\}$$

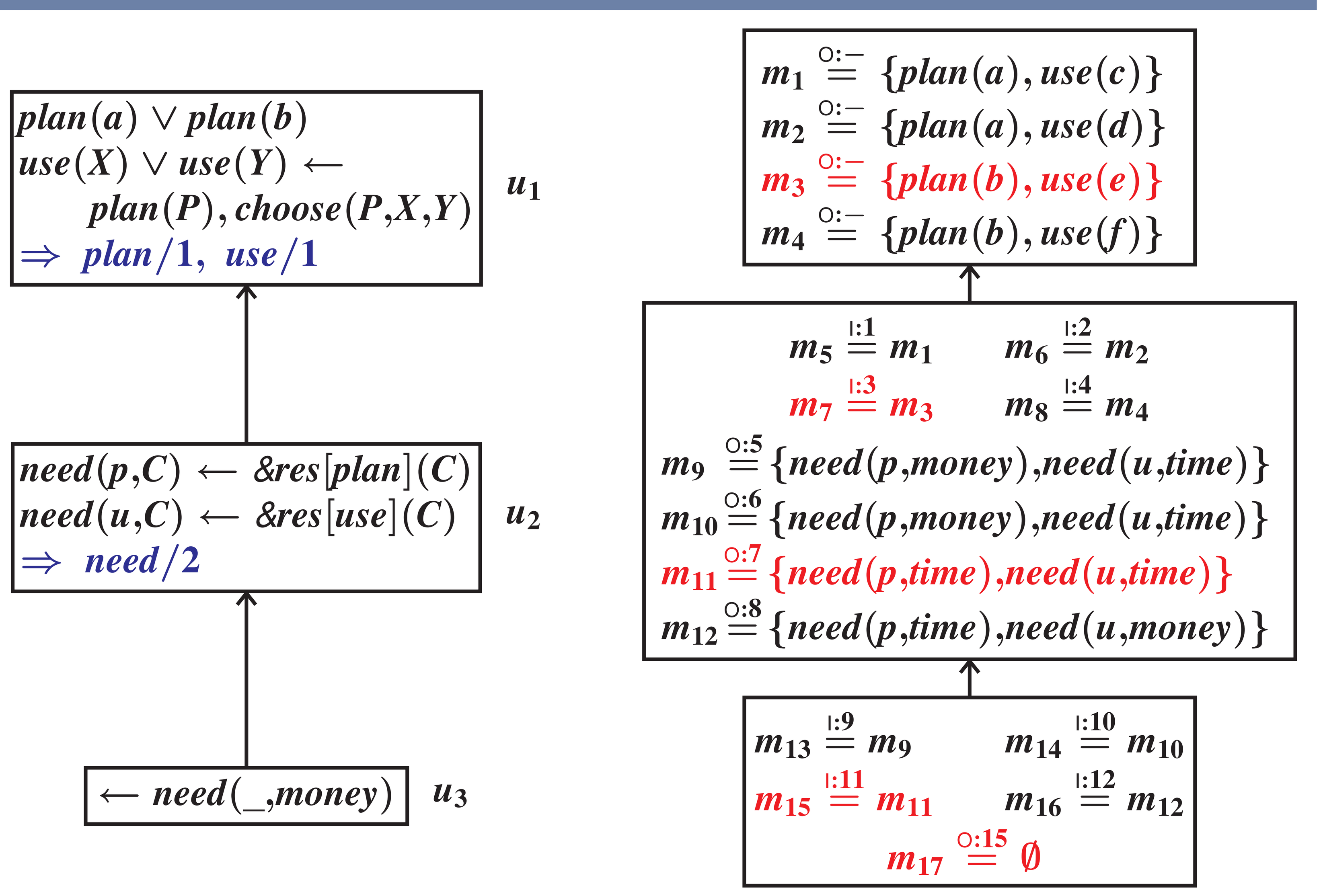
The oracle for $\&res$ is given as

$I \models \&res[p](money)$ if $p(C) \in I$ for $C \in \{a, f\}$,
$I \models \&res[p](time)$ if $p(C) \in I$ for $C \in \{b, c, d, e\}$, and
$I \not\models \&res[p](X)$ in all other cases.

The single answer set of the above HEX program is

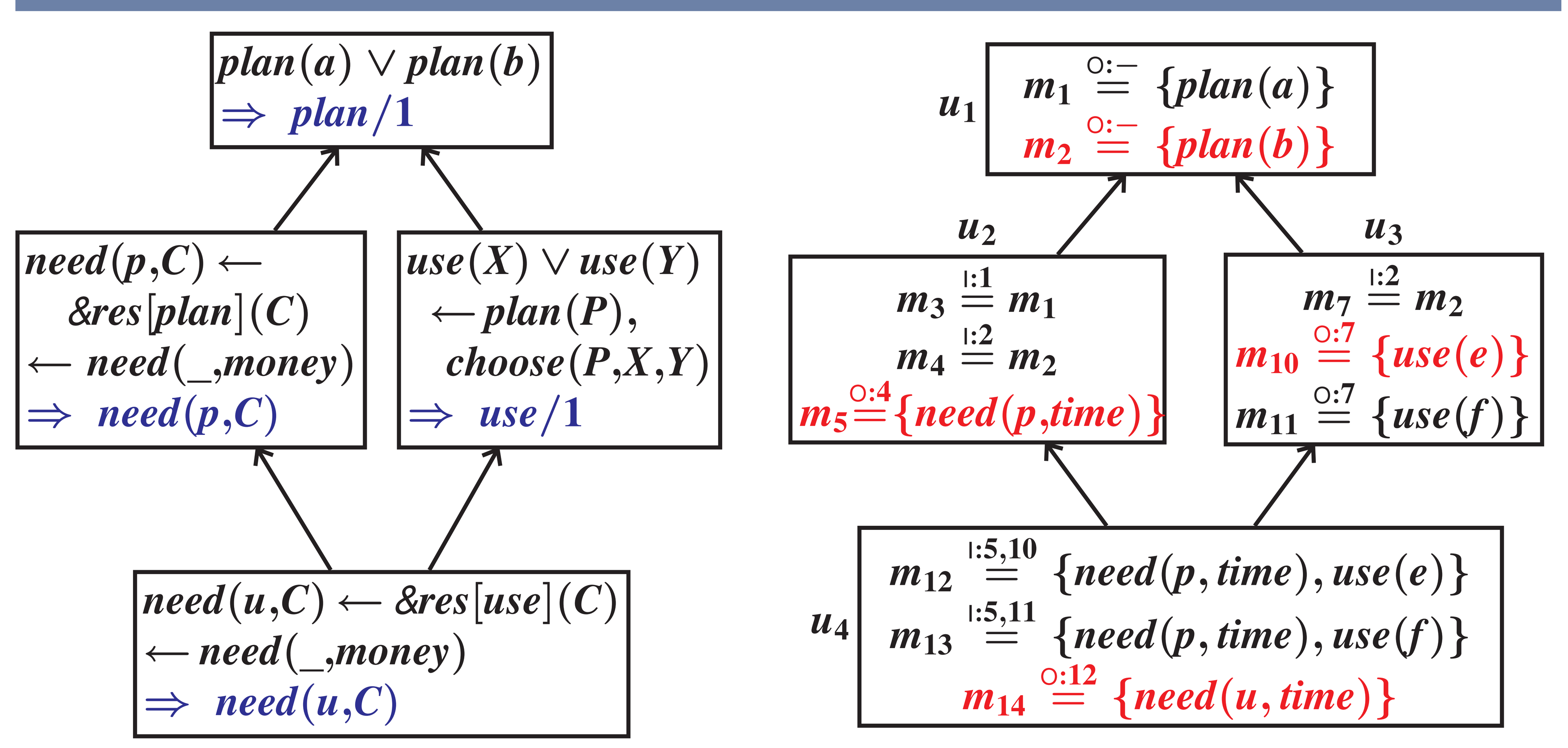
$$\left\{\begin{array}{l} plan(b), \&res[plan](time), need(p,time), \\ use(e), \&res[use](time), need(u,time) \end{array}\right\}$$

Previous Evaluation Method



- external atoms depend on output of previous program fragments
- use largest program fragments possible wrt. external calculations

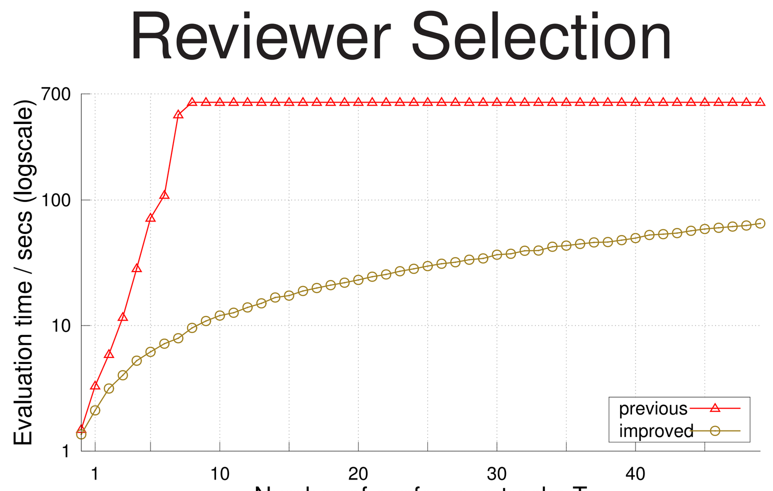
Improved Evaluation Method



- external atoms depend on output of previous program fragments
- program fragments model disjoint sets of atoms
- rule dependencies and negative constraint dependencies satisfied
- positive constraint dependencies satisfied in at least one fragment
- input models can be joined if they share a common ancestor model

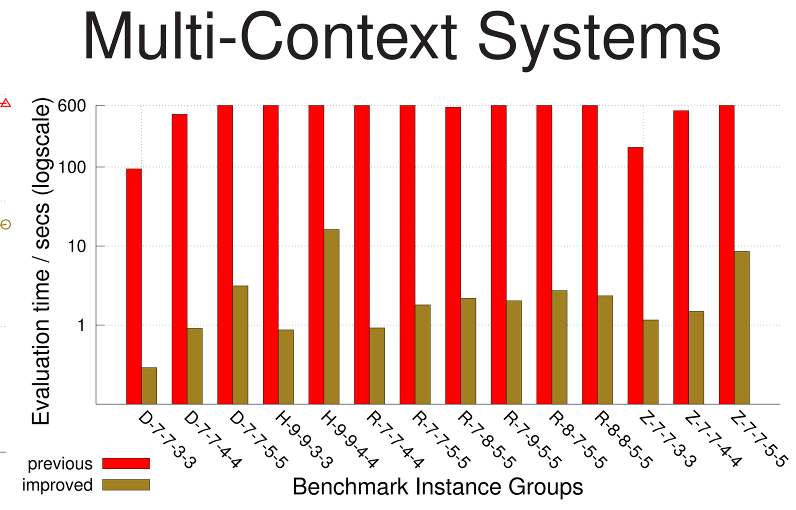
Experimental Results

Reviewer Selection



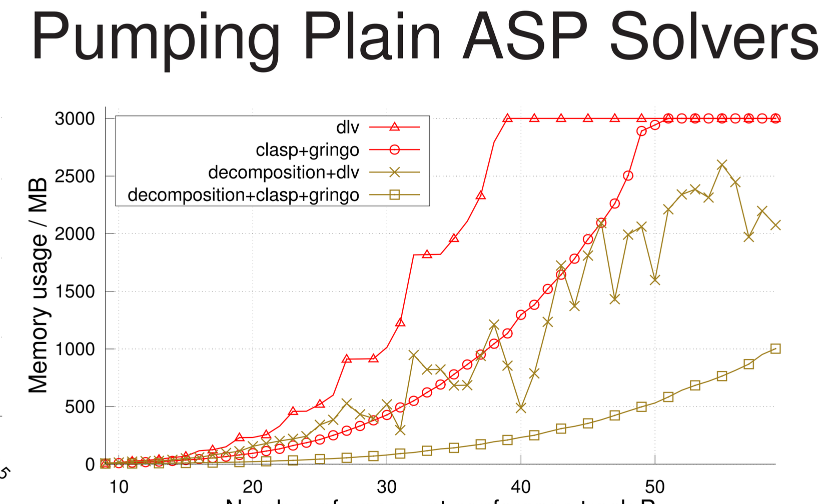
- exponential speedup for synthetic benchmark

Multi-Context Systems



- improved scalability of application! (1 sec vs. 600 sec)

Pumping Plain ASP Solvers



- even for plain ASPs without external atoms, efficiency can be improved by decomposition

References

- Thomas Eiter, Giovambattista Ianni, Roman Schindlauer, and Hans Tompits. A Uniform Integration of Higher-Order Reasoning and External Evaluations in Answer-Set Programming. In *IJCAI-05*, pages 90–96, 2005.
- Wolfgang Faber, Nicola Leone, and Gerald Pfeifer. Semantics and complexity of recursive aggregates in answer set programming. *Artificial Intelligence*, 175(1):278–298, January 2011.
- Michael Gelfond and Vladimir Lifschitz. Classical Negation in Logic Programs and Disjunctive Databases. *Next Generation Computing*, 9(3–4):365–386, 1991.

[1] Institut für Informationssysteme, Technische Universität Wien
{eiter,fink,tkren,ps}@kr.tuwien.ac.at
[2] Dipartimento di Matematica, Università della Calabria
ianni@mat.unical.it

Der Wissenschaftsfonds.
Wiener Wissenschafts-, Forschungs- und Technologiefonds