

Pushing Efficient Evaluation of HEX Programs by Modular Decomposition

Thomas Eiter Michael Fink
Thomas Krennwallner

Giovambattista Ianni
Peter Schüller



KBS Group – Institut für Informationssysteme, Technische Universität Wien
Dipartimento di Matematica, Università della Calabria

IJCAI – July 20, 2011 – Barcelona

Best Paper Track – LPNMR 2011 – Vancouver



Motivation:

- ▶ HEX extends Answer Set Programming
 - ▶ supports **external atoms** for external knowledge/computations
 - ▶ drawbacks of previous HEX evaluation algorithm
- ⇒ **applications** called for efficiency improvements



Motivation:

- ▶ HEX extends Answer Set Programming
- ▶ supports **external atoms** for external knowledge/computations
- ▶ drawbacks of previous HEX evaluation algorithm
- ⇒ **applications** called for efficiency improvements

This work:

- ▶ novel HEX **evaluation formalism**
 - ⇒ evaluation graph + model graph
- ▶ implementation + experimental evaluation
 - ⇒ can even **outperform standard ASP solvers**

Programs are sets of rules of the form

$$\alpha_1 \vee \dots \vee \alpha_k \leftarrow \beta_1, \dots, \beta_n, \text{not } \beta_{n+1}, \dots, \text{not } \beta_m$$

where *not* is **negation-as-failure** and α_i, β_j are atoms of the form

$$p(t_1, \dots, t_k).$$

Herbrand base HB_P is the set of all ground atoms using constants of P

Interpretation $I \subseteq HB_P$

$$I \models \text{rule body iff } \begin{cases} \text{positive atoms } \beta_1, \dots, \beta_n \text{ are in } I, \text{ and} \\ \text{negative atoms } \beta_{n+1}, \dots, \beta_m \text{ are not in } I \end{cases}$$

$I \models \text{rule}$ iff some head atom α_i is in I , or the rule body is not satisfied

$I \models P$ iff $I \models \text{grnd}(P)$ iff I satisfies all ground rules of program P

FLP **reduct** [Faber *et al.*, 2011] of P wrt. I :

fP^I is the set of ground rules of P where I satisfies the rule body

I is an answer set of P iff I is a \subseteq -minimal model of fP^I

Program (choose a plan, choose a usage, determine resource needs):

$$\begin{array}{l}
 \text{EDB} \quad \{ \text{choose}(a, c, d) \leftarrow, \text{choose}(b, e, f) \leftarrow \} \\
 \text{IDB} \quad \left\{ \begin{array}{l}
 r_1: \text{plan}(a) \vee \text{plan}(b) \leftarrow \\
 r_2: \quad \quad \text{need}(p, C) \leftarrow \&res[\text{plan}](C) \\
 r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow \text{plan}(P), \text{choose}(P, X, Y) \\
 r_4: \quad \quad \text{need}(u, C) \leftarrow \&res[\text{use}](C) \\
 r_5: \quad \quad \quad \quad \quad \leftarrow \text{need}(_, \text{money})
 \end{array} \right\}
 \end{array}$$

External Atom:

- ▶ $\&g[\mathbf{x}](\mathbf{y})$
- ▶ input list $\mathbf{x} = x_1, \dots, x_n$ and output list $\mathbf{y} = y_1, \dots, y_m$ of terms
- ▶ $I \models \&g[\mathbf{x}](\mathbf{y})$
 iff **oracle function** $f_{\&g}(I, \mathbf{x}, \mathbf{y})$ is true

Program (choose a plan, choose a usage, determine resource needs):

$$\begin{array}{l}
 \text{EDB} \quad \{ \text{choose}(a, c, d) \leftarrow, \text{choose}(b, e, f) \leftarrow \} \\
 \text{IDB} \quad \left\{ \begin{array}{l}
 r_1: \text{plan}(a) \vee \text{plan}(b) \leftarrow \\
 r_2: \quad \text{need}(p, C) \leftarrow \&res[\text{plan}](C) \\
 r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow \text{plan}(P), \text{choose}(P, X, Y) \\
 r_4: \quad \text{need}(u, C) \leftarrow \&res[\text{use}](C) \\
 r_5: \quad \quad \quad \leftarrow \text{need}(_, \text{money})
 \end{array} \right\}
 \end{array}$$

External Atom:

- ▶ $I \models \&res[p](\text{money})$ if $p(C) \in I$ for $C \in \{a, f\}$,
- ▶ $I \models \&res[p](\text{time})$ if $p(C) \in I$ for $C \in \{b, c, d, e\}$,
- ▶ $I \not\models \&res[p](X)$ otherwise.

Program (choose a plan, choose a usage, determine resource needs):

$$\begin{array}{l}
 \text{EDB} \quad \{ \text{choose}(a, c, d) \leftarrow, \text{choose}(b, e, f) \leftarrow \} \\
 \text{IDB} \quad \left\{ \begin{array}{l}
 r_1: \text{plan}(a) \vee \text{plan}(b) \leftarrow \\
 r_2: \quad \text{need}(p, C) \leftarrow \&res[\text{plan}](C) \\
 r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow \text{plan}(P), \text{choose}(P, X, Y) \\
 r_4: \quad \text{need}(u, C) \leftarrow \&res[\text{use}](C) \\
 r_5: \quad \quad \quad \leftarrow \text{need}(_, \text{money})
 \end{array} \right\}
 \end{array}$$

External Atom:

- ▶ $I \models \&res[p](\text{money})$ if $p(C) \in I$ for $C \in \{a, f\}$,
- ▶ $I \models \&res[p](\text{time})$ if $p(C) \in I$ for $C \in \{b, c, d, e\}$,
- ▶ $I \not\models \&res[p](X)$ otherwise.

In our example ...

$$\begin{array}{l}
 \{ \text{plan}(a) \} \models \&res[\text{plan}](\text{money}) \quad \{ \text{plan}(a) \} \not\models \&res[\text{plan}](\text{time}) \\
 \{ \text{plan}(b) \} \not\models \&res[\text{plan}](\text{money}) \quad \{ \text{plan}(b) \} \models \&res[\text{plan}](\text{time})
 \end{array}$$

Program (choose a plan, choose a usage, determine resource needs):

$$\begin{array}{l}
 \text{EDB} \quad \{ \text{choose}(a, c, d) \leftarrow, \text{choose}(b, e, f) \leftarrow \} \\
 \text{IDB} \quad \left\{ \begin{array}{l}
 r_1: \text{plan}(a) \vee \text{plan}(b) \leftarrow \\
 r_2: \quad \text{need}(p, C) \leftarrow \&res[\text{plan}](C) \\
 r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow \text{plan}(P), \text{choose}(P, X, Y) \\
 r_4: \quad \text{need}(u, C) \leftarrow \&res[\text{use}](C) \\
 r_5: \quad \quad \quad \leftarrow \text{need}(_, \text{money})
 \end{array} \right\}
 \end{array}$$

Guessed Interpretations:

- ▶ $\{ \text{plan}(a), \&res[\text{plan}](\text{money}), \text{need}(p, \text{money}), \text{use}(c), \&res[\text{use}](\text{time}), \text{need}(u, \text{time}) \}$
- ▶ $\{ \text{plan}(a), \&res[\text{plan}](\text{money}), \text{need}(p, \text{money}), \text{use}(d), \&res[\text{use}](\text{time}), \text{need}(u, \text{time}) \}$
- ▶ $\{ \text{plan}(b), \&res[\text{plan}](\text{time}), \text{need}(p, \text{time}), \text{use}(e), \&res[\text{use}](\text{time}), \text{need}(u, \text{time}) \}$
- ▶ $\{ \text{plan}(b), \&res[\text{plan}](\text{time}), \text{need}(p, \text{time}), \text{use}(f), \&res[\text{use}](\text{money}), \text{need}(u, \text{money}) \}$

Program (choose a plan, choose a usage, determine resource needs):

$$\begin{array}{l}
 \text{EDB} \quad \{ \text{choose}(a, c, d) \leftarrow, \text{choose}(b, e, f) \leftarrow \} \\
 \text{IDB} \quad \left\{ \begin{array}{l}
 r_1: \text{plan}(a) \vee \text{plan}(b) \leftarrow \\
 r_2: \quad \text{need}(p, C) \leftarrow \&res[\text{plan}](C) \\
 r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow \text{plan}(P), \text{choose}(P, X, Y) \\
 r_4: \quad \text{need}(u, C) \leftarrow \&res[\text{use}](C) \\
 c_5: \quad \quad \quad \leftarrow \text{need}(_, \text{money})
 \end{array} \right\}
 \end{array}$$

Gussed Interpretations:

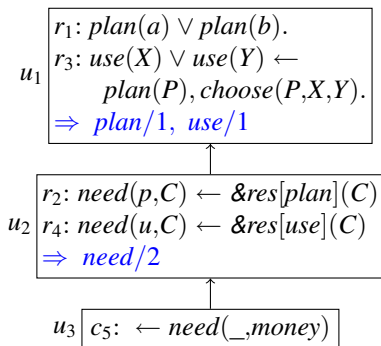
- ▶ $\{ \text{plan}(a), \&res[\text{plan}](\text{money}), \text{need}(p, \text{money}), \text{use}(c), \&res[\text{use}](\text{time}), \text{need}(u, \text{time}) \}$
- ▶ $\{ \text{plan}(a), \&res[\text{plan}](\text{money}), \text{need}(p, \text{money}), \text{use}(d), \&res[\text{use}](\text{time}), \text{need}(u, \text{time}) \}$
- ▶ $\{ \text{plan}(b), \&res[\text{plan}](\text{time}), \text{need}(p, \text{time}), \text{use}(e), \&res[\text{use}](\text{time}), \text{need}(u, \text{time}) \} \quad \Leftarrow \text{Answer Set}$
- ▶ $\{ \text{plan}(b), \&res[\text{plan}](\text{time}), \text{need}(p, \text{time}), \text{use}(f), \&res[\text{use}](\text{money}), \text{need}(u, \text{money}) \}$



- (1) calculate models of **all** program parts that do not depend on external computations that still need to be done
- (2) do external computations that depend on (1)
- (3) replace evaluated external atoms by result of (2) and restart at (1)

- (1) calculate models of **all** program parts that do not depend on external computations that still need to be done
- (2) do external computations that depend on (1)
- (3) replace evaluated external atoms by result of (2) and restart at (1)

$r_1: \text{plan}(a) \vee \text{plan}(b).$
 $r_2: \text{need}(p,C) \leftarrow$
 $\quad \&\text{res}[\text{plan}](C).$
 $r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow$
 $\quad \text{plan}(P), \text{choose}(P,X,Y).$
 $r_4: \text{need}(u,C) \leftarrow$
 $\quad \&\text{res}[\text{use}](C).$
 $c_5: \leftarrow \text{need}(_, \text{money}).$



"Evaluation Graph"

$r_1: \text{plan}(a) \vee \text{plan}(b).$
 $r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow$
 $\text{plan}(P), \text{choose}(P, X, Y).$

$r_2: \text{need}(p, C) \leftarrow$
 $\text{\&res}[\text{plan}](C)$
 $r_4: \text{need}(u, C) \leftarrow$
 $\text{\&res}[\text{use}](C)$

$c_5: \leftarrow \text{need}(_, \text{money})$

Models

$m_1 \stackrel{0:-}{=} \{\text{plan}(a), \text{use}(c)\}$

$m_2 \stackrel{0:-}{=} \{\text{plan}(a), \text{use}(d)\}$

$m_3 \stackrel{0:-}{=} \{\text{plan}(b), \text{use}(e)\}$

$m_4 \stackrel{0:-}{=} \{\text{plan}(b), \text{use}(f)\}$

$m_5 \stackrel{!:1}{=} m_1$ $m_6 \stackrel{!:2}{=} m_2$

$m_7 \stackrel{!:3}{=} m_3$ $m_8 \stackrel{!:4}{=} m_4$

"Evaluation Graph"

$r_1: \text{plan}(a) \vee \text{plan}(b).$
 $r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow$
 $\text{plan}(P), \text{choose}(P, X, Y).$

$r_2: \text{need}(p, C) \leftarrow$
 $\quad \&\text{res}[\text{plan}](C)$
 $r_4: \text{need}(u, C) \leftarrow$
 $\quad \&\text{res}[\text{use}](C)$

$c_5: \leftarrow \text{need}(_, \text{money})$

Models

$m_1 \stackrel{0:-}{=} \{ \text{plan}(a), \text{use}(c) \}$

$m_2 \stackrel{0:-}{=} \{ \text{plan}(a), \text{use}(d) \}$

$m_3 \stackrel{0:-}{=} \{ \text{plan}(b), \text{use}(e) \}$

$m_4 \stackrel{0:-}{=} \{ \text{plan}(b), \text{use}(f) \}$

$m_5 \stackrel{!:-1}{=} m_1 \quad m_6 \stackrel{!:-2}{=} m_2$

$m_7 \stackrel{!:-3}{=} m_3 \quad m_8 \stackrel{!:-4}{=} m_4$

$m_9 \stackrel{0:-5}{=} \{ \text{need}(p, \text{money}), \text{need}(u, \text{time}) \}$

$m_{10} \stackrel{0:-6}{=} \{ \text{need}(p, \text{money}), \text{need}(u, \text{time}) \}$

$m_{11} \stackrel{0:-7}{=} \{ \text{need}(p, \text{time}), \text{need}(u, \text{time}) \}$

$m_{12} \stackrel{0:-8}{=} \{ \text{need}(p, \text{time}), \text{need}(u, \text{money}) \}$

$m_{13} \stackrel{!:-9}{=} m_9 \quad m_{14} \stackrel{!:-10}{=} m_{10}$

$m_{15} \stackrel{!:-11}{=} m_{11} \quad m_{16} \stackrel{!:-12}{=} m_{12}$

"Evaluation Graph"

$$r_1: \text{plan}(a) \vee \text{plan}(b).$$

$$r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow$$

$$\text{plan}(P), \text{choose}(P, X, Y).$$

$$r_2: \text{need}(p, C) \leftarrow$$

$$\quad \&\text{res}[\text{plan}](C)$$

$$r_4: \text{need}(u, C) \leftarrow$$

$$\quad \&\text{res}[\text{use}](C)$$

$$c_5: \leftarrow \text{need}(_, \text{money})$$

Models

$$m_1 \stackrel{0:-}{=} \{\text{plan}(a), \text{use}(c)\}$$

$$m_2 \stackrel{0:-}{=} \{\text{plan}(a), \text{use}(d)\}$$

$$m_3 \stackrel{0:-}{=} \{\text{plan}(b), \text{use}(e)\}$$

$$m_4 \stackrel{0:-}{=} \{\text{plan}(b), \text{use}(f)\}$$

$$m_5 \stackrel{1:1}{=} m_1 \quad m_6 \stackrel{1:2}{=} m_2$$

$$m_7 \stackrel{1:3}{=} m_3 \quad m_8 \stackrel{1:4}{=} m_4$$

$$m_9 \stackrel{0:5}{=} \{\text{need}(p, \text{money}), \text{need}(u, \text{time})\}$$

$$m_{10} \stackrel{0:6}{=} \{\text{need}(p, \text{money}), \text{need}(u, \text{time})\}$$

$$m_{11} \stackrel{0:7}{=} \{\text{need}(p, \text{time}), \text{need}(u, \text{time})\}$$

$$m_{12} \stackrel{0:8}{=} \{\text{need}(p, \text{time}), \text{need}(u, \text{money})\}$$

$$m_{13} \stackrel{1:9}{=} m_9 \quad m_{14} \stackrel{1:10}{=} m_{10}$$

$$m_{15} \stackrel{1:11}{=} m_{11} \quad m_{16} \stackrel{1:12}{=} m_{12}$$

$$m_{17} \stackrel{0:15}{=} \emptyset$$

"Evaluation Graph"

$r_1: \text{plan}(a) \vee \text{plan}(b).$
 $r_3: \text{use}(X) \vee \text{use}(Y) \leftarrow$
 $\text{plan}(P), \text{choose}(P, X, Y).$

$r_2: \text{need}(p, C) \leftarrow$
 $\quad \&\text{res}[\text{plan}](C)$
 $r_4: \text{need}(u, C) \leftarrow$
 $\quad \&\text{res}[\text{use}](C)$

$c_5: \leftarrow \text{need}(_, \text{money})$

Models

$m_1 \stackrel{0:-}{=} \{ \text{plan}(a), \text{use}(c) \}$

$m_2 \stackrel{0:-}{=} \{ \text{plan}(a), \text{use}(d) \}$

$m_3 \stackrel{0:-}{=} \{ \text{plan}(b), \text{use}(e) \}$

$m_4 \stackrel{0:-}{=} \{ \text{plan}(b), \text{use}(f) \}$

$m_5 \stackrel{1:1}{=} m_1$ $m_6 \stackrel{1:2}{=} m_2$

$m_7 \stackrel{1:3}{=} m_3$ $m_8 \stackrel{1:4}{=} m_4$

$m_9 \stackrel{0:5}{=} \{ \text{need}(p, \text{money}), \text{need}(u, \text{time}) \}$

$m_{10} \stackrel{0:6}{=} \{ \text{need}(p, \text{money}), \text{need}(u, \text{time}) \}$

$m_{11} \stackrel{0:7}{=} \{ \text{need}(p, \text{time}), \text{need}(u, \text{time}) \}$

$m_{12} \stackrel{0:8}{=} \{ \text{need}(p, \text{time}), \text{need}(u, \text{money}) \}$

$m_{13} \stackrel{1:9}{=} m_9$ $m_{14} \stackrel{1:10}{=} m_{10}$

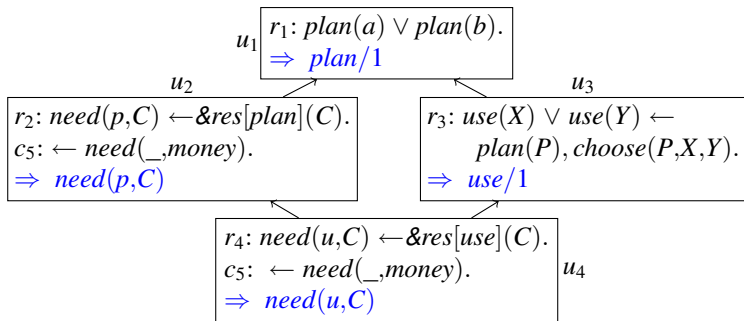
$m_{15} \stackrel{1:11}{=} m_{11}$ $m_{16} \stackrel{1:12}{=} m_{12}$

$m_{17} \stackrel{0:15}{=} \emptyset$

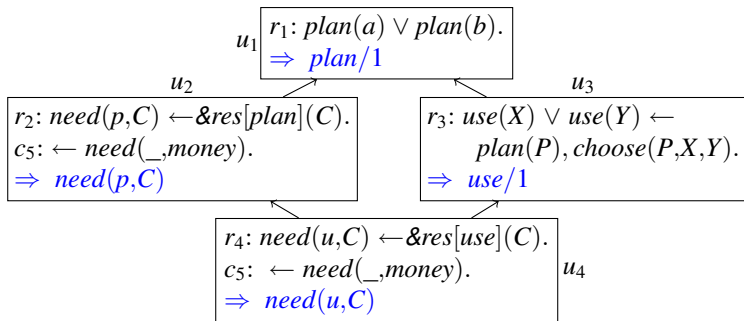


- ▶ We compute the **largest possible fragment**
 - ⇒ multiplication of independent guesses
 - ⇒ redundant evaluation of external atoms
- ▶ Constraints are considered if **all dependencies** are fulfilled
 - ⇒ some constraints could safely eliminate models earlier
- ▶ We calculate **all models** at one unit, then go to the next unit
 - ⇒ if we need only one model, we calculate unnecessary models

⇒ **Several improvements possible!**



- ▶ **Acyclic** Evaluation Graph
- ▶ Rules cannot be shared among units
- ▶ **Constraints can be shared** among units
- ▶ Rule dependencies must be fulfilled
- ▶ Constraint dependencies (less strict) must be fulfilled
 \Rightarrow e.g., c_5 has no negative body, therefore it can be freely shared

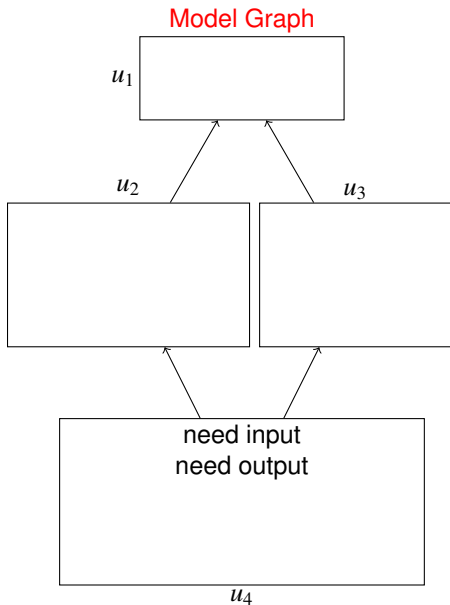
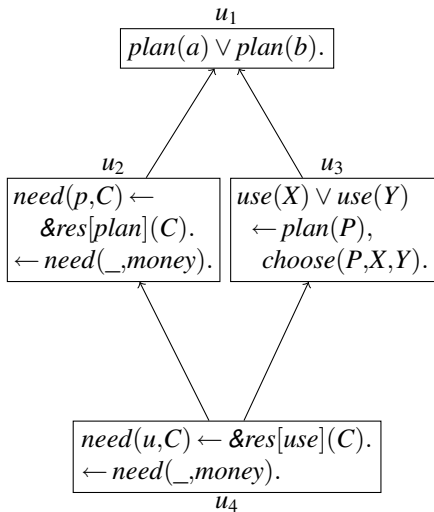


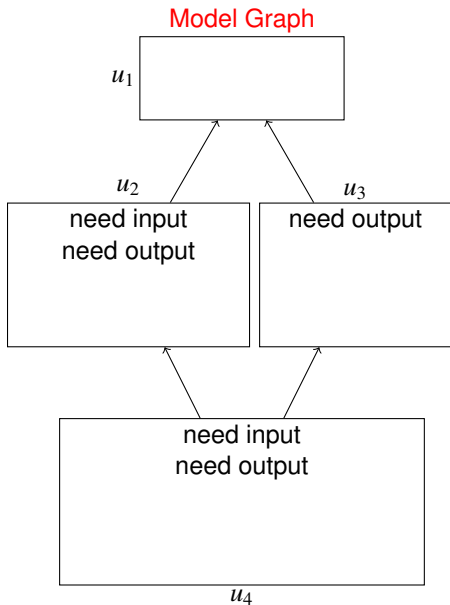
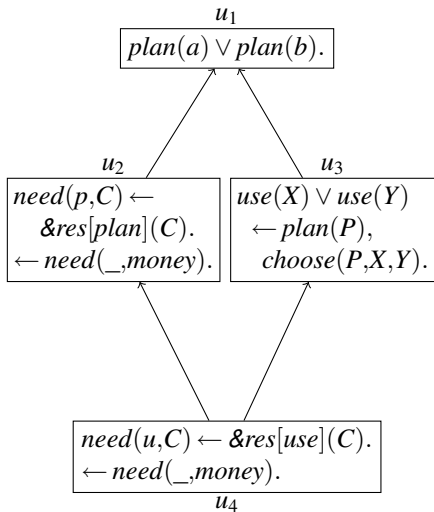
Input and Output models at each Evaluation Unit:

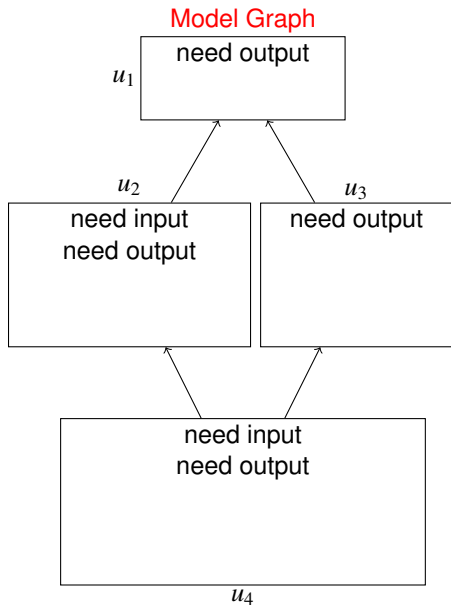
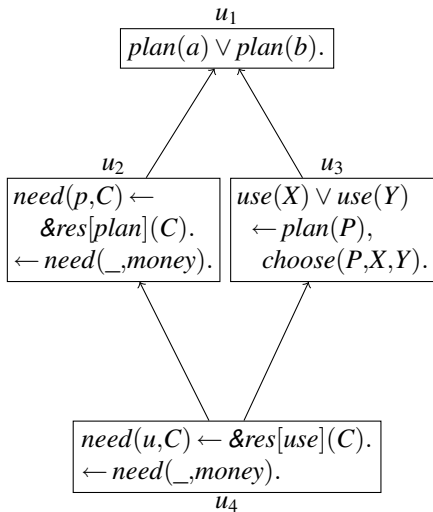
- ▶ output model = evaluation of program fragment on input model
- ▶ input model = joined output model of predecessor unit(s)
- ▶ Principle: **common ancestor model** at common ancestor units

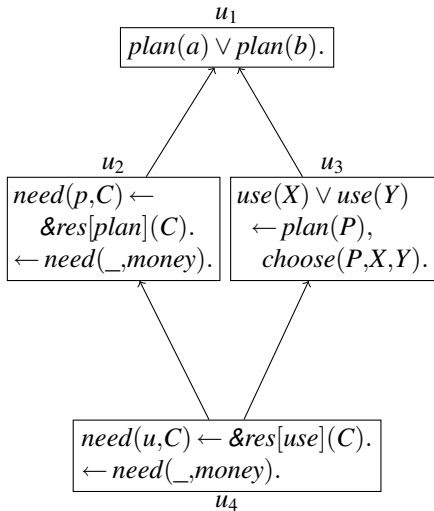
Soundness + Completeness Theorem:

I = union of connected output models at every unit
 iff I = answer set of the program

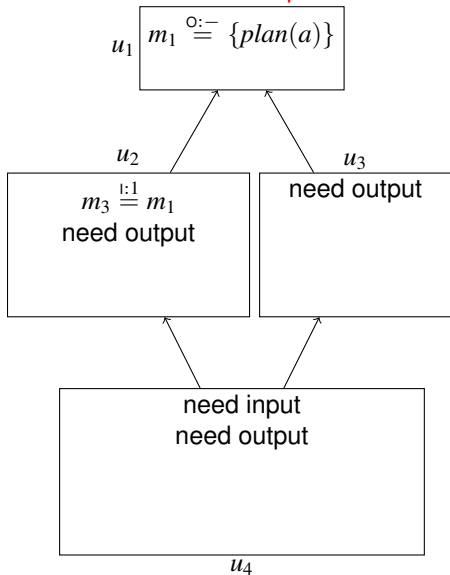


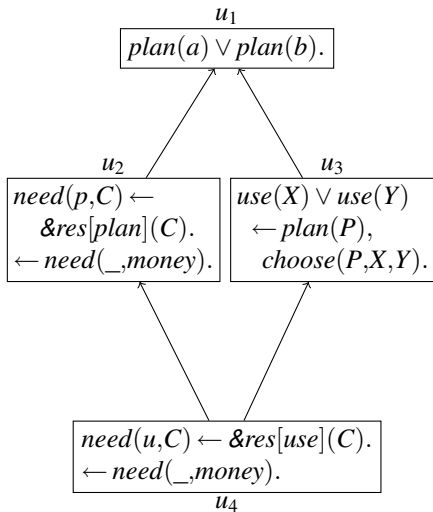




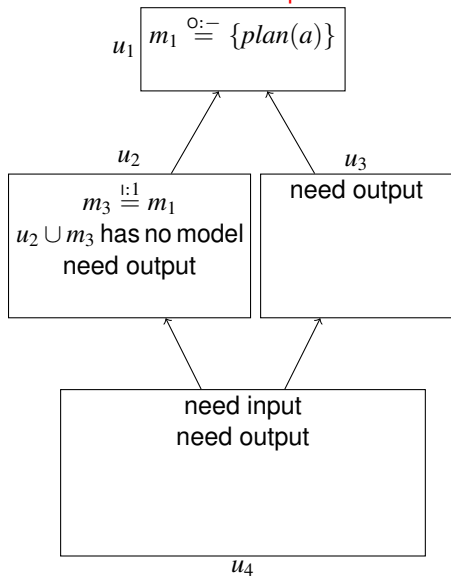


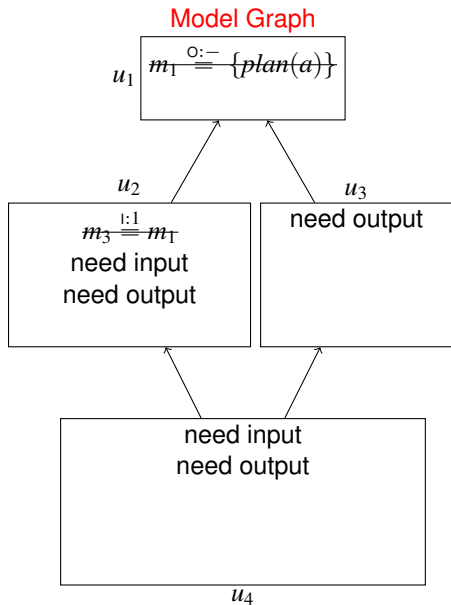
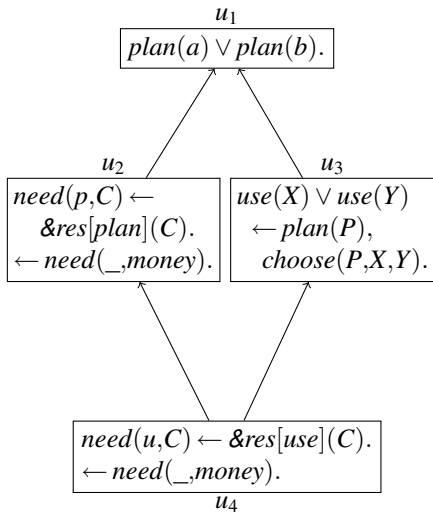
Model Graph

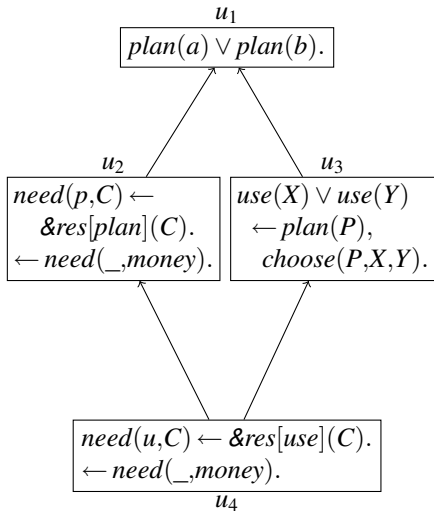




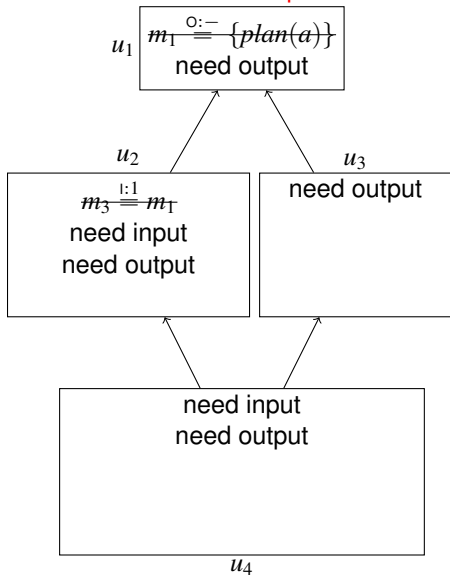
Model Graph

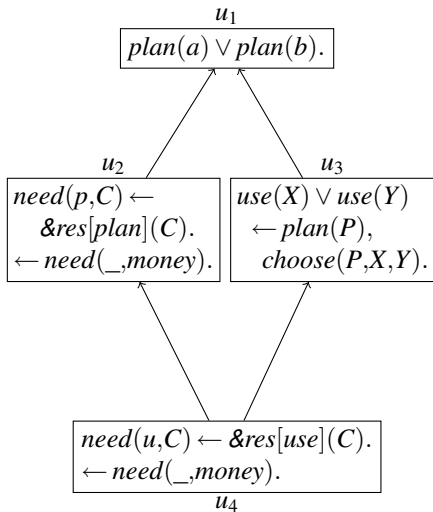




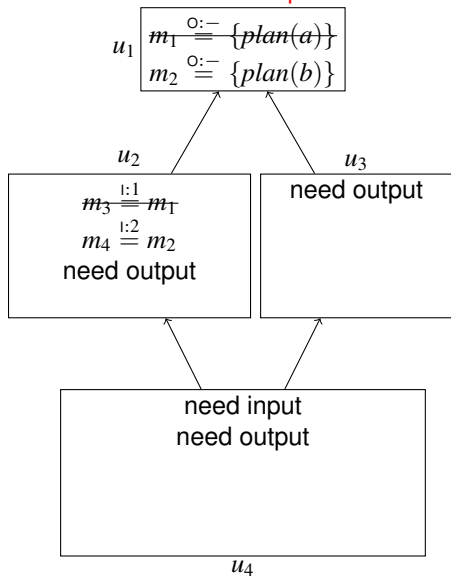


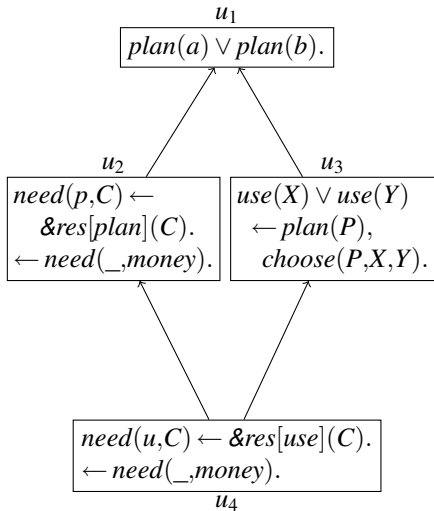
Model Graph



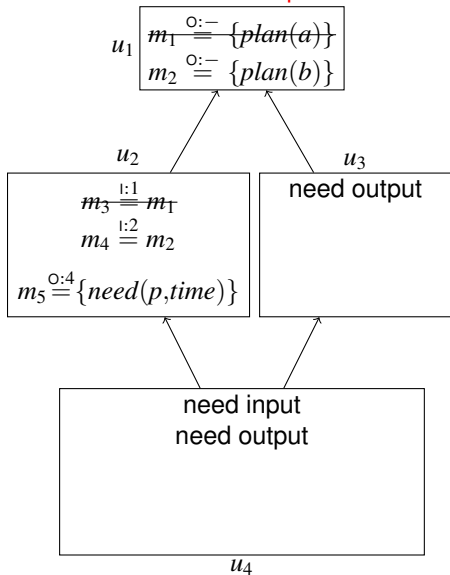


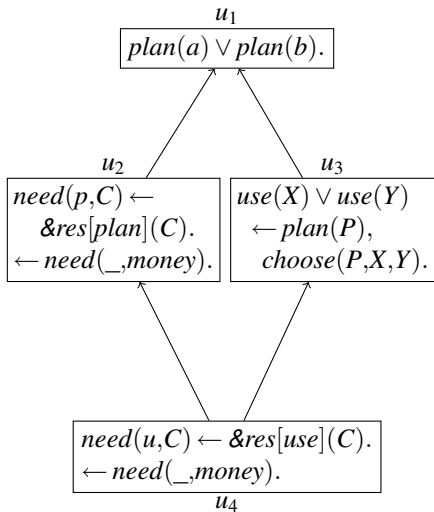
Model Graph



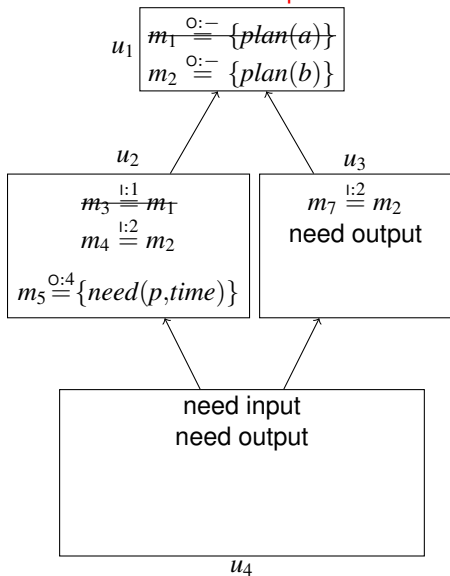


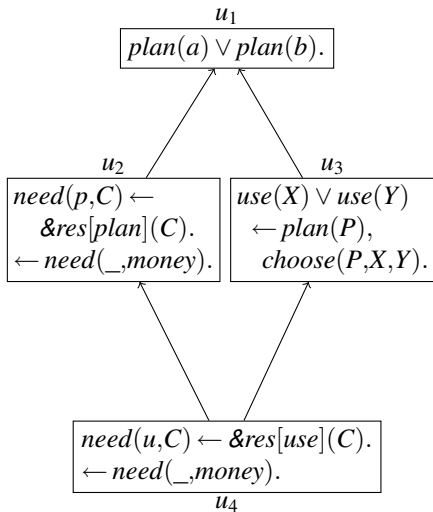
Model Graph



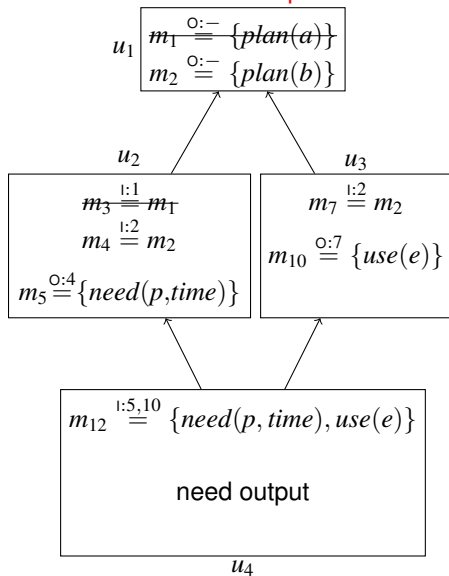


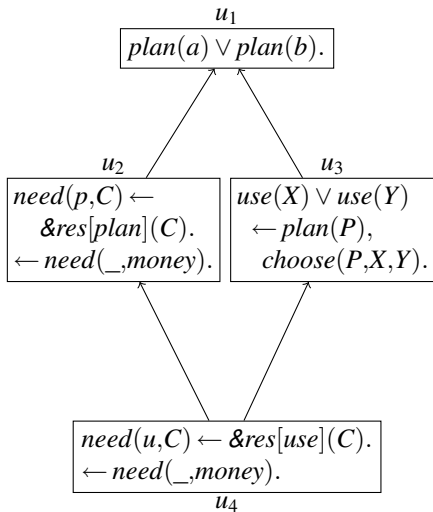
Model Graph



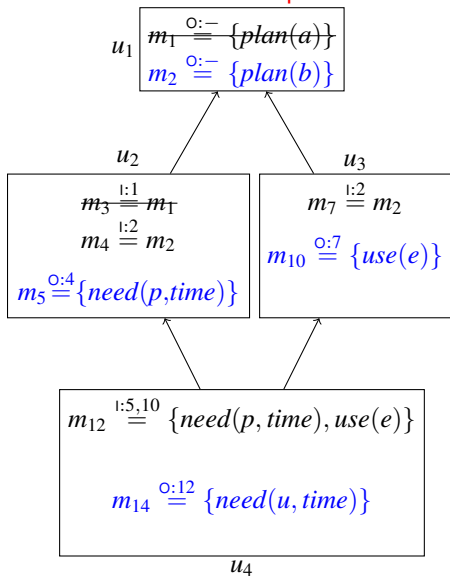


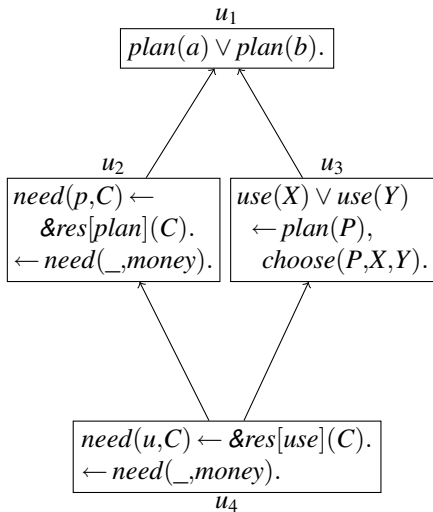
Model Graph



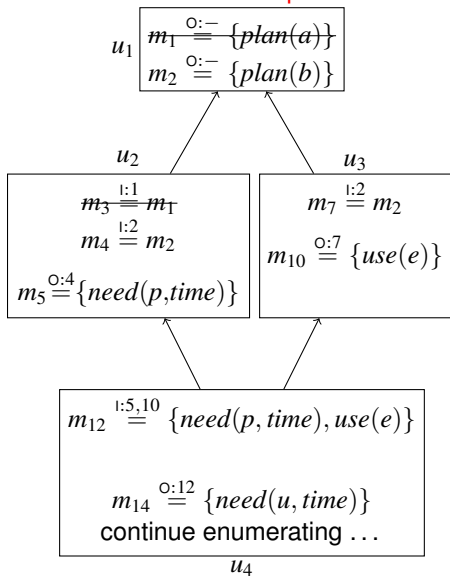


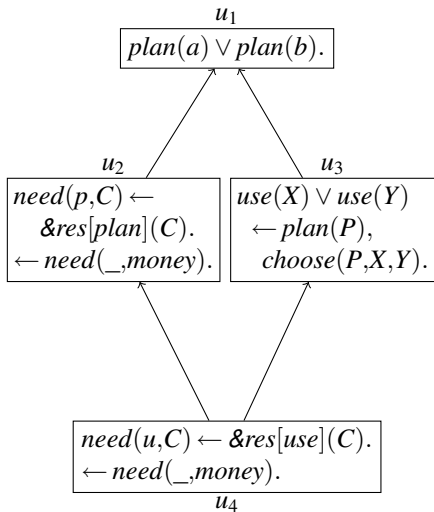
Model Graph



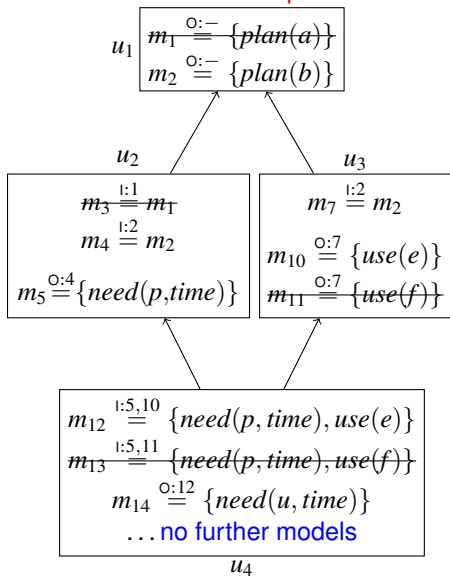


Model Graph





Model Graph





Acyclic Directed Evaluation Graph

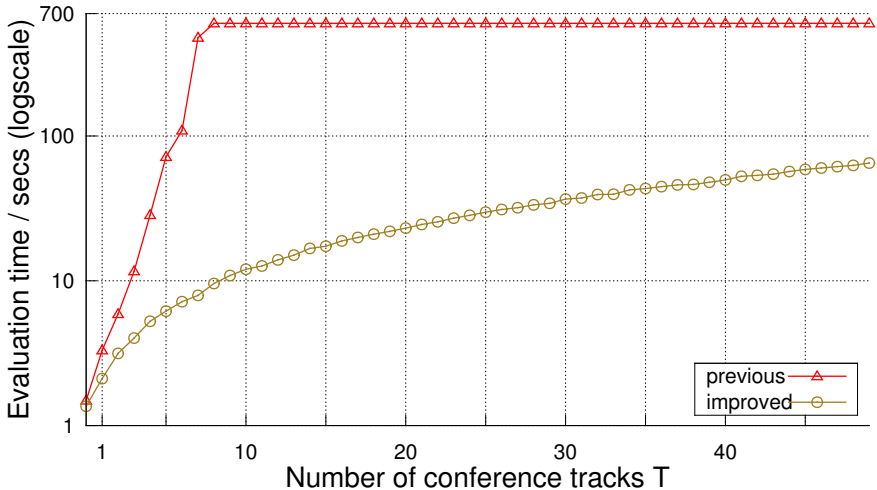
- ⇒ independent program fragments evaluated independently
- ⇒ fewer redundant evaluations
- ⇒ parallel evaluation possible

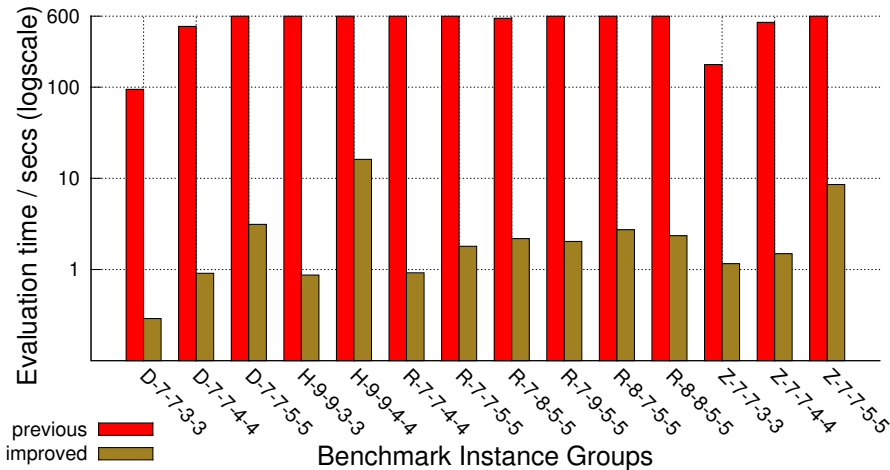
Constraints can be shared between units

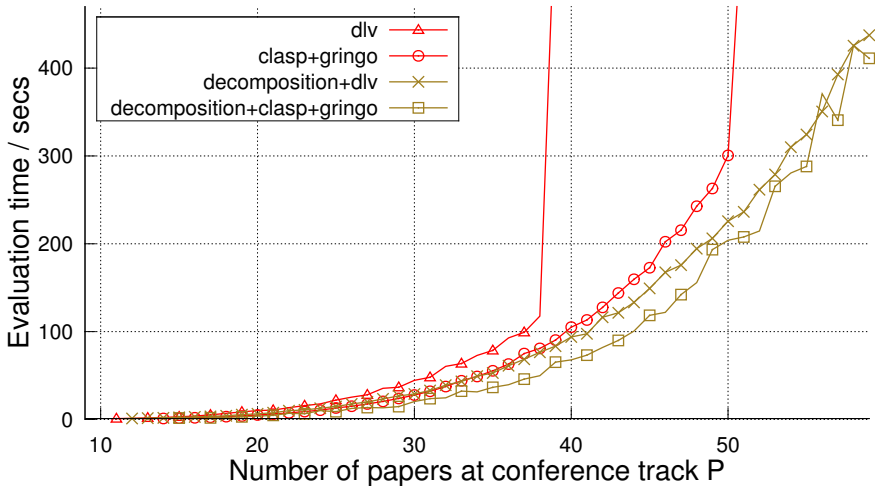
- ⇒ early model elimination

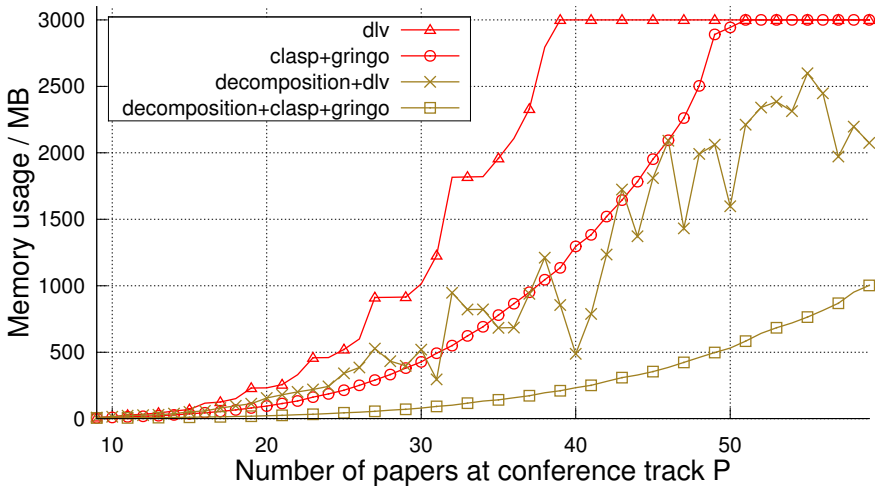
Calculating one model at a time

- ⇒ first model obtained faster
- ⇒ can discard some intermediate results early









Novel Theoretical HEX Evaluation Framework

- ⇒ Evaluation Graph + Model Graph
- ⇒ exponential speedup for certain instances
- ⇒ potential for optimization strategies
(emphasis on parallelization vs. memory usage)
- ⇒ can even outperform native ASP solvers in some cases

Improved Implementation (Open Source)

- ⇒ backends `dlv` and `clingo`
- ⇒ Instructions for building and detailed experimental results:
click Experiments on the `dlvhex` homepage
<http://www.kr.tuwien.ac.at/research/systems/dlvhex/>

- ▶ 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.