Aalto University
School of Science

Rewriting Optimization Statements in Answer-Set Programs

Jori Bomanson[1], Martin Gebser[2], Tomi Janhunen[1]

[1] Aalto University, Finland
[2] University of Potsdam, Germany

Computational Logic Day, Espoo, December 8, 2016

Answer-Set Programs and Optimization Rewriting

Input: A ground ASP program P consisting of normal rules

$$
\begin{aligned}
a &\mathrel{:\!-} \text{not } b.\\
b &\mathrel{:\!-} \text{not } a.\\
c &\mathrel{:\!-} a, b.\\
d &\mathrel{:\!-} e.\\
e &\mathrel{:\!-} d.
\end{aligned}
$$

Answer-Set Programs and Optimization Rewriting

Input: A ground ASP program P consisting of normal rules and a minimization statement:

$$a \mathrel{:-} \text{not } b.$$
$$b \mathrel{:-} \text{not } a.$$
$$c \mathrel{:-} a, b.$$
$$d \mathrel{:-} e.$$
$$e \mathrel{:-} d.$$
$$\#\text{minimize } \{5 : a; 6 : b; 2 : c; 3 : d; 10 : e\}.$$

Answer-Set Programs and Optimization Rewriting

Input: A ground ASP program P consisting of normal rules and a minimization statement:

$$
\begin{array}{l}
a \,{:}\!-\, \text{not } b. \\
b \,{:}\!-\, \text{not } a. \\
c \,{:}\!-\, a, b. \\
d \,{:}\!-\, e. \\
e \,{:}\!-\, d. \\
\#\text{minimize } \{5 : a; 6 : b; 2 : c; 3 : d; 10 : e\}.
\end{array}
$$

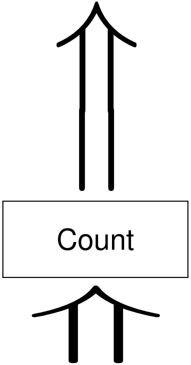
Output: P with a modified statement and added normal rules.

Goal: Keep answer sets and boost solving performance.

Optimization Rewriting vs Normalization

In this work:

$$\begin{aligned}\min\ & 1\mathrm{u}_{19} + 2\mathrm{u}_{25} + 4\mathrm{u}_{30} + \\ & 8\mathrm{u}_{34} + 8\mathrm{u}_{35} + 8\mathrm{u}_{36}\end{aligned}$$



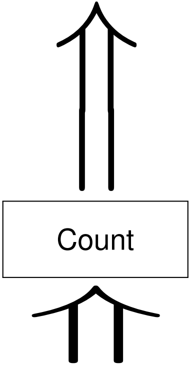
Count

$$\min\ 5a + 6b + 2c + 3d + 10e$$

Optimization Rewriting vs Normalization

In this work:

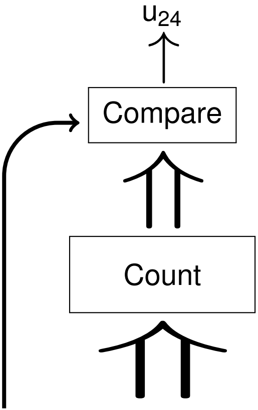
$$\min\ 1u_{19} + 2u_{25} + 4u_{30} + 8u_{34} + 8u_{35} + 8u_{36}$$



$$\min\ 5a + 6b + 2c + 3d + 10e$$

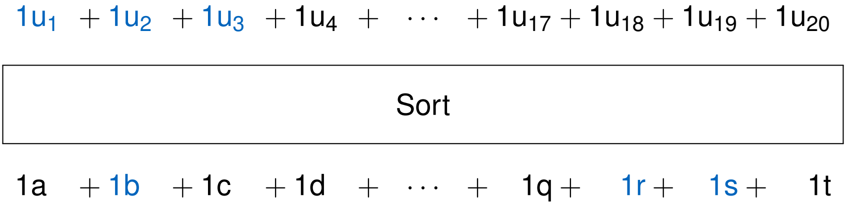
Previously:

h :– u_{24}.



$$12 \leq 5a + 6b + 2c + 3d + 10e$$

Unweighted Expressions

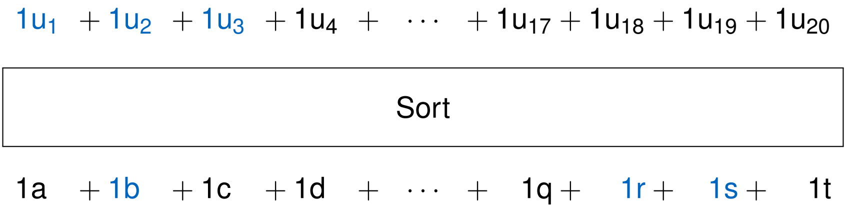


$$1u_1 + 1u_2 + 1u_3 + 1u_4 + \cdots + 1u_{17} + 1u_{18} + 1u_{19} + 1u_{20}$$

Sort

$$1a + 1b + 1c + 1d + \cdots + 1q + 1r + 1s + 1t$$

Unweighted Expressions



$$1u_1 + 1u_2 + 1u_3 + 1u_4 + \cdots + 1u_{17} + 1u_{18} + 1u_{19} + 1u_{20}$$

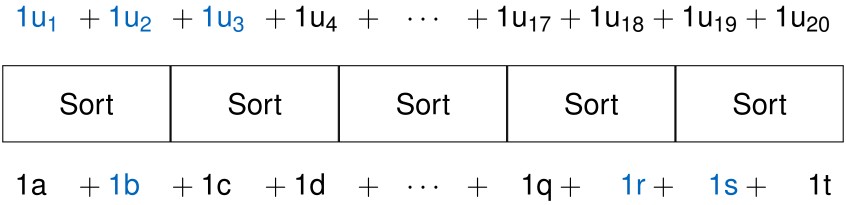
Sort

$$1a + 1b + 1c + 1d + \cdots + 1q + 1r + 1s + 1t$$

Example:

$\mathrm{u}_1 \text{ :– } a.$
$\mathrm{u}_1 \text{ :– } b.$
$\mathrm{u}_2 \text{ :– } a, b.$

Unweighted Expressions



$1u_1 + 1u_2 + 1u_3 + 1u_4 + \cdots + 1u_{17} + 1u_{18} + 1u_{19} + 1u_{20}$

Sort Sort Sort Sort Sort

$1a + 1b + 1c + 1d + \cdots + 1q + 1r + 1s + 1t$

Example:

$u_1 \text{ :- } a.$
$u_1 \text{ :- } b.$
$u_2 \text{ :- } a, b.$

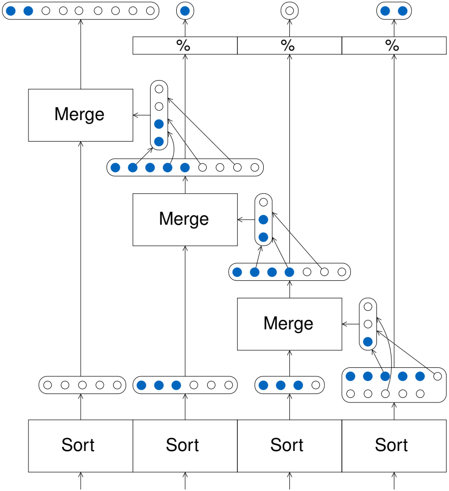
$u_3 \text{ :- } c.$
$u_3 \text{ :- } d.$
$u_4 \text{ :- } c, d.$

$\cdots$

$u_{17} \text{ :- } q.$
$u_{17} \text{ :- } r.$
$u_{18} \text{ :- } q, r.$

$u_{19} \text{ :- } s.$
$u_{19} \text{ :- } t.$
$u_{20} \text{ :- } s, t.$

Weighted Expressions



Eén and Sörensson, JSAT'06

- Pseudo-Boolean ⇒ sorting networks ⇒ SAT.

Bomanson, Gebser, and Janhunen, JELIA'14

- Weight rules ⇒ normal rules

Size: $O(b(\log b)^2)$ in the number of bits.

Selective Rewriting

- To reduce the cost of rewriting

  $$5a + 6b + 2c + 3d + 10e$$

- ... we may rewrite one chunk:

  $$= (10e + 6b + 5a) + 3d + 2c$$

Selective Rewriting

- To reduce the cost of rewriting

  $$5a + 6b + 2c + 3d + 10e$$

- ... we may rewrite one chunk:

  $$= (10e + 6b + 5a) + 3d + 2c$$

- ... or one chunk of digits:

  $$\begin{aligned} &= 4a + 1a + 4b + 2b + 2c + 2d + 1d + 8e + 2e \\ &= (4a + 4b + 8e) + 1a + 2b + 2c + 2d + 1d + 2e \end{aligned}$$

Significance Based Selection

Global $k = 2$**:**

	5	6	2	3	10
1	1	0	0	1	0
2	0	1	1	1	1
4	1	1	0	0	0
8	0	0	0	0	1

Significance Based Selection

Local $k = 1$:

	5	6	2	3	10
1	1	0	0	1	0
2	0	1	1	1	1
4	1	1	0	0	0
8	0	0	0	0	1

Example

$$\#\text{minimize}\ \{\quad 5 : a; \qquad 6 : b; \qquad 2 : c; \qquad 3 : d; \qquad 10 : e\}$$

↯

$$\#\text{minimize}\ \{1, 1 : a; \quad 2, 3 : b; \qquad\qquad\qquad 1, 2 : d; \quad 2, 4 : e\}$$

$$4, 1 : a; \quad 4, 2 : b; \quad 2, 1 : c; \quad 2, 2 : d; \quad 8, 1 : e;$$

Example

$$\#\text{minimize}\ \{\ 5:a;\quad 6:b;\quad 2:c;\quad 3:d;\quad 10:e\}$$

$$\rotatebox{-90}{$\rightsquigarrow$}$$

$$\begin{array}{llllll} \#\text{minimize}\ \{ 1,1:a; & 2,3:b; & & 1,2:d; & 2,4:e\} \\ \phantom{\#\text{minimize}\ \{}4,1:a; & 4,2:b; & 2,1:c; & 2,2:d; & 8,1:e; \end{array}$$

$$\rotatebox{-90}{$\rightsquigarrow$}$$

$$\begin{array}{llllll} \#\text{minimize}\ \{ 1,1:a; & 2,3:b; & & 1,2:d; & 2,4:e\} \\ \phantom{\#\text{minimize}\ \{}4,1:x; & 4,2:y; & 2,1:u; & 2,2:v; & 8,1:w; \end{array}$$

$$\begin{array}{lll} x \mathrel{:\!-} a. & u \mathrel{:\!-} c. & w \mathrel{:\!-} e. \\ x \mathrel{:\!-} b. & u \mathrel{:\!-} d. & \\ y \mathrel{:\!-} a, b. & u \mathrel{:\!-} c, d. & \end{array}$$

Experiments

Numbers of instances solved by `clasp --config=trendy`:

	Connected Still-Life	Crossing Minimization	Maximal Clique	Timetabling
#	120	85	186	57
–	22	50	51	28
sort	50	72	143	36
sort/64	105	76	153	39

Experiments

Numbers of instances solved by `clasp --config=trendy`:

	Connected Still-Life	Crossing Minimization	Maximal Clique	Timetabling	Bayes Alarm	Bayes Water	Bayes Hailfinder	Graphical Models
#	120	85	186	57	32	27	51	72
–	22	50	51	28	9	15	35	27
sort	50	72	143	36				
sort/64	105	76	153	39				
mixed-radix				37	5	19	32	19
global $k = 7$				28	8	19	31	23
local $k = 1$				38	9	18	45	26
local $k = 2$				39	10	19	48	27
local $k = 3$				38	10	23	48	27

Conclusion

- We rewrite optimization statements in ASP programs.
- The rewritings are similar to translations of pseudo-Boolean constraints.
- Based on experiments, limiting output size by rewriting in chunks or rewriting only important parts selectively is beneficial.