Splitting a Logic Program Revisited

Jianmin Ji[a], Hai Wan[b+], Ziwei Huo[b], and ZhenFeng Yuan[b]

[a]University of Science and Technology of China, Hefei, China [b]Sun Yat-sen University, Guangzhou, China [+]Email: wanhai@mail.sysu.edu.cn

Problem

Lifschitz and Turner introduced the notion of the splitting set and provided a method to divide a logic program into two parts. They showed that the task of computing the answer sets of the program can be converted into the tasks of computing the answer sets of these parts [1].

However, the empty set and the set of all atoms are the only two splitting sets for many programs, then these programs cannot be divided by the splitting method.

Contributions

we propose a new splitting method that allows the program to be split into two parts by an arbitrary set of atoms, while one of them, the "top" part, may introduce some new atoms.

We show that the task of computing the answer sets of the program can be converted into the tasks of computing the answer sets of these parts.

The experiments show that, the splitting result in our method is the same as the result in Lifschitz and Turner's splitting method.

Splitting Sets

A set U of atoms is called a *splitting set* of a program P, if for each $r \in P$, $head(r) \cap U \neq \emptyset$ implies $Atoms(r) \subseteq U$.

The *bottom* and *top* of P *w.r.t.* U are denoted as $b_U(P)$ and $t_U(P)$. And they are defined as : $b_U(P) = \{r \in P \mid head(r) \cap U \neq \emptyset\}$. $t_U(P) = P \setminus b_U(P)$.

And $e_U(P, X)$ is the set of rules obtained by deleting some rules from P *w.r.t.* U and X.

A *solution* of P *w.r.t.* U is a pair $\langle X, Y\rangle$ of sets of atoms *s.t.* X is an answer set of $b_U(P)$, and Y is an answer set of $e_U(P \setminus b_U(P), X)$.

Example 2. Consider the logic program P_2:
$a \leftarrow not\, d.\;\; d \leftarrow not\, c.\;\; a \leftarrow c, d.\;\; c \leftarrow .$

$\{c, d\}$ is a splitting set of P_2 and the bottom of P *w.r.t.* $\{c, d\}$ is $b_{\{c,d\}}(P_2) = \{d \leftarrow not\, c.\;\; c \leftarrow .\}$. Further more, $e_{\{c,d\}}(P_2 \setminus b_{\{c,d\}}(P_2), \{c\}) = \{a \leftarrow .\}$. And $\langle\{c\}, \{a\}\rangle$ is a solution of P_2 *w.r.t.* $\{c, d\}$.

Theorem 2 (Splitting Set Theorem). Let U be a splitting set of a program P. A set S is an answer set of P iff $S = X \cup Y$ for some solution $\langle X, Y\rangle$ of P *w.r.t.* U.

Splitting with arbitrary set of atoms for NLP

For extending the splitting set theorem to arbitrary set of atoms for NLP, we introduce some notions. Let U, X be sets of atoms, P an NLP.

$EC_U(P)$ and $ECC_U(P, X)$ are set of rules based on $Atoms(b_U(P)) \setminus U$.

The *in-rules* of P *w.r.t.* U, denoted by $in_U(P)$, is the set of rules $r \in P$ *s.t.* $head(r) \cap U \neq \emptyset$ and $(body^+(r) \cup head(r)) \not\subseteq U$, the *out-rules*, denoted by $out_U(P)$ is similar to $in_U(P)$.

A set of atoms E is a *semi-loop* of P *w.r.t.* U if there exists a loop L of P *s.t.* $E = L \cap U$ and $E \subset L$. We denote

$$SL_U(P, X) = \{E \mid E \text{ is a semi-loop of } P \text{ w.r.t. } U,\; E \subseteq X \text{ and } R^-(E, P, X) \subseteq in_U(P)\}.$$

Finally, we propose a new *top* of P *w.r.t.* U under X, denoted by $t_U(P, X)$, is the union of the following sets of rules:

- $P \setminus (b_U(P) \cup out_U(P))$,
- $\{x_E \leftarrow body(r) \mid r \in in_U(P)$ and $r \in R^-(E, P, X)\}$, for each $E \in SL_U(P, X)$,
- $\{head(r) \leftarrow x_{E_1}, \ldots, x_{E_t}, body(r) \mid r \in out_U(P)$, for all possible $E_i \in SL_U(P, X)$ $(1 \leq i \leq t)$ *s.t.* $body^+(r) \cap E_i \neq \emptyset\}$,

where x_E's are new atoms *w.r.t.* $E \in SL_U(P, X)$.

A *solution* of P *w.r.t.* U is a pair $\langle X, Y\rangle$ of sets of atoms *s.t.*

- X is an answer set of $b_U(P) \cup EC_U(P)$,
- Y is an answer set of $e_U(t_U(P, X), X) \cup ECC_U(P, X)$.

Theorem 3 . Let P be an NLP and U a set of atoms. A set S is an answer set of P iff $S = (X \cup Y) \cap Atoms(P)$ for some solution $\langle X, Y\rangle$ of P *w.r.t.* U.

We try Hamiltonian Circuit (HC) problem in our experiment. The numbers under "whole" refer to the running times (in seconds) of clasp for the whole programs.

Problem	whole	bottom	top	bottom+top
2-10	0.03	0.01	0.02	0.03
2-15	0.53	0.03	0.16	0.19
2-20	1.84	0.03	0.68	0.71
2-25	5.56	0.07	2.25	2.31
2-30	14.07	0.15	5.66	5.81
2-35	27.16	0.23	12.61	12.83
2-40	52.53	0.32	23.61	23.93
2-45	106.93	0.59	48.73	49.32
2-50	171.94	0.75	80.95	81.70

DLP and Strongly Splitting

We extend the splitting method to DLPs. All the notions are defined the same except for the *top* of a DLP P *w.r.t.* U under X, denoted by $t_U(P, X)$. It is the union of following set of rules:

- $P \setminus (b_U(P) \cup out_U(P))$,
- $\{\{x_E\} \cup head(r) \setminus E \leftarrow body(r) \mid r \in in_U(P)$ and $r \in R^-(E, P, X)\}$, for each $E \in SL_U(P, X)$,
- $\{head(r) \leftarrow x_{E_1}, \ldots, x_{E_t}, body(r) \mid r \in out_U(P)$, for all possible $E_i \in SL_U(P, X)$ $(1 \leq i \leq t)$ *s.t.* $body^+(r) \cap E_i \neq \emptyset\}$,

For any program P' that does not contain atoms in U, the answer sets of $P \cup P'$ can be computed from the answer sets of $b_U(P)$ and a program constructed from $(P \setminus b_U(P)) \cup P'$. However, the property does not hold for arbitrary sets of atoms in our splitting method. We introduce the *strong splitting method* to solve the problem. *Strong splitting method* only replaces $SL_U(P, X)$ to $SS_U(P, X)$ with all the other notions remaining the same. And, we denote

$$SS_U(P, X) = \{E \mid E \text{ is a nonempty subset of } U,\; E \subseteq X \text{ and } R^-(E, P, X) \subseteq in_U(P)\}.$$

Program Simplification

Program simplification uses a consequence to simplify the program. The answer sets of the program can be computed from resulting program.

We define $tr_p(P, L)$ to be the program obtained from P by

1. deleting each rule r that has an atom $p \in head(r)$ or $p \in body^-(r)$ with $p \in L$, and
2. replacing each rule r that has an atom $p \in body^+(r)$ with $p \in L$ by the rule $head(r) \leftarrow body^+(r) \setminus L, body^-(r)$.

And we define the *consequence top* of P w.r.t U, denoted by $ct_U(P)$, to be the union of the following sets of rules:

- $P \setminus (b_U(P) \cup out_U(P))$,
- $\{\{x_E\} \cup head(r) \setminus E \leftarrow body(r) \mid r \in in'_U(P)$ and $r \in R^-(E, P)\}$, for each $E \in CS_U(P)$,
- $\{head(r) \leftarrow x_{E_1}, \ldots, x_{E_t}, body(r) \mid r \in out_U(P)$, for all possible $E_i \in CS_U(P)$ $(1 \leq i \leq t)$ *s.t.* $body^+(r) \cap E_i \neq \emptyset\}$,
- $\{\leftarrow not\, x_E\}$, for each $E \in CS_U(P)$,

Proposition 8 Let set U of atoms be a consequence of an NLP P. A set S is an answer set of P iff there is an answer set S^* of $tr_p(ct_U(P), U)$ *s.t.* $S \setminus U = S^* \cap Atoms(P)$.

A Future Direction

The idea of splitting will find many other uses, for instance in the investigation of incremental ASP solvers and forgetting.

References

[1] Lifschitz, V., and Turner, H. 1994. Splitting a logic program.. In *Proceedings of the 11th International Conference on Logic Programming(ICLP-94)* 23-37.

Source Code

For more information and source code, please visit:

`http://ss.sysu.edu.cn/~wh/splitting.html`