A Fixed Point Representation of References

Susumu Yamasaki

Department of Computer Science, Okayama University, Okayama, Japan
yamasaki@momo.cs.okayama-u.ac.jp

Abstract. This position paper is concerned with the *reference* in computer science. We have a formal representation of lazy references in contrast to eager and failure ones. The representation problem is motivated by static analysis in Web accessibility. A fixed point theory is adopted for such an analysis.

1 Introduction

To make analyses in Web usability or accessibility, we aim at capturing the link situation on the Web sites and referential relations among Web site pages. For an apperception of the link structure, this position paper deals with static analysis of relations of *references* which are concretized as Web site pages. The total reference structure is described by a fixed point of an associated mapping for the structure. As regards static analysis, several frameworks have been well established. Hybrid logic, which involves both state-dependent and modal operators, is a formal system with logical meanings of states and worlds ([1, 2]). Relations between the events are discussed through predicates in classical and modal logic ([3, 9]). The event as the cause-and-effect relationship is made clear from the view of rule-based system ([21]). Correlation between action and knowledge has also been studied ([14]). A mathematical behaviour of action is formulated in [13], while action may be captured by modal logic ([6]). The agent technology style is current as in [15], where algebraic approach to process originates from [8, 12] such that a logical viewpoint is given in the paper ([10]). A multi-agent is well designed in terms of modal logic ([7]).

In this position paper, based on the first-order logic (or the propositional logic) analysis approach ([5]), we see a mathematical aspect of reference structures relevant to Web site pages with fixed point theory. A Web site page recursively includes page references, where the page is itself a reference from other site pages. So far we see that there is a simple structure for some page A as a primary one: A primary reference A (recursively) includes references $B_1, \ldots, B_n$, where A may be referred to by others, and some of $B_1, \ldots, B_n$ may not be available without any correct link. As regards how to make use of the references, we can think that:

- To visit the (page) reference is regarded as eager.
- Not to visit (but to see only the name of) the (page) reference is regarded as lazy.

– Non-available reference for visit is regarded as a failure.

Whether or not a (page) reference is visited is supposedly determined by the user (visitor). The primary reference (which the user now pays attention to and which includes other references in) is thus interpreted as:

(i) eager if all the included references are eager.
(ii) lazy if it never occurs as a primary one such that it is designated as lazy, or if it is a primary one where at least one included reference is not eager and other included references are eager or lazy.
(iii) a failure if it is not available as a primary one, or if a primary reference with at least one included reference is a failure.

Note that the primary reference is interpreted as eager if it contains no reference. The classification of eager and lazy references for this case looks like the standard evaluation about call-by-value (eager) and call-by-name (lazy) modes of [17]. We then have a problem to see what set of lazy references is. The set of all considerable references is still finite, but it must be large enough to want to have a treatment to cover the case that the set may be countably infinite. A fixed point theory for the complete lattice is a technique as in [11, 18], to be incorporated into analysis and classification of eager and lazy reference sets, where the references are organized likely by recursive rule structures of the form: the reference including reference sequences.

2 Representation of References

In this paper, we consider recursive structures of references, which are given as a set of finite or countably infinite rules of the form $A \rhd A_1 \dots A_l$ $(l \geq 0)$, where A, A_i are *references*. A is the head, while $A_1 \dots A_l$ is the successor (sequence). We suppose in a set of rules that each head is followed by a unique successor.

Syntactically, we assume:

(i) a set P of rules of the form $A \rhd A_1 \dots A_l$ $(l \geq 0)$ where any two rules with the same head, $A \rhd B_1 \dots B_m$ and $A \rhd C_1 \dots C_n$, have the same successor, and
(ii) a set B_P of all references occurring in the set P.

The interpretation of references is defined as *eager*, *lazy* and a *failure*: Assume a set P of rules. Given a set L, we have inferences to inductively define the predicates *eager* and $lazy_L$ which are mutually exclusive:

$$\text{(ir1)} \quad \frac{A \rhd \text{ is in } P}{eager(A)}$$

$$\text{(ir2)} \quad \frac{\begin{array}{c} A \rhd A_1 \dots A_m \text{ is in } P \quad (m > 0) \\ \text{for all } A_i \ (1 \leq i \leq m),\ eager(A_i) \end{array}}{eager(A)}$$

$$\text{(ir3)}\quad \frac{\begin{array}{l} A \text{ does not occur in the head of any rule} \\ A \text{ is in } L \end{array}}{lazy_L(A)}$$

$$\text{(ir4)}\quad \frac{\begin{array}{l} A \rhd A_1 \ldots A_m \;\text{ is in } P \;\; (m > 0) \\ \text{for all } A_j \;\; (1 \le j \le m),\ eager(A_j) \text{ or } lazy_L(A_j) \\ \text{for some } A_k \ (1 \le k \le m),\ not\ eager(A_k) \end{array}}{lazy_L(A)}$$

Semantically, we say that:

(i) If $eager(A)$, the reference A is eager.
(ii) If $lazy_L(A)$, the reference A is lazy.
(iii) If neither $eager(A)$ nor $lazy_L(A)$, the reference A is a failure.

Example 1. Assume a set P containing: (i) $A \rhd B$, and (ii) $B \rhd A\ C$. Note that neither a rule $A\rhd$ nor a rule $A \rhd B, C$ can be included into the set P, as long as the rule $A \rhd B$ (with the head A) is in P. What set of references may be lazy? To see it, we have exhaustive cases:

(1) A, B, C are failures, unless there is some lazy reference.
(2) C may be lazy, whether or not both of A and B are lazy. Neither A nor B can be lazy, if C still remains to be a failure.
(3) For A and C to be lazy, all the A, B, C are lazy. Similarly for B and C to be lazy, all are lazy.

A mapping $T_P : 2^{B_P} \to 2^{B_P}$ is defined to be

$$T_P(I) = \{A \mid \exists A \rhd A_1 \ldots A_l \in P.\ A_1, \ldots, A_l \in I\}.$$

Note that the mapping T_P is similar to the mapping associated with a logic program ([11]), such that it collects eager references based on the set I of eager references. Such a mapping is often adopted. As easily seen, if $I \subseteq J$, then $T_P(I) \subseteq T_P(J)$, that is, T_P is monotone. In what follows, we have the notation:

$$T_P^n(I) = \begin{cases} I & (n = 0) \\ T_P(T_P^{n-1}(I)) & (n > 0) \end{cases}$$

for a subset $I \subseteq B_P$. The mapping T_P is continuous: For any ω-chain $I_0 \subseteq I_1 \subseteq I_2 \subseteq \ldots$,

$$\cup_{k\in\omega}\ T_P(I_k) = T_P(\cup_{k\in\omega}\ I_k).$$

Thus T_P has the least fixed point, $\cup_{n\in\omega}\ T_P^n(\emptyset)$, which is denoted by $lfp(T_P)$.

The following mapping looks like the one for logic programs with negation (as in [16, 19, 20]), but the present usage is not relevant to the treatment of negations in 3-valued logic. To capture the set of lazy references, we make use of the following mapping S_P. With respect to a subset $K \subseteq B_P$,

$$P[K] = \{A \rhd A_1 \ldots A_m \mid \\ \exists A \rhd A_1 \ldots A_m B_1 \ldots B_n \in P\ (m \ge 0, n \ge 0).\ B_1, \ldots, B_n \in K\}.$$

Note that $P[\emptyset] = P$. A mapping $S_P : 2^{B_P} \to 2^{B_P}$ is defined to be

$$S_P(K) = \cup_{j\in\omega}\ T^j_{P[K]}(\emptyset) = lfp(T_{P[K]}).$$

The set $S_P(K)$ denotes the collection of eager and lazy references based on the set K of lazy references. It follows that $S_P(\emptyset) = lfp(T_{P[\emptyset]}) = lfp(T_P)$. When $J \subseteq K$, $A \in T^i_{P[J]}(\emptyset) \Rightarrow A \in T^i_{P[K]}(\emptyset)$. It is because:

(i) (basis) In case that $i = 0$, it trivially holds.
(ii) (induction step) In case that $i > 0$:

$$\begin{aligned}
&A \in T^i_{P[J]}(\emptyset)\\
&\Rightarrow \exists A \rhd A_1 \ldots A_m \in P[J].\ A_1, \ldots, A_m \in T^{i-1}_{P[J]}(\emptyset)\\
&\Rightarrow \exists A \rhd B_1 \ldots B_n \in P[K] \text{ such that } \{B_1, \ldots, B_n\} \subseteq \{A_1, \ldots, A_m\}
\end{aligned}$$

It follows that $B_1, \ldots, B_n \in T^{i-1}_{P[J]}(\emptyset)$. By induction hypothesis, we can assume that $B_1, \ldots, B_n \in T^{i-1}_{P[J]}(\emptyset) \Rightarrow B_1, \ldots, B_n \in T^{i-1}_{P[K]}(\emptyset)$. Therefore $A \in T^i_{P[K]}(\emptyset)$.

This concludes that $S_P(J) = \cup_{i\in\omega}\ T^i_{P[J]}(\emptyset) \subseteq \cup_{i\in\omega}\ T^i_{P[K]}(\emptyset) = S_P(K)$. That is, the mapping S_P is monotone. By monotonicity of S_P, $S_P(J_i) \subseteq S_P(\cup_{i\in\omega}\ J_i)$ for any ω-chain $J_0 \subseteq J_1 \subseteq J_2 \subseteq \ldots$. Thus $\cup_{i\in\omega}\ S_P(J_i) \subseteq S_P(\cup_{i\in\omega}\ J_i)$. On the other hand, to show the opposite subset relation, we firstly assume that $A \in S_P(\cup_{i\in\omega}\ J_i)$. Then:

$$\begin{aligned}
&A \in S_P(\cup_{i\in\omega}\ J_i)\\
&\Rightarrow \exists j \in \omega.\ A \in T^j_{P[\cup_{i\in\omega}\ J_i]}(\emptyset)\\
&\Rightarrow \exists k \in \omega.\ A \in T^j_{P[J_k]}(\emptyset)\\
&\Rightarrow A \in \cup_{j\in\omega}\ T^j_{P[J_k]}(\emptyset) = S_P(J_k)
\end{aligned}$$

Therefore $S_P(\cup_{i\in\omega}\ J_i) \subseteq S_P(J_k)$ for some $k \in \omega$ such that $S_P(\cup_{i\in\omega}\ J_i) \subseteq \cup_{k\in\omega}\ S_P(J_k)$. That is, S_P is continuous. By means of the definition of $S_P(K)$ with respect to the mapping $T_{P[K]}$, $S_P(K)$ is the least fixed point of $T_{P[K]}$ such that we can see the following lemma.

Lemma 1. (1) For any $A \rhd A_1 \ldots A_m \in P[K]$ $(m \geq 0)$,

$$A_1, \ldots, A_m \in S_P(K) \text{ iff } A \in S_P(K).$$

(2) For any $A \rhd A_1 \ldots A_m \in P$ $(m \geq 0)$,

$$A_1, \ldots, A_m \in S_P(K) \cup K \text{ iff } A \in S_P(K).$$

(3) For any $A \rhd A_1 \ldots A_m \in P$ $(m > 0)$,

$$\begin{gathered}
A_1, \ldots, A_m \in S_P(K) \cup K \text{ and there is at least one } A_i \notin S_P(\emptyset)\\
\text{iff } A \in S_P(K) - S_P(\emptyset).
\end{gathered}$$

Proof. (1) For the rule $A \rhd A_1 \dots A_m \in P[K]$ $(m \geq 0)$:

$$\begin{array}{l} A \in S_P(K) \\ \Leftrightarrow A \in \cup_{i\in\omega} T^i_{P[K]}(\emptyset) \\ \Leftrightarrow A_1, \dots, A_m \in \cup_{i\in\omega} T^i_{P[K]}(\emptyset) \\ \Leftrightarrow A_1, \dots, A_m \in S_P(K) \end{array}$$

(2) For the rule $A \rhd A_1 \dots A_m \in P$ $(m \geq 0)$, we can derive a rule $A \rhd B_1 \dots B_n \in P[K]$ such that $\{B_1 \dots B_n\} \subseteq \{A_1 \dots A_m\}$. The set $\{B_1 \dots B_n\}$ is obtained by removing each A_i of $\{A_1 \dots A_m\}$ for $A_i \in K$. By means of (1), $B_1 \dots B_n \in S_P(K)$ iff $A \in S_P(K)$. Thus

$$A_1, \dots, A_m \in S_P(K) \cup K \text{ iff } A \in S_P(K).$$

(3) By means of (2), $A_1, \dots, A_m \in S_P(K) \cup K$ $(m \geq 0)$ iff $A \in S_P(K)$. There is some $A_i \notin S(\emptyset)$ iff $A \notin S_P(\emptyset)$, by (2) for the case that $K = \emptyset$. It follows that

$$\begin{array}{c} A_1, \dots, A_m \in S_P(K) \cup K \ (m > 0) \text{ and there is at least one } A_i \notin S_P(\emptyset) \\ \text{iff } A \in S_P(K) - S_P(\emptyset). \end{array}$$

3 Lazy Reference Set Related to Fixed Point

In this section, we examine the set of lazy references.

Lemma 2. Assume the set P of rules. A reference A is in $S_P(\emptyset)$ iff it is eager.

Proof. (1) Assume $eager(A)$.

(i) If $eager(A)$ by means of (ir1), then $A\rhd$ is in P such that $A \in S_P(\emptyset)$ (by Lemma 1 (2)).

(ii) If $eager(A)$ by means of (ir2), then a rule $A \rhd A_1 \dots A_m$ is in P and for all A_i $(1 \leq i \leq m)$, the predicates $eager(A_i)$ are supposed. By induction hypothesis for $eager(A_i)$ $(1 \leq i \leq n)$, $A_i \in S_P(\emptyset)$, such that by Lemma 1 (2), $A \in S_P(\emptyset)$. This completes the induction.

(2) Assume that $A \in S_P(\emptyset)$. We prove it by induction on m for the rule $A \rhd A_1 \dots A_m$ $(m \geq 0)$, with respect to $A \in S_P(\emptyset)$.

(i) If $m = 0$, that is, $A\rhd$ is in P, then $eager(A)$ (by the inference (ir1)).

(ii) If $m > 0$ such that $A \rhd A_1 \dots A_m$ is in P, by induction hypothesis of $eager(A_i)$ $(1 \leq i \leq m)$ for $A_i \in S_P(\emptyset)$, we have $eager(A)$ with the inference (ir2). This completes the induction.

Lemma 3. Assume the set P of rules. A reference $A \in B_P$ does not occur in the head of any rule iff $A \in \overline{S_P(B_P)}$.

Proof. (i) Assume that the reference A occurs in the head of some rule such that there is a rule $A \rhd A_1 \dots A_m$ in P $(m \geq 0)$. It follows that $A\rhd$ is in $P[B_P]$. Thus $A \in T_{P[B_P]}(\emptyset) \subseteq S_P(B_P)$.

(ii) On the other hand, assume that $A \in S_P(B_P)$. Then $A \in S_P(B_P) = \cup_{i\in\omega} T^i_{P[B_P]}(\emptyset)$, which demonstrates that A occurs in the head of some rule.

For the lazy reference, we need the superset relation $L \supseteq S_P(L) - S_P(\emptyset)$ for a subset $L \subseteq B_P$. By Lemma 2, a set of lazy references has no common reference with the set $S_P(\emptyset)$ (the set of eager references). Assume a set $M \subseteq \overline{S_P(B_P)} \subseteq \overline{S_P(\emptyset)}$ such that M may be a set of references not occurring in the heads and be designated as lazy. We next investigate a fixed point of the equation $L = (S_P(L) - S_P(\emptyset)) \cup M$ for some $M \subseteq \overline{S_P(B_P)}$ by the following two theorems.

Theorem 1. *The set P of rules is supposedly given, where $L \subseteq \overline{S_P(\emptyset)}$. If $L = \{A \mid lazy_L(A)\}$,*

$$L = (S_P(L) - S_P(\emptyset)) \cup M \text{ for some set } M \subseteq \overline{S_P(B_P)}.$$

Proof. If $L = \emptyset$, then the theorem trivially holds. Assume that $L = \{A \mid lazy_L(A)\} \neq \emptyset$. Suppose $lazy_L(A)$ ($A \in L$ by the assumption). We prove inductively that:

- $A \in L$ occurs in the head of some rule iff $A \in S_P(L) - S_P(\emptyset)$.
- $A \in L$ does not occur in the head of any rule iff $A \in M$ for some $M \subseteq \overline{S_P(B_P)}$.

We see that:

$$\begin{aligned}
&A \text{ occurs in the head of some rule} \\
&\Leftrightarrow \text{there is a rule } A \rhd A_1 \ldots A_m \in P \ (m > 0) \text{ such that} \\
&\qquad\qquad \exists A_i.\ (A_i \text{ is not eager}), \text{ and} \\
&\qquad\qquad \forall A_j.\ (A_j \text{ is eager or lazy}) \\
&\Leftrightarrow \text{there is a rule } A \rhd A_1 \ldots A_m \in P \ (m > 0) \text{ such that:} \\
&\qquad\qquad \exists A_i.(A_i \notin S_P(\emptyset)) \text{ and } \forall A_j.(A_j \in S_P(L) \cup L) \\
&\Leftrightarrow A \in S_P(L) - S_P(\emptyset) \\
&\qquad \text{(by Lemma 1 (3))}
\end{aligned}$$

$A \in L$ does not occur in the head of any rule iff $A \in \overline{S_P(B_P)}$ (Lemma 3) such that $A \in M$ for some $M \subseteq \overline{S_P(B_P)}$. This completes the proof.

Lemma 4. Assume a fixed point L of the equation $L = (S_P(L) - S_P(\emptyset)) \cup M$ for some set $M \subseteq \overline{S_P(B_P)}$. Then

(i) $L \subseteq \overline{S_P(\emptyset)}$.
(ii) $S_P(L) - S_P(\emptyset) \subseteq S_P(\overline{S_P(\emptyset)})$.
(iii) $M \subseteq \overline{S_P(\overline{S_P(\emptyset)})}$.

Proof. (i) $S_P(L) - S_P(\emptyset) \subseteq \overline{S_P(\emptyset)}$. $M \subseteq \overline{S_P(B_P)} \subseteq \overline{S_P(\emptyset)}$. It follows that $L \subseteq \overline{S_P(\emptyset)}$.
(ii) By (i), applying the monotone mapping S_P, $S_P(L) \subseteq S_P(\overline{S_P(\emptyset)})$. Then $S_P(L) - S_P(\emptyset) \subseteq S_P(\overline{S_P(\emptyset)})$.
(iii) Since $S_P(\overline{S_P(\emptyset)}) \subseteq S_P(B_P)$ by monotonicity of the mapping of S_P, $\overline{S_P(B_P)} \subseteq \overline{S_P(\overline{S_P(\emptyset)})}$. On the assumption that $M \subseteq \overline{S_P(B_P)}$, $M \subseteq \overline{S_P(\overline{S_P(\emptyset)})}$.

In Lemma 4, we suppose that a set M is designated as lazy.

Theorem 2. *Assume that a set P of rules is given, such that $L = (S_P(L) - S_P(\emptyset)) \cup M$ where $M \subseteq \overline{S_P(B_P)}$. Then $L = \{A \mid lazy_L(A)\}$.*

Proof. If $L = \emptyset$, the theorem trivially holds. We now suppose that $L \neq \emptyset$.
(1) Take any reference $A \in L$. We prove inductively with the following cases (i) and (ii) that $lazy_L(A)$. (It follows that $L \subseteq \{A \mid lazy_L(A)\}$.)

(i) Assume that $A \in S_P(L) - S_P(\emptyset) \neq \emptyset$.

$A \in S_P(L) - S_P(\emptyset)$
$\Rightarrow$ there is a rule $A \rhd A_1 \ldots A_m$ $(m > 0)$ in P such that:
$A_1, \ldots, A_m \in S_P(L) \cup L$ and at least one A_i is not in $S_P(\emptyset)$
(by Lemma 1 (3))
$\Rightarrow$ there is a rule $A \rhd A_1 \ldots A_m$ $(m > 0)$ in P such that:
$A_1, \ldots, A_m$ are eager or lazy, and at least one A_i is not eager
(by induction hypothesis) : $A_j \in S_P(L) - S_P(\emptyset) \Rightarrow A_j$ is lazy;
$A_j \in S_P(\emptyset) \Rightarrow A_j$ is eager; $A_j \in L - (S_P(L) - S_P(\emptyset)) \Rightarrow A_j$ is lazy
$\Rightarrow A$ is lazy, that is, $lazy_L(A)$

(ii) Assume that $A \in M \subseteq \overline{S_P(B_P)}$. By Lemma 3, A does not occur in the head of any rule. If $A \in L$, then $lazy_L(A)$.
By (i) and (ii), we conclude that $L \subseteq \{A \mid lazy_L(A)\}$.

(2) We next prove that if $lazy_L(A)$ then $A \in L$.

(i) If A occurs in the head of some rule, then there is a rule $A \rhd A_1 \ldots A_m$ such that each A_j is eager or lazy ($A_j \in S_P(\emptyset) \cup L$), and at least one A_i is not eager ($A_i \notin S_P(\emptyset)$). It follows that $A \in S_P(L) - S_P(\emptyset) \subseteq L$.
(ii) If A does not occur in the head of any rule, $A \in L$ because of $lazy_L(A)$.

As the conclusion of (2), $L \supseteq \{A \mid lazy_L(A)\}$, by which we conclude that $L = \{A \mid lazy_L(A)\}$, as well as the proof (1). This completes the proof.

By Theorems 1 and 2, we see that L is a fixed point of the equation:

$$L = (S_P(L) - S_P(\emptyset)) \cup M \text{ for some set } M \subseteq \overline{S_P(B_P)}$$

iff $L = \{A \mid lazy_L(A)\}$. As is seen, there is a least fixed point of the equation. Note that $M \subseteq \overline{S_P(B_P)}$ is not uniquely determined for the equation $L = (S_P(L) - S_P(\emptyset)) \cup M$. In the next section, instead of the equation $L = (S_P(L) - S_P(\emptyset)) \cup M$, we take a superset relation $L \supseteq S_P(L) - S_P(\emptyset)$ without such a set M.

4 Soundness and Completeness of Reference Laziness

We firstly have a soundness theorem of the predicate $lazy_L(A)$ (which states that A is lazy with the set $L \subseteq \overline{S_P(\emptyset)}$), with respect to membership of A in L or in $S_P(L) - S_P(\emptyset)$ with some set L', where

$$S_P(L) - S_P(\emptyset) \subseteq S_P(L') - S_P(\emptyset) \subseteq L' \subseteq \overline{S_P(\emptyset)}.$$

Theorem 3. *Given a set P of rules, assume $lazy_L(A)$, where $L \subseteq \overline{S_P(\emptyset)}$. Then $A \in L$, or there is L' such that $A \in S_P(L) - S_P(\emptyset) \subseteq S_P(L') - S_P(\emptyset) \subseteq L' \subseteq \overline{S_P(\emptyset)}$.*

Proof. Assume that $lazy_L(A)$. (1) We prove inductively that $A \in L$, or $A \in S_P(L) - S_P(\emptyset)$ as follows:

(i) If A doe not occur in the head of any rule, A must be in L because of the predicate $lazy_L(A)$.

(ii) If A occurs in the head of some rule, then:

$$\begin{array}{l} \text{there is a rule } A \rhd A_1 \ldots A_m \in P \ (m > 0) \text{ such that:} \\ \qquad \exists A_i.\ (A_i \text{ is not eager}), \text{ and} \\ \qquad \forall A_j.\ (A_j \text{ is eager or lazy}) \\ \Rightarrow \text{there is a rule } A \rhd A_1 \ldots A_m \in P \ (m > 0) \text{ such that:} \\ \qquad \exists A_i.(A_i \notin S_P(\emptyset)) \text{ and } \forall A_j.(A_j \in S_P(L) \cup L) \\ \Rightarrow A \in S_P(L) - S_P(\emptyset) \end{array}$$

(2) Now we assume the case that $lazy_L(A)$ such that $A \in S_P(L) - S_P(\emptyset)$. With $L_0 = L$ and $L_1 = S_P(L) - S_P(\emptyset)$, we have an ω-chain $L_1 \subseteq L_2 \subseteq \ldots$, owing to monotonicity of S_P,

$$\begin{array}{l} S_P(L_0) - S_P(\emptyset) = L_1 \\ S_P(L_0 \cup L_1) - S_P(\emptyset) = L_2 \\ \cdots\ \cdots \\ \cdots\ \cdots \\ S_P(\cup_{i\in\omega}\ L_i) - S_P(\emptyset) = \cup_{i\geq 1}\ L_i \end{array}$$

where $S_P(\cup_{i\in\omega}\ L_i) = \cup_{i\in\omega}\ S_P(L_i)$ by continuity of S_P. Take $L' = \cup_{i\in\omega}\ L_i \supseteq \cup_{i\geq 1}\ L_i$. Then

$$A \in L_1 \subseteq \cup_{i\geq 1}\ L_i = S_P(\cup_{i\in\omega}\ L_i) - S_P(\emptyset) = S_P(L') - S_P(\emptyset) \subseteq L'.$$

Because $L_i \subseteq \overline{S_P(\emptyset)}$ $(i \in \omega)$ by the construction of L_i, $L' = \cup_{i\in\omega}\ L_i \subseteq \overline{S_P(\emptyset)}$. This completes the proof.

We next have a completeness theorem of the predicate $lazy_L(A)$ (which states that A is lazy with the set $L \subseteq \overline{S_P(\emptyset)}$), with respect to membership of A in $S_P(L) - S_P(\emptyset)$, where

$$S_P(L) - S_P(\emptyset) \subseteq L \subseteq \overline{S_P(\emptyset)}.$$

Theorem 4. *Assume a set P of rules such that $\emptyset \neq S_P(L) - S_P(\emptyset) \subseteq L$ for a set $L \subseteq \overline{S_P(\emptyset)}$. If $A \in S_P(L) - S_P(\emptyset)$, then $lazy_L(A)$.*

Proof. Assume that $A \in S_P(L) - S_P(\emptyset)$. By Lemma 1 (3), there is a rule

$$A \rhd A_1 \ldots A_m \ (m > 0)$$

such that $A_1, \ldots, A_m \in S_P(L) \cup L$ and at least one A_i in in $\overline{S_P(\emptyset)}$. Because $A_1, \ldots, A_m$ are all in $S_P(L) \cup L$ and at least one A_i is in $\overline{S_P(\emptyset)}$, we see the cases for each A_j:

(i) $A_j \in L$
(ii) $A_j \in S_P(\emptyset) \subseteq S_P(L) \Rightarrow eager(A_j)$ (by Lemma 2)
(iii) $A_j \in S_P(L) - S_P(\emptyset) \subseteq S_P(L) \Rightarrow lazy_L(A_j)$ (by induction hypothesis for A_j)

If A_i is in $\overline{S_P(\emptyset)}$, then A_i is in L or $lazy_L(A_i)$ excluding the case (ii). By the inferences (ir3) and (ir4), we can conclude that $lazy_L(A)$.

5 Concluding Remarks

We have dealt with a finite or countably infinite set of rules, where the set of lazy references is represented by means of fixed point approach. Practically only a finite set is needed, where the theoretical considerations are available from static analysis views as in this paper. Given a set of P of rules with a set L of designated lazy references, we have soundness and completeness of reference laziness in the following sense:

(1) (soundness) The predicate $lazy_L(A)$ (which states that the reference A is lazy with the set $L \subseteq \overline{S_P(\emptyset)}$) is sound with respect to membership of A in L or in $S_P(L) - S_P(\emptyset)$, with some set L' such that

$$S_P(L) - S_P(\emptyset) \subseteq S_P(L') - S_P(\emptyset) \subseteq L' \subseteq \overline{S_P(\emptyset)}.$$

(2) (completeness) The predicate $lazy_L(A)$ (which states that A is lazy with the set $L \subseteq \overline{S_P(\emptyset)}$) is complete with respect to membership of A in $S_P(L) - S_P(\emptyset)$, where

$$S_P(L) - S_P(\emptyset) \subseteq L \subseteq \overline{S_P(\emptyset)}.$$

In addition to the soundness, the designation of lazy references may step by step construct some set L' which is relative to the soundness of the predicate $lazy_L(A)$ with respect to membership of A in $S_P(L) - S_P(\emptyset)$.

The set of finite-failure references (the finite-failure set) may be defined. This is similar to finite failure of logic programming ([11]), however, a unique successor (which may be the empty) for each head may be allowable in this case.

We can define the finite-failure set FF_P to be $FF_P = \cup_{d \in \omega} FF_P^d$, where:

$$FF_P^0 = \{A \in B_P \mid A \text{ does not occur in the head of any rule}\} - L,$$
$$FF_P^d = \{A \in B_P \mid \exists A \rhd A_1 \ldots A_m \in P,\ \exists A_i.\ A_i \in FF_P^{d-1}\} - L \ (d > 0).$$

When $L = \emptyset$, regarding the reference as a proposition with the propositional Horn logic, we have

$FF_P = \overline{\cap_{i \in \omega} T_P^i(B_p)}$ (where T_P^i stands for i-times applications to the set B_P).

If we allow the case that there are more than two rules with a head including different successors, which is prohibited in the set of rules of this paper, the rule set conceives the interpretation that the reference A is both eager and lazy. Even if such a case is involved, the properties as in the propositional Horn logic may be of use for the treatments of references. It may be a problem to see a relation between the lazy reference set and the set $\cap_{i\in\omega} T_P^i(B_P)$. How we temporarily have a set L may affect some reasonable considerations about the relation.

References

1. Areces,C. and Blackburn,P., Repairing the interpolation in quantified logic, Annals of Pure and Applied Logic, 123, 287–299, 2003.
2. Brauner,T., Natural deduction for hybrid logics, J. of Logic and Computation, 14, 329–353, 2004.
3. Cervesato,I., Chittaro,L. and Montanari,A., A general modal framework for the event calculus and its skeptical and credulous variants, Proc. of 12th European Conference on Artificial Intelligence, pp.12–16, 1996.
4. Dean,T. and Boddy,M., Reasoning about partially ordered events, Artificial Intelligence, 36, pp.375–399, 1988.
5. Genesereth,M.R. and Nilsson,N.J., Logical Foundations of Artificial Intelligence, Morgan Kaufmann, 1988.
6. Giordano,L., Martelli,A. and Schwind,C., Ramification and causality in a modal action logic, J. of Logic and Computation, 10, pp.625–662, 2000.
7. Harpern,J.Y. and Lakemeyer,G., Multi-agent only knowing, J. of Logic and Computation, 11, pp.41–70, 2001.
8. Hoare,C.A.R., Communicating Sequential Processes, Prentice-Hall, 1985.
9. Kowalski,R.A., Database updates in the event calculus, J. of Logic Programming, 12, 121–146, 1992.
10. Kucera,A. and Esparza,J., A logical viewpoint on process-algebraic quotients, J. of Logic and Computation, 13, pp.863–880, 2003.
11. Lloyd,J.W., Foundations of Logic Programming, 2nd, Extended Edition, Springer-Verlag, 1993.
12. Milner,R., Communication and Concurrency, Prentice-Hall, 1989.
13. Mosses,P.M., Action Semantics, Cambridge University, 1992.
14. Reiter,R., Knowledge in Action, The MIT Press, 2001.
15. Russell,S. and Norvig,P., Artificial Intelligence–A Modern Approach–, Prentice-Hall, 1995.
16. Shepherdson,J.C., Negation in logic programming, In Minker,J. (ed.), Foundations of Deductive Databases and Logic Programming, 19–88, 1987.
17. Winskel,G., The Formal Semantics of Programming Languages, MIT Press, 1993.
18. Yamasaki,S., A denotational semantics and dataflow construction for logic programs, Theoretical Computer Science, 124, pp.71-91, 1994.
19. Yamasaki,S. and Kurose,Y., A sound and complete proof procedure for a general logic program in no-floundering derivations with respect to the 3-valued stable model semantics, Theoretical Computer Science, 266, pp.489–512, 2001.
20. Yamasaki,S., Logic programming with default, weak and strict negations, Theory and Practice of Logic Programming, 6, pp.737-749, 2006.
21. Yamasaki,S. and Sasakura,M., A calculus effectively performing event formation with visualization, Lecture Notes in Computer Science, 4759, pp.287-294, 2008.