Flattening and Implication*

Kouichi Hirata **

Department of Artificial Intelligence,
Kyushu Institute of Technology,
Kawazu 680-4, Iizuka 820-8502, Japan
hirata@ai.kyutech.ac.jp

Abstract. *Flattening* is a method to make a definite clause function-free. For a definite clause C, flattening replaces every occurrence of a term $f(t_1, \cdots, t_n)$ in C with a new variable v and adds an atom $p_f(t_1, \cdots, t_n, v)$ with the associated predicate symbol p_f with f to the body of C. Here, we denote the resulting function-free definite clause from C by $\mathit{flat}(C)$. In this paper, we discuss the relationship between flattening and implication. For a definite program Π and a definite clause D, it is known that if $\mathit{flat}(\Pi) \models \mathit{flat}(D)$ then $\Pi \models D$, where $\mathit{flat}(\Pi)$ is the set of $\mathit{flat}(C)$ for each $C \in \Pi$. First, we show that the converse of the above statement does not hold even if $\Pi = \{C\}$, that is, there exist definite clauses C and D such that $C \models D$ but $\mathit{flat}(C) \not\models \mathit{flat}(D)$. Furthermore, we investigate the conditions of C and D satisfying that $C \models D$ if and only if $\mathit{flat}(C) \models \mathit{flat}(D)$. Then, we show that, if (1) C is not self-resolving and D is not tautological, (2) D is not ambivalent, or (3) C is singly recursive, then the statement holds.

1 Introduction

The purpose of Inductive Logic Programming is to find a hypothesis that explains a given sample. It is a normal setting of Inductive Logic Programming that a hypothesis is a definite clause or a definite program and a sample is the set of (labeled) ground definite clauses. In this setting, the word "explain" is interpreted as either "subsume (denoted by $\succeq$)" or "imply (denoted by $\models$)". In the latter case, note that the problem of whether or not a definite clause C implies another definite clause, called an *implication problem*, is undecidable in general [8]. On the other hand, if C is function-free, then it is obvious that the implication problem is decidable.

Flattening, which has been first introduced in the context of Inductive Logic Programming by Rouveirol [14] (though similar ideas had already been used in other fields), is a method to make a definite clause function-free. For a definite clause C, flattening replaces every occurrence of a term $f(t_1, \cdots, t_n)$ in C

* This paper is published in Proc. of 10th International Conference on Algorithmic Learning Theory (ALT'99), Lecture Notes in Artificial Intelligence 1720, 157–168, 1999 ©Springer-Verlag.

** This work is partially supported by Japan Society for the Promotion of Science, Grants-in-Aid for Encouragement of Young Scientists 11780284.

with a new variable v and adds an atom $p_f(t_1, \cdots, t_n, v)$ with the associated predicate symbol p_f with f to the body of C. Additionally, the unit clause $p_f(x_1, \cdots, x_n, f(x_1, \cdots, x_n)) \leftarrow$ is introduced to the background theory for each function symbol f in C. We denote the resulting function-free definite clause by $\mathit{flat}(C)$ and the set of unit clauses by $\mathit{defs}(C)$.

Rouveirol [14] has investigated the several properties of flattening. Muggleton [9, 11] has dealt with flattening in order to characterize his inverting implication. De Raedt and Džeroski [2] have analyzed their PAC-learnability of jk-clausal theories by transforming possibly infinite Herbrand models into approximately finite models according to flattening. Nienhuys-Cheng and de Wolf [13] have revised the properties of flattening with sophisticated duscussion.

Rouveirol [14] (and Nienhuys-Cheng and de Wolf [13]) has shown that flattening "preserves" subsumption: Let C and D be definite clauses. Then, it holds that:

$$C \succeq D \text{ if and only if } \mathit{flat}(C) \succeq \mathit{flat}(D).$$

Also Rouveirol [14] (and Nienhuys-Cheng and de Wolf [13]) has claimed that flattening "preserves" implication: Let Π be a definite program $\{C_1, \cdots, C_n\}$ and D be a definite clause. We denote $\{\mathit{flat}(C_1), \cdots, \mathit{flat}(C_n)\}$ and $\mathit{defs}(C_1) \cup \cdots \cup \mathit{defs}(C_n)$ by $\mathit{flat}(\Pi)$ and $\mathit{defs}(\Pi)$, respectively. Then, Rouveirol's Theorem is described as follows:

$$\Pi \models D \text{ if and only if } \mathit{flat}(\Pi) \cup \mathit{defs}(\Pi) \models \mathit{flat}(D).$$

As the stronger relationship between flattening and implication than Rouveirol's Theorem, Nienhuys-Cheng and de Wolf [13] have shown the following theorem:

$$\text{If } \mathit{flat}(\Pi) \models \mathit{flat}(D), \text{ then } \Pi \models D.$$

If the converse of this theorem holds, then the several learning techniques for propositional logic such as [1, 3] are directly applied to Inductive Logic Programming. On the other hand, if the converse holds, then the implication problem $\Pi \models D$ is decidable, because $\mathit{flat}(\Pi)$ and $\mathit{flat}(D)$ are function-free. However, it contradicts the undecidability of the implication problem [8, 15] or the satisfiability problem [5]. In this paper, we show that the converse does not hold even if $\Pi = \{C\}$, that is, there exist definite clauses C and D such that:

$$C \models D \text{ but } \mathit{flat}(C) \not\models \mathit{flat}(D).$$

Furthermore, we investigate the conditions of C and D satisfying that $C \models D$ if and only if $\mathit{flat}(C) \models \mathit{flat}(D)$. Gottlob [4] has introduced the concepts of *self-resolving* and *ambivalent* clauses. A definite clause C is *self-resolving* if C resolves with a copy of C, and *ambivalent* if there exists an atom in the body of C with the predicate symbol same as one of the head of C. As the corollary of Gottlob's results [4], we show that, if C is not self-resolving and D is not tautological, or D is not ambivalent, then the statement holds. Furthermore, note that the C in the counterexample stated above is given as a *doubly* recursive definite clause,

that is, the body of C contains two atoms that are unifiable with the head of a variant of C. Then, we show that, if C is *singly* recursive, that is, the body of C contains at most one atom that is unifiable with the head of a variant of C, then the statement also holds.

2 Preliminaries

A *literal* is an atom or the negation of an atom. A *positive literal* is an atom and a *negative literal* is the negation of an atom. A *clause* is a finite set of literals. A *unit clause* is a clause containing one positive literal. A *definite clause* is a clause containing one positive literal. A set of definite clauses are called a *definite program*. Conventionally, a definite clause is represented as $A \leftarrow A_1, \cdots, A_m$, where A and A_i $(1 \leq i \leq m)$ are atoms.

Let C be a definite clause $A \leftarrow A_1, \cdots, A_m$. Then, the atom A is called a *head* of C and denoted by $head(C)$, and the sequence $A_1, \cdots, A_m$ of atoms is called a *body* of C and denoted by $body(C)$.

Let C and D be definite clauses. We say that C *subsumes* D, denoted by $C \succeq D$, if there exists a substitution θ such that $C\theta \subseteq D$, i.e., every literal in $C\theta$ also appears in D. Also we say that C *implies* D or D is *a logical consequence of* C, denoted by $C \models D$, if every model of C is also a model of D. C is *logically equivalent to* D, denoted by $C \equiv D$, if $C \models D$ and $D \models C$. For definite programs Π and Σ, $\Pi \succeq D$, $\Pi \succeq \Sigma$, $\Pi \models D$ and $\Pi \models \Sigma$ are defined similarly.

Let C and D be two clauses $\{L_1, \cdots, L_i, \cdots, L_l\}$ and $\{M_1, \cdots, M_j, \cdots, M_m\}$ which have no variables in common. If the substitution θ is an mgu for the set $\{L_i, \neg M_j\}$, then the clause $((C - \{L_i\}) \cup (D - \{\neg M_j\}))\theta$ is called a (*binary*) *resolvent* of C and D. All of the resolvents of C and D are denoted by $Res(C, D)$.

Let Π be a definite program and C be a definite clause. An *SLD-derivation* of C from Π is a sequence $(R_1, C_0, \theta_1), \ldots, (R_k, C_{k-1}, \theta_k)$ such that $R_0 \in \Pi$, $R_k = C$, C_{i-1} is a variant of an element of Π, $R_i \in Res(R_{i-1}, C_{i-1})$, and θ_i is an mgu of the selected literals of R_{i-1} and C_{i-1} for each $1 \leq i \leq k$. If an SLD-derivation of C from Π exists, we write $\Pi \vdash C$. In particular, $\{C\} \vdash D$ is denoted by $C \vdash D$.

Theorem 1 (Subsumption Theorem [13]). *Let Π be a definite program and D be a definite clause. Then, $\Pi \models D$ if and only if there exists a definite clause E such that $\Pi \vdash E$ and $E \succeq D$.*

For a definite clause C, the *lth self-resolving closure* of C, denoted by $\mathcal{S}^l(C)$, is defined inductively as follows:

1. $\mathcal{S}^0(C) = \{C\}$,
2. $\mathcal{S}^l(C) = \mathcal{S}^{l-1}(C) \cup \{R \in Res(C, D) \mid D \in \mathcal{S}^{l-1}(C)\}$ $(l \geq 1)$.

Here, the logically equivalent clauses are regarded as identical. Note that $C \vdash D$ if and only if $D \in \mathcal{S}^l(C)$ for some $l \geq 0$. Then:

Corollary 1 (Implication between Definite Clauses [12]). *Let C and D be definite clauses. Then, $C \models D$ if and only if there exists a definite clause E such that $E \in \mathcal{S}^l(C)$ and $E \succeq D$ for some $l \geq 0$.*

For each n-ary function symbol f, the associated $(n+1)$-ary predicate symbol p_f, called a *flattened predicate symbol* (on f), is introduced uniquely in the process of flattening. Also we call a definite clause C or a definite program Π *regular* if C or Π contains no flattened predicate symbols.

Let C be a definite clause, t be a term appearing in C and v be a variable not appearing in C. Then, $C|_t^v$ denotes the definite clause obtained from C by replacing all occurrences of t in C with v.

There exist several variants (but equivalent) of the definition of flattening:

1. Do we introduce an equality theory [9, 14] or not [2, 13]?
2. Do we transform a constant symbol to an atom with an unary predicate symbol [2, 14] or not [13]?

As the definition of flattening, we adopt the definition similar as De Raedt and Džeroski [2] that does not introduce an equality theory and does not transform a constant symbol.

Let C be a definite clause. Then, the *flattened clause* $\mathit{flat}(C)$ of C is defined as follows:

$$\mathit{flat}(C) = \begin{cases} C & \text{if } C \text{ is function-free,} \\ \mathit{flat}(C') & \text{if } t = f(t_1, \cdots, t_n)(n \geq 1) \text{ appears in } C, \end{cases}$$

where $C' = C|_t^v \cup \{\neg p_f(t_1, \cdots, t_n, v)\}$ and each t_i $(1 \leq i \leq n)$ is a variable or a constant. Also $\mathit{defs}(C)$ is the set $\{p_f(x_1, \cdots, x_n, f(x_1, \cdots, x_n)) \leftarrow \mid f(t_1, \cdots, t_n) \text{ appears in } C\}$ of unit clauses. Furthermore, the number of calls of flat that is necessary to obtain the function-free clause $\mathit{flat}(C)$ of C is called a *rank* of C and denoted by $\mathit{rank}(C)$.

For a definite program $\Pi = \{C_1, \cdots, C_n\}$, we define $\mathit{flat}(\Pi)$ and $\mathit{defs}(\Pi)$ as follows:

$$\mathit{flat}(\Pi) = \{\mathit{flat}(C_1), \cdots, \mathit{flat}(C_n)\},$$
$$\mathit{defs}(\Pi) = \mathit{defs}(C_1) \cup \cdots \cup \mathit{defs}(C_n).$$

3 Flattening and Implication

As the relationship between flattening and subsumption, Rouveirol [14] (and Nienhuys-Cheng and de Wolf [13]) has shown the following theorem:

Theorem 2 (Rouveirol [14], Nienhuys-Cheng & de Wolf [13]). *Let C and D be regular definite clauses. Then, $C \succeq D$ if and only if $\mathit{flat}(C) \succeq \mathit{flat}(D)$.*

Furthermore, Rouveirol [14] (and Nienhuys-Cheng and de Wolf [13]) has proposed the following relationship between flattening and implication. Let Π be a regular definite program and D be a regular definite clause. Then, Rouveirol's Theorem is described as follows:

Theorem 3 (Rouveirol [14], Nienhuys-Cheng & de Wolf [13]). *Let Π be a regular definite program and D be a regular definite clause. Then, $\Pi \models D$ if and only if $\mathit{flat}(\Pi) \cup \mathit{defs}(\Pi) \models \mathit{flat}(D)$.*

In Appendix, we discuss the proof of Rouveirol's Theorem.

Furthermore, Nienhuys-Cheng and de Wolf [13] have shown the following theorem, which is a stronger relationship between flattening and implication than Rouveirol's Theorem:

Theorem 4 (Nienhuys-Cheng & de Wolf [13]). *Let Π be a regular definite program and D be a regular definite clause. If $\mathit{flat}(\Pi) \models \mathit{flat}(D)$, then $\Pi \models D$.*

On the other hand, the converse of Theorem 4 does not hold even if $\Pi = \{C\}$:

Theorem 5. *There exist regular definite clauses C and D such that*

$$C \models D \text{ but } \mathit{flat}(C) \not\models \mathit{flat}(D).$$

Proof. Let C and D be the following regular definite clauses:

$$C = p(f(x_1), f(x_2)) \leftarrow p(x_1, x_3), p(x_3, x_2),$$
$$D = p(f(f(x_1)), f(f(x_2))) \leftarrow p(x_1, x_3), p(x_3, x_4), p(x_4, x_5), p(x_5, x_2).$$

By resolving C to a copy of C itself twice, it holds that $C \vdash D$ as Figure 1. Hence, it holds that $C \models D$.

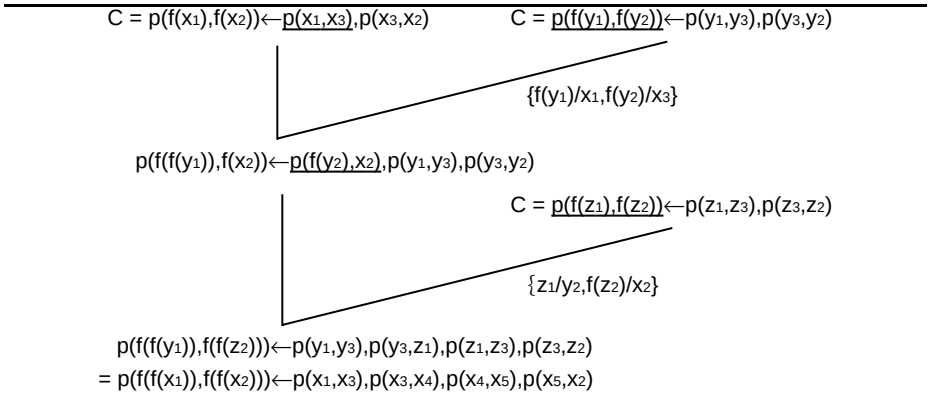

Fig. 1. The SLD-derivation of D from C

On the other hand, $\mathit{flat}(C)$ and $\mathit{flat}(D)$ are constructed as follows:

$$\mathit{flat}(C) = p(x_1, x_2) \leftarrow p(x_3, x_4), p(x_4, x_5), p_f(x_3, x_1), p_f(x_5, x_2),$$
$$\mathit{flat}(D) = p(x_1, x_2) \leftarrow p(x_3, x_4), p(x_4, x_5), p(x_5, x_6), p(x_6, x_7),$$
$$p_f(x_3, x_8), p_f(x_8, x_1), p_f(x_7, x_9), p_f(x_9, x_2).$$

$$
\begin{aligned}
\mathcal{S}^1(\mathit{flat}(C)) &= \{\mathit{flat}(C)\} \\
&\cup \left\{ \begin{array}{l} p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_6,x_7), \\ \qquad\qquad p_f(x_5,x_8), p_f(x_8,x_1), p_f(x_4,x_2), p_f(x_7,x_3) \\ p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_6,x_7), \\ \qquad\qquad p_f(x_3,x_1), p_f(x_7,x_8), p_f(x_8,x_2), p_f(x_5,x_4) \end{array} \right\}, \\
\mathcal{S}^2(\mathit{flat}(C)) &= \mathcal{S}^1(\mathit{flat}(C)) \\
&\cup \left\{ \begin{array}{l} p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_4,x_5), p(x_6,x_7), p(x_7,x_8), \\ \qquad\qquad p_f(x_3,x_9), p_f(x_9,x_1), p_f(x_8,x_{10}), p_f(x_{10},x_2), \\ \qquad\qquad p_f(x_5,x_{11}), p_f(x_6,x_{11}) \\ p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_7,x_8), p(x_8,x_9), \\ \qquad\qquad p_f(x_7,x_{10}), p_f(x_{10},x_{11}), p_f(x_{11},x_1), p_f(x_4,x_2), \\ \qquad\qquad p_f(x_6,x_3), p_f(x_9,x_5) \\ p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_7,x_8), p(x_8,x_9), \\ \qquad\qquad p_f(x_5,x_{10}), p_f(x_{10},x_1), p_f(x_4,x_2), p_f(x_7,x_6), \\ \qquad\qquad p_f(x_9,x_{11}), p_f(x_{11},x_3) \\ p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_7,x_8), p(x_8,x_9), \\ \qquad\qquad p_f(x_3,x_1), p_f(x_6,x_{10}), p_f(x_{10},x_2), p_f(x_7,x_{11}), \\ \qquad\qquad p_f(x_{11},x_4), p_f(x_9,x_5) \\ p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_7,x_8), p(x_8,x_9), \\ \qquad\qquad p_f(x_3,x_1), p_f(x_9,x_{10}), p_f(x_{10},x_{11}), p_f(x_{11},x_2), \\ \qquad\qquad p_f(x_5,x_4), p_f(x_7,x_6) \end{array} \right\}.
\end{aligned}
$$

Fig. 2. The first and second self-resolving closures of $\mathit{flat}(C)$

The first and second self-resolving closures of $\mathit{flat}(C)$ are constructed as Figure 2. Then, there exists no definite clause $E \in \mathcal{S}^2(\mathit{flat}(C))$ such that $E \succeq \mathit{flat}(D)$. By paying our attention to the number of atoms with the predicate p_f and its relation, it holds that $\mathit{flat}(C) \models \mathit{flat}(D)$ if and only if there exists a definite clause $E \in \mathcal{S}^2(\mathit{flat}(C))$ such that $E \succeq \mathit{flat}(D)$ by Corollary 1. Hence, we can conclude that there exists no definite clause $E \in \mathcal{S}^2(\mathit{flat}(C))$ such that $E \succeq \mathit{flat}(D)$, so it holds that $\mathit{flat}(C) \not\models \mathit{flat}(D)$. □

Note that, for the definite clauses C and D given in Theorem 5, it holds that $\{\mathit{flat}(C)\} \cup \mathit{defs}(C) \vdash \mathit{flat}(D)$ as Figure 3, so it holds that $\{\mathit{flat}(C)\} \cup \mathit{defs}(C) \models \mathit{flat}(D)$. Hence, $\mathit{defs}(C)$ is necessary for not only "unflattening" but also unifying with two variables

4 Improvement

In this section, we investigate the conditions of definite clauses C and D satisfying that $C \models D$ if and only if $\mathit{flat}(C) \models \mathit{flat}(D)$.

First, we give the following lemma by Gottlob [4]. A definite clause C is *self-resolving* if C resolves with a copy of C. A definite clause C is *ambivalent* if there exists an atom in $\mathit{body}(C)$ with the predicate symbol same as one of $\mathit{head}(C)$. Then:

Lemma 1 (Gottlob [4]). *Let C and D be definite clauses.*

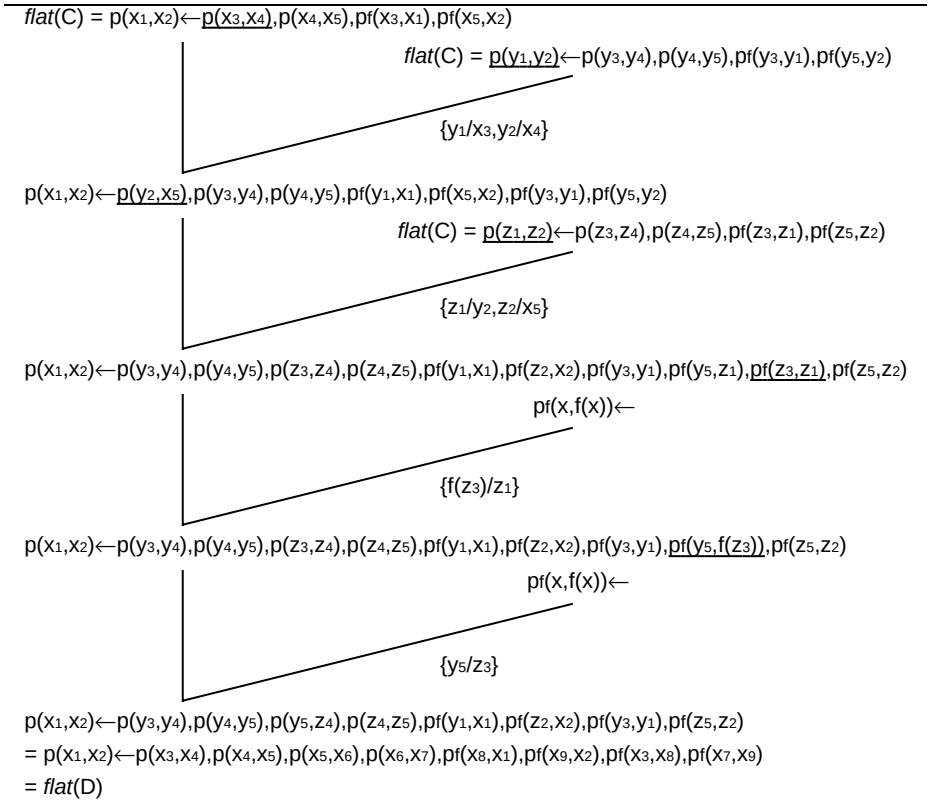

Fig. 3. The SLD-derivation of $\mathit{flat}(D)$ from $\{\mathit{flat}(C)\} \cup \mathit{defs}(C)$ in Theorem 5

1. *Suppose that C is not self-resolving and D is not tautological. Then, $C \models D$ if and only if $C \succeq D$.*
2. *Suppose that D is not ambivalent. Then, $C \models D$ if and only if $C \succeq D$.*

By incorporating Lemma 1 with the previous theorems, we obtain the following corollary:

Corollary 2. *Let C and D be regular definite clauses.*

1. *Suppose that C is not self-resolving and D is not tautological. Then, $C \models D$ if and only if $\mathit{flat}(C) \models \mathit{flat}(D)$.*
2. *Suppose that D is not ambivalent. Then, $C \models D$ if and only if $\mathit{flat}(C) \models \mathit{flat}(D)$.*

Proof. 1. By Lemma 1, $C \models D$ if and only if $C \succeq D$. By Theorem 2, $C \succeq D$ if and only if $\mathit{flat}(C) \succeq \mathit{flat}(D)$. By the definition of $\succeq$ and $\models$, if $\mathit{flat}(C) \succeq \mathit{flat}(D)$ then $\mathit{flat}(C) \models \mathit{flat}(D)$. So it holds that if $C \models D$ then $\mathit{flat}(C) \models \mathit{flat}(D)$. Hence, the statement holds by Theorem 4.

2. By the definition of ambivalence, the predicate symbol of the head of D is different from all of the predicate symbols appearing in the body of D. This condition is preserved in $\mathit{flat}(D)$, because the flattened predicate symbols, which does not appear in D, are introduced only in the body of D. Hence, $\mathit{flat}(D)$ is not ambivalent. By Lemma 1 and Theorem 2, the statement is obvious. □

In Theorem 5, C is given as a doubly recursive definite clause, that is, $\mathit{body}(C)$ contains two atoms that are unifiable with $\mathit{head}(C')$, where C' is a variant of C. In the remainder of this section, we restrict the form of C to *singly recursive*. Here, a definite clause C is *singly recursive* if $\mathit{body}(C)$ contains at most one atom that is unifiable with $\mathit{head}(C')$, where C' is a variant of C.

Lemma 2 (Gottlob [4]). *If $C \models D$, then $\mathit{head}(C) \succeq \mathit{head}(D)$ and $\mathit{body}(C) \succeq \mathit{body}(D)$.*

Let C be a singly recursive definite clause. It is obvious that $|\mathcal{S}^l(C)| \leq l+1$ and $|\mathcal{S}^{l+1}(C) - \mathcal{S}^l(C)| \leq 1$ for each $l \geq 0$. Then, the *lth self-resolvent* C_l of C is defined inductively as follows:

1. $C_0 = C$,
2. $C_{l+1} = \begin{cases} D \text{ if } \mathcal{S}^{l+1}(C) - \mathcal{S}^l(C) = \{D\}, \\ C \text{ otherwise.} \end{cases}$

Lemma 3. *Let C be a singly recursive regular definite clause with function symbols. Suppose that C contains a term $t = f(t_1, \cdots, t_n)$, where each t_i is either a variable or a constant. Also let C' be a definite clause $C|_t^v \cup \{\neg p_f(t_1, \cdots, t_n, v)\}$. Then, it holds that $\mathit{flat}(C_l) \equiv \mathit{flat}(C'_l)$ for each $l \geq 0$.*

Proof. We show the statement by induction on l. If $l = 0$, then the statement is obvious, since $C_0 = C$, $C'_0 = C'$ and $\mathit{flat}(C) = \mathit{flat}(C')$.

Suppose that the statement holds for $l \leq k$. It is sufficient to show the case that C is of the form $p(\overline{t}) \leftarrow p(\overline{s})$. Consider C_{k+1} and C'_{k+1}. By the definition of the $(k+1)$th self-resolvent, C_{k+1} is a resolvent of C and C_k, and C'_{k+1} is a resolvent of C' and C'_k. Then, we can suppose that C_{k+1} is of the form $(\mathit{head}(C) \leftarrow \mathit{body}(C_k)\mu)\theta$, where θ is an mgu of $\mathit{head}(C_k)\mu$ and $\mathit{body}(C)$ and μ is a renaming substitution. Hence, C'_{k+1} is of the form $(\mathit{head}(C') \leftarrow \mathit{body}(C'_k)\mu', p_f(t_1, \cdots, t_n, v))\theta'$, where θ' is a substitution obtained from θ by replacing the binding $t\mu/x$ in θ with v/x, and μ' is a renaming substitution by adding the binding u/v (u is a new variable) to μ. By induction hypothesis, it holds that $\mathit{flat}(C_k) \equiv \mathit{flat}(C'_k)$ and $\mathit{flat}(C) \equiv \mathit{flat}(C')$. By the forms of C_{k+1} and C'_{k+1}, it holds that $\mathit{flat}(C_{k+1}) \equiv \mathit{flat}(C'_{k+1})$. □

Lemma 4. *For a singly recursive definite clause C, it holds that $\mathit{flat}(C_l) \equiv (\mathit{flat}(C))_l$ for each $l \geq 0$.*

Proof. We show the statement by induction on $\mathit{rank}(C)$. If $\mathit{rank}(C) = 0$, then the statement is obvious, because $\mathit{flat}(C_l) = C_l$ and $\mathit{flat}(C) = C$ for each $l \geq 0$.

Suppose that the statement holds for C such that $\mathit{rank}(C) \leq k$. Let C be a singly recursive definite clause such that $\mathit{rank}(C) = k+1$. Since C contains

some function symbols, suppose that C contains the term $t = f(t_1, \cdots, t_n)$, where each t_i is either a variable or a constant. Let C' be a definite clause $C|_t^v \cup \{\neg p_f(t_1, \cdots, t_n, v)\}$. Then, it holds that $\mathit{flat}(C') \equiv \mathit{flat}(C)$ and $\mathit{rank}(C') = k$. By Lemma 3, it holds that $\mathit{flat}(C'_l) \equiv \mathit{flat}(C_l)$ for each $l \geq 0$. By induction hypothesis, it holds that $\mathit{flat}(C_l) \equiv \mathit{flat}(C'_l) \equiv (\mathit{flat}(C'))_l \equiv (\mathit{flat}(C))_l$ for each $l \geq 0$. Hence, the statement holds for $\mathit{rank}(C) = k + 1$. □

Lemma 5. *For a singly recursive definite clause C, it holds that $\mathit{flat}(C) \models \mathit{flat}(C_l)$ for each $l \geq 0$.*

Proof. We show the statement by induction on l. If $l = 0$, then $C_0 = C$, so the statement is obvious.

Suppose that the statement holds for $l \leq k$. Since C_{k+1} is a resolvent of C and C_k and by Lemma 4, $\mathit{flat}(C_{k+1})$ is a resolvent of $\mathit{flat}(C)$ and $\mathit{flat}(C_k)$. By the soundness of SLD-resolution (*cf.* [7, 13]), it holds that $\{\mathit{flat}(C), \mathit{flat}(C_k)\} \models \mathit{flat}(C_{k+1})$. By induction hypothesis, it holds that $\mathit{flat}(C) \models \mathit{flat}(C_k)$. Hence, it holds that $\mathit{flat}(C) \models \mathit{flat}(C_{k+1})$, so the statement holds for $l = k + 1$. □

Theorem 6. *Let C be a singly recursive regular definite clause and D be a regular definite clause. Then, $C \models D$ if and only if $\mathit{flat}(C) \models \mathit{flat}(D)$.*

Proof. By Theorem 4, it is sufficient to show the only-if direction. We show it by induction of $\mathit{rank}(D)$. If $\mathit{rank}(D) = 0$, that is, D is function-free, then so is C by Lemma 2. Then, $\mathit{flat}(C) = C$ and $\mathit{flat}(D) = D$, so the statement is obvious.

Suppose that the statement holds for D such that $\mathit{rank}(D) \leq k$. Let D be a regular definite clause such that $\mathit{rank}(D) = k + 1$. Since D contains some function symbols, suppose that D contains a term $t = f(t_1, \cdots, t_n)$, where each t_i $(1 \leq i \leq n)$ is a variable or a constant. Also let D' be a definite clause $D|_t^v \cup \{\neg p_f(t_1, \cdots, t_n, v)\}$. Then, $\mathit{rank}(D) = k$ and $\mathit{flat}(D') \equiv \mathit{flat}(D)$. Suppose that $C \models D$. Then, by Corollary 1 and the definition of the lth self-resolvent, there exists an index $l \geq 0$ such that $C_l \succeq D$.

As similar as the proof of Lemma 19.6 in [13], we can construct the definite clause C' from C_l such that $C' \succeq D'$ and $\mathit{flat}(C_l) \equiv \mathit{flat}(C')$ as follows: Suppose that $C_l\theta \subseteq D'$. Let $\{s_1, \cdots, s_m\}$ be the set of distinct terms occurring in C_l such that $s_i\theta = t$. If s_i is a variable, then replace the binding t/s_i with v/s_i. If s_i is of the form $f(r_1, \cdots, r_n)$, in which case the r_j are variables or constants, then replace all occurrences of s_i in C_l with a new variable v_i, add $\neg p_f(r_1, \cdots, r_n, v_i)$ in C_l, and add the binding v/v_i to θ. We call the definite clause resulting from these m adjustments C'. Finally, replace all occurrences of t in bindings in θ with y, and call the resulting substitution θ'. Then, it holds that $C'\theta' \subseteq D'$, so $C' \succeq D'$. Hence, $C' \models D'$. Furthermore, by the construction of C', it holds that $\mathit{flat}(C_l) \equiv \mathit{flat}(C')$.

By induction hypothesis, it holds that $\mathit{flat}(C') \models \mathit{flat}(D')$, so $\mathit{flat}(C_l) \models \mathit{flat}(D)$. By Lemma 5, it holds that $\mathit{flat}(C) \models \mathit{flat}(D)$. Hence, the statement holds for $\mathit{rank}(D) = k + 1$. □

5 Conclusion

In this paper, we have investigated the relationship between flattening and implication [13, 14]. Let Π be a regular definite program and C and D be definite clauses. As the stronger relationship between flattening and implication than Rouveirol's Theorem [14], Nienhuys-Cheng and de Wolf [13] have shown the following theorem:

$$\text{If } \mathit{flat}(\Pi) \models \mathit{flat}(D), \text{ then } \Pi \models D.$$

In this paper, we have shown that there exist definite clauses C and D such that:

$$C \models D \text{ but } \mathit{flat}(C) \not\models \mathit{flat}(D).$$

Furthermore, we have shown that if C and D satisfy one of the following conditions, then it holds that $C \models D$ if and only if $\mathit{flat}(C) \models \mathit{flat}(D)$:

1. C is not self-resolving and D is not tautological,
2. D is not ambivalent,
3. C is singly recursive.

Note that the class of definite clauses that flattening preserves implication is corresponding to the class that the implication problem is decidable [4, 6, 7], and the class of definite clauses that flattening does not preserve implication in the above sense is corresponding to the class that the implication problem is undecidable [8, 15]. It is a future work to investigate the relationship between the classes of definite clauses that flattening preserves implication and that implication is decidable.

Acknowledgment

The author would thank to anonymous referees for valuable comments.

References

1. Angluin, D., Frazier, M. and Pitt, L.: *Learning conjunctions of Horn clauses*, Machine Learning **9**, 147–164, 1992.
2. De Raedt, L. and Džeroski, S.: *First-order jk-clausal theories are PAC-learnable*, Artificial Intelligence **90**, 375–392, 1994.
3. Frazier, M. and Pitt, L.: *Learning from entailment: An application to propositional Horn sentences*, Proc. 10th International Conference on Machine Learning, 120–127, 1993.
4. Gottlob, G.: *Subsumption and implication*, Information Processing Letters **24**, 109–111, 1987.
5. Hanschke, P. and Würtz, J.: *Satisfiability of the smallest binary program*, Information Processing Letters **45**, 237–241, 1993.
6. Leitsch, A.: *Implication algorithms for classes of Horn clauses*, Statistik, Informatik und Ökonomie, Springer, 172–189, 1988.

7. Leitsch, A.: *The resolution calculus*, Springer-Verlag, 1997.
8. Marcinkowski, J. and Pacholski, L.: *Undecidability of the Horn-clause implication problem*, Proc. 33rd Annual IEEE Symposium on Foundations of Computer Science, 354–362, 1992.
9. Muggleton, S.: *Inverting implication*, Proc. 2nd International Workshop on Inductive Logic Programming, ICOT Technical Memorandum TM-1182, 1992.
10. Muggleton, S. (ed.): *Inductive logic programming*, Academic Press, 1992.
11. Muggleton, S.: *Inverse entailment and Progol*, New Generation Computing **13**, 245–286, 1995.
12. Muggleton, S. and Page Jr., C., D.: *Self-saturation of definite clauses*, Proc. 4th International Workshop on Inductive Logic Programming, 162–174, 1994.
13. Nienhuys-Cheng, S.-H. and de Wolf, R.: *Foundations of inductive logic programming*, Lecture Notes in Artificial Intelligence **1228**, 1997.
14. Rouveirol, C.: *Extensions of inversion of resolution applied to theory completion*, in [10], 63–92.
15. Schmidt-Schauss, M.: *Implication of clauses is undecidable*, Theoretical Computer Science **59**, 287–296, 1988.

Appendix: Rouveirol's Theorem

Rouveirol's original proof is insufficient for Rouveirol's Theorem. On the other hand, Nienhuys-Cheng and de Wolf [13] have shown Rouveirol's Theorem as the concequence of Theorem 4 and the following lemma:

Lemma 6. *Let Π be a regular definite program. Then, $\mathit{flat}(\Pi) \cup \mathit{defs}(\Pi) \models \Pi$.*

On the other hand, we obtain the following theorem:

Theorem 7. *There exist regular definite clauses C and D such that*

$$C \vdash D \text{ but } \{\mathit{flat}(C)\} \cup \mathit{defs}(C) \not\vdash \mathit{flat}(D).$$

Proof. Let C and D be the following regular definite clauses:

$$\begin{aligned} C &= p(f(x_1, x_3), f(x_3, x_2)) \leftarrow p(x_1, x_3), p(x_3, x_2), \\ D &= p(f(f(x_1, x_2), f(x_2, x_3)), f(f(x_2, x_3), f(x_3, x_4))) \\ &\qquad \leftarrow p(x_1, x_2), p(x_2, x_3), p(x_2, x_3), p(x_3, x_4). \end{aligned}$$

By resolving C with C itself twice, we can show that $C \vdash D$.

On the other hand, $\mathit{flat}(C)$ and $\mathit{flat}(D)$ are constructed as follows:

$$\begin{aligned} \mathit{flat}(C) &= p(x_1, x_2) \leftarrow p(x_3, x_4), p(x_4, x_5), p_f(x_3, x_4, x_1), p_f(x_4, x_5, x_2), \\ \mathit{flat}(D) &= p(x_1, x_2) \leftarrow p(x_3, x_4), p(x_4, x_5), p(x_4, x_5), p(x_5, x_6), \\ &\qquad p_f(x_3, x_4, x_7), p_f(x_4, x_5, x_8), p_f(x_5, x_6, x_9), \\ &\qquad p_f(x_7, x_8, x_1), p_f(x_8, x_9, x_2). \end{aligned}$$

Also $\mathit{defs}(C) = \{p_f(x, y, f(x, y)) \leftarrow\}$.

The first and second self-resolving closures of $\mathit{flat}(C)$ are constructed as Figure 4. By paying our attention to the number of atoms with the predicate p_f and its relation, it holds that $\{\mathit{flat}(C)\} \cup \mathit{defs}(C) \vdash \mathit{flat}(D)$ if and only if there exists

$$
\begin{array}{l}
\mathcal{S}^1(\mathit{flat}(C)) = \{\mathit{flat}(C)\} \\
\cup \left\{ \begin{array}{l}
p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_6,x_7), \\
\qquad\qquad p_f(x_5,x_6,x_8), p_f(x_8,x_3,x_1), p_f(x_3,x_4,x_2), p_f(x_6,x_7,x_3) \\
p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_6,x_7), \\
\qquad\qquad p_f(x_3,x_4,x_1), p_f(x_6,x_7,x_8), p_f(x_4,x_8,x_2), p_f(x_5,x_6,x_4)
\end{array} \right\}, \\
\mathcal{S}^2(\mathit{flat}(C)) = \mathcal{S}^1(\mathit{flat}(C)) \\
\cup \left\{ \begin{array}{l}
E_1 : p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_4,x_5), p(x_6,x_7), p(x_7,x_8), \\
\qquad\qquad p_f(x_3,x_4,x_9), p_f(x_9,x_{10},x_1), p_f(x_7,x_8,x_{11}), p_f(x_{10},x_{11},x_2), \\
\qquad\qquad p_f(x_4,x_5,x_{10}), p_f(x_6,x_7,x_{10}) \\
E_2 : p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_4,x_5), p(x_6,x_7), p(x_7,x_8), \\
\qquad\qquad p_f(x_3,x_4,x_{10}), p_f(x_9,x_{10},x_1), p_f(x_4,x_5,x_{11}), p_f(x_{10},x_{11},x_2), \\
\qquad\qquad p_f(x_7,x_8,x_{10}), p_f(x_6,x_7,x_9) \\
\quad p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_7,x_8), p(x_8,x_9), \\
\qquad\qquad p_f(x_8,x_3,x_1), p_f(x_3,x_4,x_2), p_f(x_{11},x_5,x_{10}), p_f(x_5,x_6,x_3), \\
\qquad\qquad p_f(x_7,x_8,x_{11}), p_f(x_8,x_9,x_5) \\
\quad p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_7,x_8), p(x_8,x_9), \\
\qquad\qquad p_f(x_{10},x_3,x_1), p_f(x_3,x_4,x_2), p_f(x_5,x_6,x_{10}), p_f(x_6,x_{11},x_3), \\
\qquad\qquad p_f(x_3,x_8,x_6), p_f(x_8,x_9,x_{11}) \\
\quad p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_7,x_8), p(x_8,x_9), \\
\qquad\qquad p_f(x_3,x_4,x_1), p_f(x_4,x_{10},x_2), p_f(x_{11},x_5,x_4), p_f(x_5,x_6,x_{10}), \\
\qquad\qquad p_f(x_7,x_8,x_{11}), p_f(x_8,x_9,x_5) \\
\quad p(x_1,x_2) \leftarrow p(x_3,x_4), p(x_5,x_6), p(x_7,x_8), p(x_8,x_9), \\
\qquad\qquad p_f(x_3,x_4,x_1), p_f(x_4,x_{10},x_2), p_f(x_5,x_6,x_4), p_f(x_6,x_{11},x_{10}), \\
\qquad\qquad p_f(x_7,x_8,x_6), p_f(x_8,x_9,x_{11})
\end{array} \right\}.
\end{array}
$$

Fig. 4. The first and second self-resolving closures of $\mathit{flat}(C)$

a definite clause $E \in \mathcal{S}^2(\mathit{flat}(C))$ such that $\mathit{flat}(D)$ is obtained by resolving E with $p_f(x, y, f(x, y)) \leftarrow$ some times. Note that we cannot obtain the above E from each element in $\mathcal{S}^2(\mathit{flat}(C))$ except E_1 and E_2. Furthermore, the resolvent of E_i $(i = 1, 2)$ with $p_f(x, y, f(x, y)) \leftarrow$ twice, where the selected atoms in E_i are atoms of which the third argument's term is x_{10}, contains a term with f. Hence, it holds that $\{\mathit{flat}(C)\} \cup \mathit{defs}(C) \not\vdash \mathit{flat}(D)$. □

Hence, we cannot conclude Rouveirol's Theorem from Theorem 4 and Lemma 6.

Noet that the definite clauses C and D in Theorem 7 are *not* a counterexample of the if-direction of Rouveirol's Theorem, because E_1 and E_2 subsume $\mathit{flat}(D)$ by the following substitutions σ_1 and σ_2:

$$
\begin{array}{l}
\sigma_1 = \{x_4/x_6, x_5/x_7, x_6/x_8, x_7/x_9, x_8/x_{10}, x_9/x_{11}\}, \\
\sigma_2 = \{x_4/x_3, x_5/x_4, x_6/x_5, x_4/x_7, x_5/x_8, x_7/x_9, x_8/x_{10}, x_9/x_{11}\}.
\end{array}
$$