Research on Logic and Computation in Hypothesis Finding

Yoshitaka Yamamoto

DOCTOR OF PHILOSOPHY

Department of Informatics,
School of Multidisciplinary Sciences

The Graduate University for Advanced Studies

2010

Contents

List of Figures

Acknowledgements

I am very grateful to my supervisor, Pr. Katsumi Inoue. It was still in my undergraduate course at his laboratory when I involved in the research topic on hypothesis finding. Since this time, I have continued to study this topic. I could not complete this thesis without his generous support and patience during my time as an undergraduate, graduate and PhD student. I am very grateful to Pr. Ken Sato, Pr. Makoto Tatsuta, Pr. Akihiro Yamamoto and Pr. Taisuke Sato for their inspiration and fruitful comments on this work. Their advice on the research topics encouraged me to write the thesis. I would also like to present my sincere thanks to Pr. Koji Iwanuma for his constant and kind support.

Motivation for this thesis also emanated from precious meeting with my colleagues in National Institute of Informatics (NII). Pr. Andrei Doncescu impressed me with the interests in systems biology. I benefited from the broad experience to study on Systems Biology at the laboratory of Pr. Gérald Goma in Toulouse. Another inspiring stay was my visit to Dr. Oliver Ray in Bristol. Discussions with him greatly stimulated me and gave some insights on the research.

I would like to thank Pr. Hidetomo Nabeshima for his kind advice on SOLAR and implementation of CF-induction, Pr. Yoshitaka Kameya for his beneficial comments on relational learning, and Pr. Stephen Muggleton and Pr. Gilles Richard for their constructive discussions in NII. I would also like to thank the anonymous refers of several conferences and workshops for their feedback on parts of this thesis.

I gladly acknowledge the financial support of NII as well as both Japan Science and Technology Agency (JST) and Centre National de la Recherche

Scientifique (CNRS, France) under the 2007-2009 Strategic Japanese-French Cooperative Program "Knowledge-based Discovery in Systems Biology", and also 2008-2011 JSPS Grant-in-Aid Scientific Research (A) (No. 20240016). I owe my thanks to Takehide Soh, the secretaries at our laboratory and the staffs at the Graduate University for Advanced Studies for their kind help and constant encouragement.

Last, I wish to thank my parents Takemi and Shizuko and my family Ayuka and our kid, Kousei for their understanding and continuous supports that have made me overcome any difficulties in my life.

Abstract

The thesis studies the logical mechanism and its computational procedures in hypothesis finding. Given a background theory and an observation that is not logically derived by the prior theory, we try to find a hypothesis that explains the observation with respect to the background theory. The hypothesis may contradict with a newly observed fact. That is why the logic in hypothesis finding is often regarded as ampliative inference.

In first-order logic, the principle of inverse entailment (IE) has been actively used to find hypotheses. Previously, many IE-based hypothesis finding systems have been proposed, and several of them are now being applied to practical problems in life sciences concerned with the study of living organisms, like biology. However, these state-of-the-art systems have some fundamental limitation on hypothesis finding: They make the search space restricted due to computational efficiency. For the sake of incompleteness in hypothesis finding, there is an inherent possibility that they may fail to find such sufficient hypotheses that are worth examining.

The thesis first provides such a practical problem, where those incomplete procedures cannot work well. In contrast, this motivating problem is solved by CF-induction, which is an IE-based procedure that enables us to find every hypothesis. On the other hand, complete procedures like CF-induction have to deal with a huge search space, and thus, are usually achieved by consisting of many non-deterministic procedures.

The thesis next shows an alternative approach for finding hypotheses, which is based on the inverse relation of subsumption, instead of entailment. The proposed approach is used to simplify the IE-based procedures by reducing their non-determinisms without losing completeness in hypothesis finding. Together with this result, we logically reconstruct the current procedure of CF-induction into a more simplified one, while it ensures the completeness. Through the thesis, we will see underlying nature and insights to overcome limitations in the current IE-based hypothesis finding procedures.

Chapter 1

Introduction

1.1 Background

We are used to *hypothesize* in various aspects of daily life. When infant children are crying, their mothers would give some milk to the infants. That is because mothers consciously or not hypothesize that infants want some milk when they cry. Hypothesizing also plays an important role in business. When quality-engineers face some claim to a product from the market, they would examine whether or not the other products in the same lot have the same defect. That is because they hypothesize that if some product includes a defect, the others in the same lot can also include the same defect.

According to a dictionary, the word "hypothesis" means a proposition made as a basis for reasoning, *without the assumption of its truth.* In other words, generated hypotheses are not necessarily always in the right. Indeed, crying children may want to change diapers and products with some defect may be occurred singly. That is why the logic in hypothesizing is viewed as *ampliative* inference: *abduction* and *induction*, which are epistemologically distinguished with *deductive* inference.

Research topics on inference-based hypothesis finding have been actively studied in the community of inductive logic programming (ILP). Historically, ILP has been first defined as the intersection of machine learning and logic programming to deal with induction in first-order logic [43, 48, 36]. Compared with the other machine learning techniques, ILP works on both tasks of *classification* and *theory completion* in complex-structured data.

Oceans of data are now being generated in great volumes and in diverse formats. In life sciences, biochemical data is also rapidly produced from emerging high throughput techniques, whereas the whole life-systems, such as signal transduction, gene regulation and metabolism, are still incompletely clarified. For this situation, ILP techniques are recently being applied in life sciences to find hypotheses that can complete the prior life-systems. The huge and diverse biochemical data can be lumped with richer knowledge representation formalisms in first-order logic. Thus, unlike the other machine learning techniques, ILP has applicability to discovering causal relations and missing facts that are lacked in the biochemical knowledge.

In this section, we first review abduction and induction by explaining their tasks, similarities and differences as well as introducing several kinds of inductive tasks. Next, we review inductive logic programming by describing the task and advantages as well as its history in brief. Lastly, we consider an inherent possibility to apply ILP techniques in life sciences by reviewing several past application examples.

1.1.1 Abduction and Induction

Both abduction and induction are *ampliative* inference to seek a hypothesis that accounts for given observations or examples. Generated hypotheses provide us more information by adding them to the prior background knowledge. On the other hand, it may be falsified when new knowledge is obtained. According to the philosopher C. S. Peirce [53, 24], whereas induction infers similarities or regularities confirmed in the observations, abduction infers causalities that are different from what we directly observe. For instance, from the positions of planets in the space, inferring that the planets move in ellipses with the sun is an inductive task. Because this task is related to finding a similarity hidden in the orbits of planets. In contrast, from the observation that an apple drops on the ground, inferring the existence of gravity is an abductive task. That is because though the concept of gravity explains the observation, we cannot explicitly sense the existence of gravity. We note that the word “abduction” is used to mention the property taking subjects away illegally in general. Indeed, we often see this word in articles on the North Korea *abductions* of Japanese. Similarly, abduction as the inference is also used to bring us unexpected inspirations with respect to observations.

As the discovery of universal gravitation, discerning hypotheses often have loaded to paradigm shifts in science. Not only in science, but also in business or everyday life, we use induction and abduction especially for generating theories. Let us give a toy example (1): Suppose that when we walk around a university in autumn, we observe that a tree on the east side begins to turn color faster than others. From the observation, we may infer the abductive hypothesis that the sunlight causes the autumn color in the tree. That is because the sunlight tends to get into trees on the east side more brightly than others on the west. By generalizing the abductive hypothesis, we may obtain the inductive hypothesis: if an arbitrary tree is much sun-exposed, then it turns color faster than others. This inductive hypothesis can be verified by checking whether or not it is applicable to another tree. If the verification results in rejection of the current hypothesis, we need to refine it and verify once again the modified one. Otherwise, we can keep the current hypothesis as it is. In this way, we generate a concrete theory using abduction and induction in the cycle of hypothesis formation and verification.

From an epistemological point of view, abduction and induction are clearly two different processes. On the one hand, induction refers to inductive *generalization*: It is used to find general laws that account for given observations possibly with the background theory. On the other hand, abduction is used to infer an *explanation* for some specific observed situations or properties. However, in a logical standpoint, they are not necessarily distinguished with each other, because induction can often induce explanations for the observations. Recall the example [22] where we know Tweety is a bird and we suddenly observe that he is able to fly. Then, we may infer that every bird can fly. This hypothesis is an inductive generalization, and simultaneously, can be regarded as an explanation for the observation.

Lachiche [33] pointed out this identification in abduction and induction, and especially called the “explanation-based” induction as *explanatory in-*

duction. Given a background theory B and observations E, the task of explanatory induction is to find a hypothesis H such that

$$B \wedge H \models E, \tag{1.1}$$

$$B \wedge H \textit{ is consistent}, \tag{1.2}$$

where $B \wedge H \models E$ denotes the entailment relation[1] that if B and H are true, then E is also true in brief.

Formulas (1.1) and (1.2) can be adopted in the logical formalization of abduction. That is because hypotheses generated by abduction are explanations of observations that satisfy two formulas. Hence, explanatory induction provides an integrated framework of induction with abduction. In other word, explanatory induction can find both inductive and abductive hypotheses. The above example (1), described in Figure 1.1, is such a case that abduction and induction are necessary to generate the target hypothesis.

$$\begin{array}{ll}
\mathbf{B}\text{: } \mathit{locate}(X, \mathit{east}) \rightarrow \mathit{sunlight}(X) & \\
\mathbf{E}\text{: } \mathit{locate}(\mathit{tree}(a), \mathit{east}) \rightarrow & \mathit{autumn_color}(\mathit{tree}(a)) \\
\hline
(\mathit{abduction}) \quad \mathit{sunlight}(\mathit{tree}(a), \mathit{east}) \rightarrow & \mathit{autumn_color}(\mathit{tree}(a)) \\
\hline
(\mathit{induction}) \quad \mathit{sunlight}(\mathit{tree}(X), \mathit{east}) \rightarrow & \mathit{autumn_color}(\mathit{tree}(X))
\end{array}$$

Figure 1.1: Integration of Abduction with Induction

Along with explanatory induction, several different formalizations of induction has been proposed in the literature [22, 21, 33, 27, 66] such as *Descriptive induction*, *Circumscriptive induction* and *Brave induction*. In explanatory induction, it is often difficult to infer regularities confirmed in observations [22]. For instance, suppose that we have two plastic bottles such that the one is hot and the other is cold, and we suddenly notice that the former has a red cap whereas the latter has a white cap. Then, we may infer that plastic bottles with a red (*resp.* white) cap are hot (*resp.* cold). This is an inductive hypothesis that shows a general relation between cap color

[1]Please see Chapter 2 for the precise definition.

and temperature in plastic bottles. However, explanatory induction cannot generate this hypothesis. Instead, it infers an alternative hypothesis that hot (*resp.* cold) bottles have a red (*resp.* hot) cap. We can represent this example with the logical formalization as follows:

$$\begin{aligned}
&B: \ hot(b_1) \wedge cold(b_2).\\
&E: \ cap(b_1, red) \wedge cap(b_2, white).\\
&H_1: \ (cap(X, red) \rightarrow hot(X)) \wedge (cap(X, white) \rightarrow cold(X)).\\
&H_2: \ (hot(X) \rightarrow cap(X, red)) \wedge (cold(X) \rightarrow cap(X, white)).
\end{aligned}$$

Though H_1 is a considerable hypothesis generated by induction, it cannot explain E with respect to B. Accordingly, H_1 cannot be obtained in the context of explanatory induction. Descriptive induction [22, 21, 33] has been proposed to come up with this limitation in explanatory induction. A hypothesis H in descriptive induction is usually defined with so-called *completion technique*, and satisfies the following condition:

$$Comp(B \wedge E) \models H. \tag{1.3}$$

where $Comp(B \wedge E)$ denotes the predicate completion relative to all predicates in $B \wedge E$ [8]. In the above example, $Comp(B \wedge E)$ is obtained by adding the following theories to $B \wedge E$:

$$\begin{aligned}
&(hot(X) \rightarrow X = b_1) \wedge (cold(X) \rightarrow X = b_2).\\
&(cap(X, red) \rightarrow X = b_1) \wedge (cap(X, white) \rightarrow X = b_2).
\end{aligned}$$

Roughly speaking, these formulas complete the extensions of each predicate *hot*, *cold* and *cap* using individuals that appear in $B \wedge E$. Since $Comp(B \wedge E)$ derives H_1 and H_2, both are hypotheses of descriptive induction, whereas only H_1 can be derived by explanatory induction.

If we should accept the task of induction as *completion of individuals* that Lachiche [33] pointed out, descriptive induction based on Clark's predicate completion is an alternative but different kind of inductive inference from explanatory induction. Compared with descriptive induction, explanatory induction finds classifications rather than regularities or similarities hidden in observations. In fact, the hypothesis H_2 obtained by explanatory induction classifies each plastic bottle into two classes (that is, the one with a red cap and another with a white cap) according to its temperature. Note that every hypothesis obtained by explanatory induction is not necessarily solved by descriptive induction as follows:

$$B : plus(X, 0, X).$$
$$E : plus(X, s(0), s(X)).$$
$$H : plus(X, Y, Z) \rightarrow plus(X, s(Y), s(Z)).$$

The predicate $plus(X, Y, Z)$ means that the sum of X and Y is equal to Z. $s(X)$ denotes the successor function of X that satisfies $s^0(X) = X$ and $s^n(X) = s(s^{n-1}(X))$. Since H logically explains E with respect to B, H is a hypothesis in explanatory induction. On the other hand, this hypothesis cannot be obtained by descriptive induction. Indeed, the completion theory $Comp(B \wedge E)$ does not derive H, which is obtained by adding the following theory to $B \wedge E$:

$$plus(X, Y, Z) \rightarrow (Y = 0 \wedge Z = X) \vee (Y = s(0) \wedge Z = s(X)).$$

In the case of $Y = 0$ and $Z = X$, $plus(X, s(Y), s(Z))$ holds since this becomes equivalent to $E = plus(X, s(0), s(X))$. However, in the other case of $Y = s(0)$ and $Z = s(X)$, $plus(X, s(Y), s(Z))$ does not hold since $Comp(B \wedge E)$ never state whether or not $plus(X, s^2(0), s^2(X))$ is true. We may notice that H is a missing definition of addition. In other word, H completes the prior (incomplete) background theory. In this sense, explanatory induction is suitable for inductive tasks such as theory completion [27].

There is an integrated framework of explanatory induction with descriptive induction, called *circumscriptive induction* [27]. This overcomes *inductive leap* that explanatory induction can pose. Let the background theory B and the observation E as follows:

$$B : bird(Tweety) \wedge bird(Oliver).$$
$$E : flies(Tweety).$$

Explanatory induction then infers the hypothesis $bird(X) \rightarrow flies(X)$. This hypothesis logically explains not only the observation $flies(Tweety)$, but also $flies(Oliver)$ that is regarded as an inductive leaf. Because we cannot state that $Oliver$ also flies like $Tweety$ by the prior knowledge. In contrast, circumscriptive induction infers an alternative hypothesis H based on the notion of *circumscription* [39] as follows:

$$bird(X) \wedge X \neq Oliver \rightarrow flies(X).$$

Since H logically explains E with B and also is derived from $Comp(B \wedge E)$, H is a hypothesis in both explanatory and descriptive induction.

There is an extented framework of explanatory induction, called *brave induction* [66]. Brave induction is based on the notion of *minimal models.* Explanatory induction requires the condition that every model of $B \wedge H$ is a model of E, since $B \wedge H \models E$ holds[2]. On the other hand, brave induction requires the condition that *at least one* model of $B \wedge H$ is a model of E. Hence, every hypothesis in explanatory induction is a hypothesis in brave induction. Figure 1.2 represents the sets of hypotheses in each framework of induction using Venn diagrams.

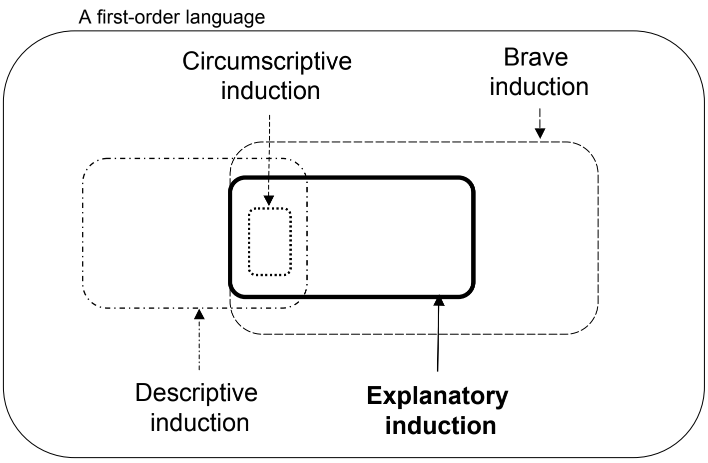


Figure 1.2: Hypotheses in Each Framework of Induction

[2]Please see Formula 1.1 in Page 4.

Induction played a central role at the beginning of machine learning field, and it has been evolved into powerful systems for solving classification problems and recently for theorem completion in a relative young branch of machine learning: ILP. We next briefly describes the inter-related histories of machine learning and ILP.

1.1.2 Inductive Logic Programming

Within Machine Learning, the research field of ILP is characterized as one of the paradigms, called *inductive learning*. Unlike the other paradigms such as the analytic paradigms, the connectionist paradigm and the genetic paradigm, its aim is to induce a general concept description from a sequence of instances of the concept and known counterexamples of the concept [6].

In the historical perspective[3], the inductive learning has been developed interacting with development of deduction as its counterpart. It seems likely to be natural to go back two underlying theorems given by Gödel [18] around early 1930's. Gödel demonstrated that a small collection of sound rules of inference was complete for deriving all consequeces, and after a year that he proved this completeness result, in 1931, he proved the more famous incompleteness theorem that Peano's axiomisation of arithmetic, and any first-order theory containing it, is either-contradictory or incomplete for deriving certain arithmetic statements. These theorems by far influences both research fields of deduction and induction. The second incompleteness theorem involved many computer scientists like Turing [77] in noticing that intelligent machines require the capability of learning from examples. In turn, the first complete theorem much later resulted as the fruitful discovery of resolution principle given by Robinson [64], where a single rule of inference, called *resolution*, is both sound and complete for proving statements within this calculus. Based on his discovery, a simple question might arise through visionary researchers: "If all the consequences can be derived from logical axioms by deduction, then where do the axioms come from?". Indeed, around the most same time as Robinson's discovery, Banerji [4] tried to introduce

[3] Much of this subsection is adopted from [34, 43, 80]

the predicate logic formalization in inductive learning.

From view point of correctness in inductive learning, it is worth nothing that Gold [20] introduced the concept, called *Identification in the limit*. An inductive learning algorithm reads some finite number of examples, and derives a correct description from which all the positive and none of the negative examples can be derived. However, for concepts with infinite instances, the inductive task faces with the *Problem of Induction* by D. Hume [23]. That is, positive examples that are not yet presented can be outside the induced concept description, though some instances of the prior concept description will appear as counterexamples later [80]. Against this problem, Gold, based on a Popperian idea (roughly speaking that if theories are incorrect, then *eventually* they will be falsified) suggested that if an inductive learning algorithm can examine some finite number of examples, refute an incorrect part of the prior concept description and modify it, then the algorithm will eventually find a correct concept description. This kind of convergence in his work has continued in *Version Space* by Mitchell [42], *Model Inference System* by Shapiro [72] and *PAC-learning* by Valiant [79].

The germ of ILP has been already seen in Plotkin's work of the early 1970's. In the thesis [54], he considered the generalization in the clausal formalization with the subsumption order, and introduced the inductive mechanism, called *least general generalization* to compute the least generalization between clauses. Based on this mechanism, he also introduced the concept *Relative subsumption*, which enables us to interact with the usage of the background theory. One essential feature of ILP that distinguishes it with other inductive learners is the usage of the background theory. Thus, his thesis laid the foundations for much of the present activity in ILP. On the other hand, his theoretical result was negative: that is, he showed that there is in general no finite relative least generalization of two clauses. It seems that this negative result motivated Shapiro [72] to take an approach to refine the current hypothesis from specific to general, rather than Plotkin's general to specific. Besides, it is important that he first clearly distinguished the semantics and syntax in the context of refinement of hypotheses.

Despite of his theoretical works with far-sightedness, most successes within machine learning field have derived from systems which construct hypotheses within the limits of propositional logic. For instance, MYCIN by Shortliffe, Buchaman [73] and BMT based on Quinlan's ID3 algorithm [57] were efficiently applied as expert systems for specific domains such as medical diagnosis. In 1980's, several inductive systems within the predicate logic formalization have been proposed for the sake of the limitations in propositional logic. Sammut and Banerji [67] introduced a system called MARVIN which generalizes a single example at a time with reference to a set of background clauses. In turn, Quinlan [58] described a system, called FOIL, which performs a heuristic search from specific to general with the notion of an information criterion related to entropy. Note that, whereas MARVIN uses the background theory, FOIL does not distinguish the inputs into examples and the background theory. From this difference, MARVIN would be characterized as one of the ancestors continuing in the present systems. Indeed, Muggleton et al. showed that the generalization in MARVIN is a specific case of inverting reolution processes, and described a system, called GOLEM, which is based on a relative general generaltization [49]. In 1991, a year before he described GOLEM, the first international conference on ILP has held at Viana Do Castelo in Porto.

After the establishment of this conference, ILP has been evolved as the research area including theory concerning induction (not only explanatory induction, but also the other inductive frameworks that we showed above) and abduction, implementation and experimental applications so far. In summary, compared with the other inductive learning techniques, we list three merits of ILP as follows:

- Rich representation formalisms: ILP uses the first-order predicate logic. This feature enables to bring us beyond the restricted formalisms in propositional logic. As a result, ILP can deal with highly-complex structured data, which attribute-based algorithms like decision trees cannot do.

- Usage of the background knowledge: ILP distinguishes the input formulas into examples and the background knowledge if it is necessary.

In everyday life, we are used to utilize the background knowledge. In this sense, it would be natural to use the background theory for finding hypotheses.

- Integrated framework of abduction and induction: In the context of explanatory induction, ILP can perform both abduction and induction. For this feature, it can potentially find some missing general rules or facts in the prior background theory. This becomes important especially in case that the background theory is assumed to be incomplete.

- Readability of the output theory: Unlike other inductive learning techniques, such as Neural networks, Baysian networks and SVM, outputs of ILP are represented as formulas that we can easily read and verify them. For readability of the output theory, ILP can involve hypothesis formation in scientific discovery by directly interacting with users.

We recall the Michalski's train example [35]. This example would give us an insight to see what kind of problems ILP sufficiently works on, compared with other inductive learning algorithms.
Michalski's train example [35]: We observe two kinds of trains: One goes west and the other goes east (See Figure 1.3). Every train contains several cars each of which carries cargos and has its own shape. The task is to discover a regulatory relation that decides the direction of each train. The prior knowledge of each train and car is represented in first-order language. For instance, $Train1$ and $Car1$ are as follows:

$$Train\ 1: \ east(t_1).\ has\_car(t_1, c_3).\ has\_car(t_1, c_4).\ has\_car(t_1, c_5).$$
$$Car\ 1: \ closed\_top(c_1).\ length(c_1, long).\ has\_cargo(c_1, circle).$$

Note that the predicate $has\_car(t, c)$ means the train t contains the car c, the predicate $closed\_top(c)$ means the car c is closed at its top, the predicate $length(c, long)$ (*resp.* $length(c, short)$) means the length of the car c is long (*resp.* short) and the predicate $has\_cargo(c, X)$ means the car c has at least one X-shaped cargo. The positive examples E^+ correspond to the fact that $Train1$, 2 and 3 go east. In contrast, we treat as the negative examples the

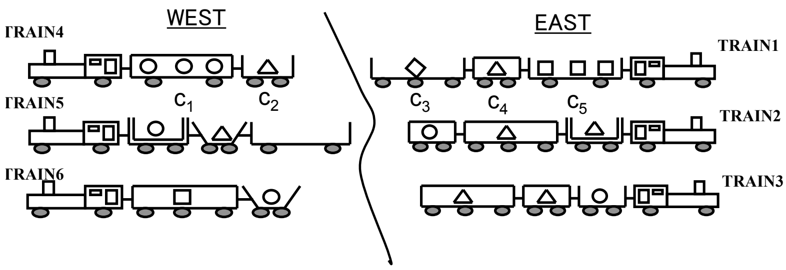


Figure 1.3: Michalski's Train Example

fact that $Train4$, 5 and 6 go west. The background theory B consists of two kinds of the information: which car each train has and what kind of features each car has. In the problem setting of ILP, the task is to find a classification rule that logically explains E^+ but does not explain E^-, with respect to B. The following is such a consistent rule H:

$$H = \mathit{has\_car}(X, Y) \wedge \mathit{closed\_top}(Y) \wedge \mathit{length}(Y, \mathit{short}) \rightarrow \mathit{east}(X), \quad (1.4)$$

where X and Y correspond to a train and car, respectively.

This classification rule cannot be obtained by so-called *attribute-based* algorithms [65] like decision trees. Attribute-based algorithms deal with a collection of *objects*, each of which consists of a *class* and *attributes* that represent properties of the object. In case that objects have one class and n attributes, any rules obtained by those algorithms are represented into the following logical form:

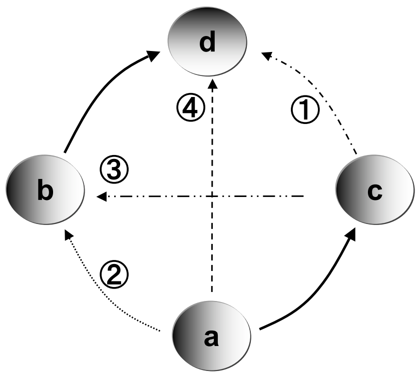
$$\mathit{attr}_1(\mathit{Obj}, \mathit{val}_1) \wedge \mathit{attr}_2(\mathit{Obj}, \mathit{val}_2) \wedge \cdots \wedge \mathit{attr}_n(\mathit{Obj}, \mathit{val}_n) \rightarrow \mathit{class}(\mathit{Obj}),$$

where the predicate $\mathit{attr}_i(\mathit{Obj}, \mathit{val}_i)$ $(1 \leq i \leq n)$ means that the value of the i th attribute is val_i in the object Obj. In other words, attribute-based algorithms can seek *only* such rules that are described in the above form. However, the above hypothesis (1.4) cannot be represented with this form, since the predicates *closed_top* and *length* are not so much attributes of trains as attributes of cars. Indeed, those predicates refer to features of cars in terms of their roofs and lengths. That is why the hypothesis (1.4) cannot be obtained by the attribute-based algorithms.

This example shows that ILP can find a classification rule between a target class (i.e. Direction) and properties (i.e. roof and length) of an attribute (i.e. car) in objects (i.e. trains). In this way, ILP focuses on not only attributes but also those features using rich representation formalisms in first-order logic. For this feature, ILP has been applied to classification problems that deal with multiple hierarchic structures.

Along with classification, ILP is used for theory completion tasks.

Graph completion problem: Let us give such a toy example. Consider the left graph consisting of four nodes a, b, c and d, and two arcs $a \rightarrow c$ and $b \rightarrow d$. The left graph describes the background theory. Suppose that we newly observe there is a path from a to d. This cannot be explained by the background theory. This means the prior graph is incomplete in the sense that the graph has some missing arc. For this incomplete graph, ILP can find possible arcs like $c \rightarrow d$ (1), $a \rightarrow b$ (2), $c \rightarrow b$ (3) or $a \rightarrow d$ (4) in the context of explanatory induction. It is not straightforward that the other machine learning techniques such as decision trees, Neural networks and SVM perform this kind of learning task, called *theory completion*.



As we described above, ILP is sufficient for the classification and theory completion problems that deal with complex structured data. Based on this feature, recently it has been growing interest to apply ILP techniques to practical problems in life sciences. Its common motivation comes from the perspective that ILP can find some unknown causal relations or missing facts that are lacked in biochemical knowledge database. We next describe such applicability in life sciences by introducing several practical problems where ILP techniques were actually applied and successfully working.

1.1.3 Applicability in Life Sciences

We first introduce an application [45] to the *protein secondary structure prediction*. It is known that each protein has some three-dimensional structure, called *folding*. Every folding is dominated by the sequence of so-called *secondary structures* each of which corresponds to a regulatory cubic structure emerged in the polypeptide chain. Figure 1.4 describes two foldings of pro-

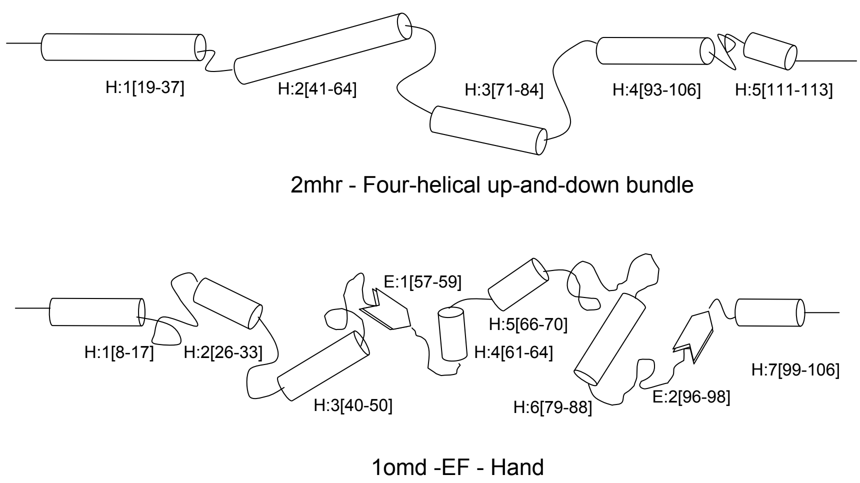


Figure 1.4: Protein Folding Prediction as Classification Problem

teins: "2mhr four helical up and down" and "1omd EF hand". Whereas there are around 300 known foldings, half of all known proteins are member of the 20 most populated foldings [45].

The inductive task involved in the protein secondary structure prediction is to discover some characteristics on secondary structures of proteins that dominate their foldings. We may notice that this problem setting has a similarity with the Michalski's train example. Indeed, if cars, trains and their directions in the example should be identified with secondary structures, proteins and their foldings, respectively, the task of the Michalski's example is the same as the protein secondary structure prediction.

Thus, Progol [44], one of the state-of-the-art ILP systems, has been applied to those most populated 20 foldings and resulted in around 70% accuracy at the cross-validated prediction, which is higher on average, compared with other machine learning techniques [45, 50]. For example, in the case of "Four helical up and down bundle" in Figure 1.4, Progol generated the following considerable hypothesis:

The protein P has this fold if it contains a long helix H1 at a position between 1 and 3, and H1 is followed by a second helix H2.

In this problem, ILP was used as a classification tool in inductive learning. As we explained before, ILP can be also used in a theory completion tool in the context of explanatory induction.

B. Zupan et al. have developed so-called *GenePath* system to automatically construct the genetic networks from mutant data [93, 94]. Their work can be viewed as another application of ILP using the function of theory completion. A *gene regulatory network* is usually described by a collection of interactions between genes, which is involved in physiological behavior, called *phenotype*, in a target life system. For understanding the influence of a target gene on phenotype, geneticists make a mutant obtained by usually knocking out the gene, and verify how phenotype emerges in the mutant. For example, let A and B be two genes that are involved in some target phenotype, and $\varDelta A$, $\varDelta B$ and $\varDelta AB$ be three mutants obtained by knocking out Gene A, B and both of A and B, respectively. In case that the phenotype more strongly emerges in $\varDelta A$, it is assumed that Gene A has an inhibitory effect on the phenotype. In contrast, if the emergence becomes weaker in $\varDelta B$, Gene B is assumed to have an activating effect. There are three possibilities for graphically representing these two interactions. In the middle graph of Figure 1.5, both paths from A and B are parallel with each other. On the other hand, the left (*resp.* right) graph is described in such a way that Gene A (*resp.* B) is upstream from Gene B (*resp.* A). These possible cases are uniquely determined by checking how the phenotype emerges in the mutant $\varDelta AB$. If it more strongly emerges like $\varDelta B$, we can predict that Gene A should influence the phenotype in upstream process from Gene B. This case corresponds to the left graph in Figure 1.5. In contrast, if there is no change in the emergence, it is assumed that the counter influences of Gene A and B cancel out each other. This case corresponds to the middle graph.

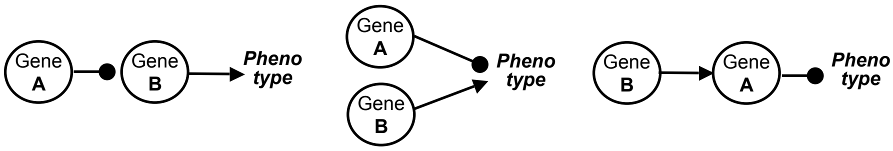


Figure 1.5: Possible Gene Regulatory Networks

Using these internal rules that are simple but actually used by experts, ZenePath automatically constructs an gene regulatory network from mutant data. In the literature [93, 94], Zupan et al. focused on the developmental cycle from independent cells to a multicellular form emerged in the social amoeba *Dictyostelium*. Under starvation, the amoebae stop growing and aggregate into a multicellular fruiting body. In contrast, they keep the original single cells in nutrient-rich environment. They showed that ZenePath succeeded in generating a considerable regulatory network involved in this physiological transition from mutant data in Dictypstelium (See Figure 1.6 [93]).

Exp No.	Genotype	Aggregation {−, ±, +, ++}
1	wild-type	+
2	*yakA*$^-$	−
3	*pufA*$^-$	++
4	*pkaR*$^-$	++
5	*pkaC*$^-$	−
6	*acaA*$^-$	−
7	*regA*$^-$	++
8	*acaA*$^+$	++
9	*pkaC*$^+$	++
10	*pkaC*$^-$, *regA*$^-$	−
11	*yakA*$^-$, *pufA*$^-$	++
12	*yakA*$^-$, *pkaR*$^-$	+
13	*yakA*$^-$, *pkaC*$^-$	−
14	*pkaC*$^-$, *yakA*$^+$	−
15	*yakA*$^-$, *pkaC*$^+$	++

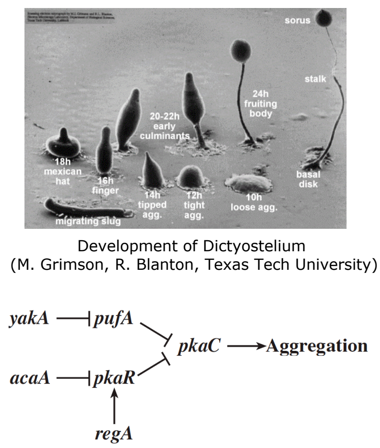


Figure 1.6: Mutant Data and Generated Gene Network

The task of ZenePath is to find possible gene networks that explain the phenotype emerged in mutants with respect to several internal rules used in experts. Thus, we may view this as theory completion on biochemical networks in the context of explanatory induction. As a similar application that aims at completing biochemical networks, there is the *Robot Scientist* project by the team of R. King [32]. In this project, they have developed a physical implement which can automatically detect unknown gene functions in yeast using ILP techniques and also experimentally evaluate those detected functions. In the verification step, if the hypothesis contradicts with the experiment result, the robot rejects it and additionally generates another hypothesis. In the literature [32], they showed that the robot could automatically refine the metabolic network on the amino acid synthesis.

Tamaddoni-Nezhad et al. have applied to estimate *inhibitory effects* in metabolic pathways [76]. This work deeply involves drug discovery. In general, chemical compounds used as drugs effectively work by inhibiting some target enzyme reactions. It is thus necessary for drug discovery to experimentally examine if a target chemical compound (drug candidate) has specific inhibitory effects to be expected. Those inhibitory effects are confirmed using laboratory rats. First, their urine is sampled before and after administration of the chemical compound, and next the concentrations of several observable metabolites in each urine (before and after the administration) are measured by NMR. Experts then detect which enzyme reactions are possibly inhibited from the measured concentration changes based on their own knowledge. Tamaddoni-Nezhad et al. constructed a logical model that describes this internal knowledge used in experts. The model includes causal relations between enzyme inhibitions and concentration changes of metabolites. Thus, this model enables us to consider which enzyme inhibitions can cause the observed concentration changes. Hence, the estimation of possible inhibitory effects can be achieved with the logical model in the context of explanatory induction. In the literature [76], they showed several inhibitory effects of particular toxins were in fact sufficiently found using Progol.

Recently, a variety of biochemical data is rapidly being produced in great volumes and in diverse formats. For this situation, a new research field, called *systems biology*, is emerging in life sciences. Molecular biology tends to adopt the stance as *reductionism*. In contrast, systems biology tends to integrate life systems, such as genomics, molecular signaling and metabolism, which have been *individually* studied so far (See Figure 1.7). Significant development in

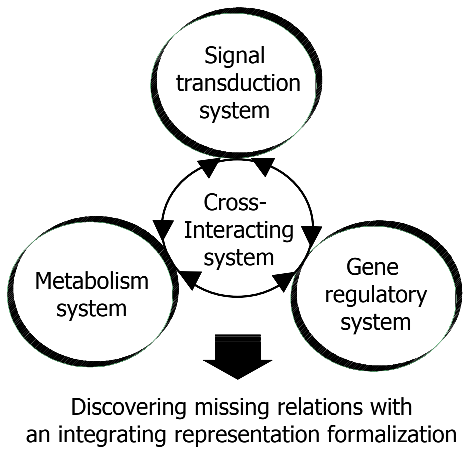


Figure 1.7: Knowledge Discovery in Systems Biology

molecular biology is gradually clarifying the mechanism of each life-system. However, there are still unknown relations or missing facts in each system as well as interactions between systems. It is thus important for systems biology to discover those hidden knowledge lacked in the prior data.

The previous applications show that explanatory ILP techniques are effective to discover classification rules and missing causalities (genes interactions or inhibitory effects) from incomplete biochemical knowledge. Moreover, rich representation formalisms of ILP is sufficient to integrate life-systems that are individually represented in diverse formats so far. We thus believe that explanatory ILP techniques have an inherent capability to play an important role for knowledge discovery in systems biology.

1.2 Motivation

As we explained in the above, several state-of-the-art ILP systems have been applied to practical problems in life sciences. The applicability would be increasing along with growing interests to systems biology. However, there are two fundamental problems in the previously proposed ILP systems. Modern ILP systems can be divided into two types: The one restricts the search space due to computational efficiency. The other ensures the completeness in hypothesis finding, though it has to deal with a huge search space.

The first problem is involved in the former incomplete systems. Every incomplete system has an inherent possibility that there is another hidden hypothesis such that is more beneficial than the actual output. In this sense, those systems make it difficult to guarantee the validity of their solutions. In life sciences, we need to experimentally evaluate the generated hypothesis. If ILP systems do not ensure the validity of their outputs, they cannot be positively accepted by cagy experts who have to pay the costs, which are often quite expensive. Incompleteness of hypothesis finding can make it restrained to apply ILP systems in life sciences. Thus, it becomes necessary to overcome the problem caused in incomplete ILP systems.

This problem never occurs in the latter complete systems. On the other hand, those systems have to deal with a huge search space for preserving completeness in hypothesis finding. Such complete systems are used to consist of non-deterministic procedures. Each non-deterministic procedure makes many choice points where users have to select relevant one by hand. This fact makes it difficult to apply them to practical problems that deal with a large amount of data. The second problem lies in the non-determinisms of the complete systems.

In this thesis, we first provide a practical example in systems biology that the previously proposed incomplete systems cannot solve. That is because this example needs an advanced inference technique that simultaneously integrates abduction and induction. The expected solutions are in the form of abductive and inductive hypotheses in the context of explanatory induction. For this task, we also show *CF-induction*, which is a complete explanatory ILP method, efficiently works together with several interactions to users.

Most of the modern explanatory ILP methods are based on the principle of *Inverse Entailment* (IE). This principle uses the following formula equivalent to Formula (1.1) in the two conditions of explanatory induction:

$$B \wedge \neg E \models \neg H, \tag{1.5}$$

where B, E and H denote a background theory, observations and a hypothesis, respectively. Formula (1.5) means that for any hypothesis, its negation can be derived from the background theory and the negation of observations with *entailment*. Every IE-based method is used to compute hypotheses in two steps: by first constructing an intermediate theory F such that

$$B \wedge \neg E \models \cdots \models F \models \cdots \models \neg H \tag{1.6}$$

and next by generalizing its negation $\neg F$ into the hypothesis H with the inverse relation of entailment $H \models \neg F$.

Non-deterministic procedures in complete ILP systems mainly arise in those two tasks: construction of an intermediate theory and generalization of its negation. Both tasks involve in the problem how to realize the entailment relation. Given a background theory B and observations E, there are many possible intermediate theories to be constructed, each of which is derived from $B \wedge \neg E$ with the entailment relation. Moreover, for some constructed theory F, there are also many possible hypotheses to be generated, each of which is derived from $\neg F$ with the inverse relation of entailment.

This thesis thus considers the issue on how those two tasks can be logically simplified, while completeness in hypothesis finding is preserved. We first focus on the sequence of intermediate theories that constructs a derivation from $B \wedge \neg E$ to $\neg H$ in Formula 1.6. We then show the negations of those intermediate theories can be represented with inverse subsumption. This logical reduction enables us to use the subsumption relation in generalization, instead of entailment, without losing the completeness. Based on this result, we next logically reconstruct the current procedure of CF-induction into a more simplified form. For its theoretical advantage preserving the completeness, CF-induction potentially has plenty of practical applications in systems biology like our motivating example. On the other hand, it required users interactions each of which has to be selected one from many choice points by hand. This fact has held the practical applications of CF-induction back. By logically simplifying the current procedure, the non-determisims in CF-induction can be reduced, and thus it would become possible to automatically compute sufficient hypotheses that users wish to obtain.

1.3 Contribution

The contribution mainly consists of the following three works:

- The first contribution is to show new applicability of ILP techniques in life sciences [88, 86, 13]. We provide a new practical example in systems biology that cannot be solved by the previously proposed incomplete systems. This example shows one limitation of those incomplete systems as well as the necessity of somehow ensuring the validity of the generated hypothesis. The task in this example is achieved by an advanced inference integrating abduction and induction. We also show how this task can be performed using CF-induction.

- The second contribution is to prove that the complete generalization in the IE-based methods can be achieved by inverse relation of subsumption [90]. Previously, it has been known that the generalization based on the entailment relation can ensure completeness in hypothesis finding. However, this procedure needs to consist of many so-called *generalization operators* such as *inverse resolution*. Each generalization operator has many ways to be applied and any combination of them is also applied as another generalization operator. This fact makes the generalization procedure highly non-determinisitc. For this problem, we show inverse subsumption, instead of entailment, is sufficient to ensure the completeness in generalization.

- The third contribution is to logically reconstruct the procedure of CF-induction into a more simplified form [91, 92]. Like other IE-based methods, CF-induction consists of two procedures: construction of intermediate theories and generalization of its negation. In the previous CF-induction, each procedure required several users interactions where some relevant one should be selected from many choice points. In contrast, we first show a deterministic procedure to construct intermediate theories, while our proposal does not lose any completeness in hypothesis We next propose two possible approaches for generalization task.

> The first approach is based on the logical relation between the negation of an intermediate formula and a hypothesis. This logical relation can be realized with inverse subsumption using the result is the second contribution. Alternatively, the second approach is based on the logical relation between an intermediate formula and the negation of a hypothesis. Compared with the first approach, the second approach actively uses deductive inference. We thus show that in both approach, the non-determinism in generalization can be dramatically reduced. We also consider efficient implementation in CF-induction [89].

In summary, the thesis shows new applicability of inference-based hypothesis-finding techniques in life sciences as well as essential limitations in the previously proposed ILP methods. The thesis also provides us fundamental properties in hypothesis finding that can be commonly applied in the IE-based explanatory ILP methods, and also propose sound and complete procedures obtained by logically simplifying CF-induction. We believe that the contents shown in the thesis would give us underlying nature and insights to clarify the logic and computation in hypothesis finding.

1.4 Overview

The rest of this thesis is organized as follows. Chapter 2 reviews the notions and terminologies in this thesis, which include the syntax and semantics in first-order logic, clausal forms and consequence finding as well as the dualization problem. Chapter 3 reviews the principle of inverse entailment and introduces each previously proposed hypothesis-finding method based on inverse entailment including CF-induction. Chapter 4 provides a new practical application in systems biology. Its task is to find both abductive and inductive hypothesis that can complete the prior background theory. We show how this advanced inference can be realized using CF-induction. The content in Chapter 4 corresponds to the first contribution in the previous section. Chapter 5 shows that the generalization relation in hypothesis finding based on inverse entailment can be reduced to inverse subsumption. The content in Chapter 5 corresponds to the second contribution in the previous section. In

Chapter 6, we focus on the current procedure of CF-induction and logically reconstruct each non-deterministic procedure into a more simplified one. We also discuss about efficient implementation of CF-induction with consideration of issues concerning the non-monotone dualization problem. Chapter 7 concludes and describes future works.

1.5 Publications

The referred publications are as follows:

1. Yoshitaka Yamamoto, Katsumi Inoue and Koji Iwanuma. From inverse entailment to inverse subsumption. *Proceedings of the 20th Int. Conf. on Inductive Logic Programming*, to appear, 2010 [90].

2. Yoshitaka Yamamoto, Katsumi Inoue and Koji Iwanuma. Hypothesis enumeration by CF-induction. *Proceedings of the Sixth Workshop on Learning with Logics and Logics for Learning* (LLLL2009), pages 80-87. The Japanese Society for Artificial Intelligence, 2009 [89].

3. Yoshitaka Yamamoto, Katsumi Inoue and Andrei Doncescu. Integrating Abduction and Induction in Biological Inference using CF-Induction. Huma Lodhi and Stephen Muggleton (eds.), *Elements of Computational Systems Biology, Wiley Book Series on Bioinformatics*, pages 213-234, John Wiley and Sons, Inc., 2009 [88].

4. Yoshitaka Yamamoto, Katsumi Inoue and Andrei Doncescu. Abductive reasoning in cancer therapy. *Proceedings of the 23rd Int. Conf. on Advanced Information Networking and Applications*(AINA 2009), pages 948-953. IEEE Computer Society, 2009 [87]. This paper deals with another biological application of our ILP system based on SOLAR. In this thesis, we do not include the content of this paper.

5. Yoshitaka Yamamoto, Oliver Ray and Katsumi Inoue. Towards a logical reconstruction of CF-induction. K. Satoh, A. Inokuchi, K. Nagao and T. Kawamura (eds.), *In New Frontiers in Artificial Intelligence: JSAI 2007 Conference and Workshop Revised Selected Papers*, *Lecture Notes in Artificial Intelligence*, volume 4914, pages 330-343, Springer, 2008 [92].

6. Yoshitaka Yamamoto and Katsumi Inoue. An efficient hypothesis-finding system implemented with deduction and dualization. *Proceedings of the 22nd Workshop on Logic Programming* (WLP 2008), pages 92-103. University Halle-Wittenberg Institute of Computer Science Technical Report, 2008 [85].

7. Yoshitaka Yamamoto, Katsumi Inoue and Andrei Doncescu. Estimation of possible reaction states in metabolic pathways using inductive logic programming. *Proceedings of the 22nd Int. Conf. on Advanced Information Networking and Applications* (AINA 2008), pages 808-813. IEEE Computer Society, 2008 [86].

8. Yoshitaka Yamamoto, Oliver Ray and Katsumi Inoue. Towards a logical reconstruction of CF-Induction. *Proceedings of the 4th Workshop on Learning with Logics and Logics for Learning* (LLLL 2007), pages 18-24, The Japanese Society for Artificial Intelligence, 2007 [91].

9. Andrei Doncescu, Katsumi Inoue and Yoshitaka Yamamoto. Knowledge-based discovery in systems biology using CF-induction. *New Trends in Applied Artificial Intelligence: Proceedings of the 20th Int. Conf. on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (IEA/AIE 2007), *Lecture Notes in Artificial Intelligence*, volume 4570, pages 395-404, Springer, 2007 [13].

10. Andrei Doncescu, Yoshitaka Yamamoto and Katsumi Inoue. Biological systems analysis using Inductive Logic Programming. *Proceedings of the 21st Int. Conf. on Advanced Information Networking and Applications* (AINA 2007), pages 690-695, IEEE Computer Society, 2007 [14].

Chapter 2

Preliminaries

This chapter reviews the notion and terminology that are used throughout the thesis. Section 2.1 describes the syntax and semantics in the first-order logic as well as its prenex normal and clausal forms (Skolem standard forms). We next review the resolution principle in Section 2.2. In this section, we also introduce several important theorems such as *Herbrand's theorem* and *Subsumption theorem*. Much of Section 2.1 and 2.2 is adopted from [7, 52, 59]. Section 2.3 describes issues on the dualization problem to translate a given conjunctive normal form formula into a logically equivalent disjunctive (*resp.* conjunctive) normal form formula. Note that dualization plays an important role in hypothesis finding based on the principle of inverse entailment.

2.1 First-Order Logic

Through this thesis, we represent the logical formulas using the classical first-order predicate logic. Here, we formally define this representation formalization. Then, we start with the syntax of first-order logical formulas, which is formalized with an alphabet of the first-order logic language $\mathcal{L}$ defined below.

Definition 2.1. An *alphabet* of the first-order logic consists of the following symbols:

- A set of *constant* symbols: $\{$"a", "b", "c", $\ldots\}$.

- A set of *variable* symbols: $\{$"x", "x_1", "y", $\ldots$ $\}$.
- A set of *function* symbols: $\{$"f", "f_1", "g", $\ldots$ $\}$.
- A set of *predicate* symbols: $\{$"p", "p_1", "q", $\ldots$ $\}$.
- The *logical* symbols: "$\forall$", "$\exists$", "$\neg$", "$\wedge$", "$\vee$" and "$\rightarrow$".
- The *punctuation* symbols: "(", ")" and ",".

The two logical symbols "$\forall$" and "$\exists$" are called the *quantifiers*, respectively. The other logical symbols are called the *connectives*. Every function and predicate symbol has a certain number of arguments, called its *arity*. In particular, function and predicate symbols of arity zero are called *constant* and *proposition* symbols, respectively.

Definition 2.2. *Well-formed expressions* are constructed as follows:

1. A *term* is either a variable x, a constant c or a function $f(t_1, \ldots, t_n)$ of arity $n \geq 1$ where $t_1, \ldots, t_n$ are terms.
2. A *atom* is a predicate $p(t_1, \ldots, t_n)$ of arity $n \geq 0$ where $t_1, \ldots, t_n$ are terms. In the case of $n = 0$, $p()$ will be simply be written p.
3. A *formula* is either an atom, a universal $(\forall x)\phi$, an existential $(\exists x)\phi$, a negation $(\neg\phi)$, a conjunction $(\phi \wedge \psi)$, a disjunction $(\phi \vee \psi)$, an implication $(\phi \rightarrow \psi)$, where x is a variable and ϕ and ψ are formulas. Especially, the *equivalence* formula $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ is simply written $\phi \leftrightarrow \psi$.

An expression is called *ground* iff no variables appear in the expression. Let ϕ be a formula. Then, two formulas $(\forall x)\phi$ and $(\exists x)\phi$ are said to be *universally* and *existentially quantified in x*, respectively. ϕ is said to be the *scope* of $\forall x$ and $\exists x$ in $(\forall x)\phi$ and $(\exists x)\phi$, respectively. An occurrence of a variable x in a formula is *bound* if the occurrence immediately follows a quantifier or it lies within the scope of some quantifier that is immediately followed by x. An occurrence of a variable which is not bound, is called *free*. For example, the first occurrence of x in $((\exists x)Q(x) \vee P(x, f(a))$ is bound, whereas the second occurrence of x is free. A formula is *closed* if it does not contain any free occurrences of variables. The *first-order language* given by an alphabet is the set of all the well-formed expressions which can be constructed by the alphabet. In the following, we will not explicitly specify the alphabet used in each example. Instead, we assume that the alphabet includes all the symbols we use in the example. In addition, we assume that every formula is closed, i.e. it has no free variables.

The meaning of each expression in a first-order language like terms, functions and predicates is given by considering what it refers to or whether or not it is true. Hence, the meanings of expressions depend on what the domain of discourse we assume and how we interpret expressions over the domain. The following is the formal definition of an interpretation of the expressions in a given first-order language.

Definition 2.3. Let $\mathcal{L}$ and $\mathcal{D}$ be a first-order language and a nonempty domain of discourse. An *interpretation* I wrt $\mathcal{L}$ and $\mathcal{D}$ consists the following three assignments to each constant, function and predicate occurring in F:

1. To each constant c, we assign an element c^I in $\mathcal{D}$.
2. To each function f of arity $n \leq 1$, we assign a function $f^I\colon D^n \to D$, where $D^n = \{(x_1, \ \ldots, \ x_n) \mid x_1 \in D, \ \ldots, \ x_n \in D\}$.
3. To each predicate p of arity $n \leq 0$, we assign a relation $p^I \subseteq D^n$. (Note that in case that $n = 0$, D^0 denotes a set that includes only one element.)

Given an interpretation in a first-order language $\mathcal{L}$ over a domain $\mathcal{D}$, it is further necessary to associate each variable to some element in $\mathcal{D}$ for determining the value of every expression in $\mathcal{L}$. We define this as follows:

Definition 2.4. Let $\mathcal{L}$ and $\mathcal{D}$ be a first-order language and a nonempty domain. A *variable assignment* h wrt $\mathcal{L}$ and $\mathcal{D}$ is a mapping from the set of variables in $\mathcal{L}$ to the domain $\mathcal{D}$.

Let h and I be a variable assignment and an interpretation wrt a given first-order language $\mathcal{L}$ and domain $\mathcal{D}$.

The *value* of a term t in I under h is determined by the function $[t]^{I,h}$ from the terms of $\mathcal{L}$ to $\mathcal{D}$, defined as follows:

- $[c]^{I,h} = c^I$, for a constant c;
- $[x]^{I,h} = h(x)$, for a variable x;
- $[f(t_1, \ldots, t_n)]^{I,h} = f^I([t_1]^{I,h}, \ldots, [t_n]^{I,h})$, for a function f of arity $n \geq 1$.

The *truth* of a formula F in I under h is determined by the relation $I, h \models F$ over the formulas in $\mathcal{L}$, defined as follows:

- $I, h \models p(t_1, \ldots, t_n)$ iff $([t_1]^{I,h}, \ldots, [t_n]^{I,h}) \in p^I$, for a predicate p;
- $I, h \models (F \rightarrow G)$ iff $I, h \not\models F$ or $I, h \models G$;
- $I, h \models (F \vee G)$ iff $I, h \models F$ or $I, h \models G$;
- $I, h \models (F \wedge G)$ iff $I, h \models F$ and $I, h \models G$;
- $I, h \models (\neg F)$ iff $I, h \not\models F$;
- $I, h \models (\exists x)F$ iff $I, h_{x/d} \models F$ for some $d \in \mathcal{D}$, where $h_{x/d}$ is the variable assignment such that if $x = y$ then $h_{x/d}(y) = d$, otherwise $h_{x/d}(y) = h(y)$.
- $I, h \models (\forall x)F$ iff $I, h_{x/d} \models F$ for every $d \in \mathcal{D}$.

F be a formula in the language. The relation $I, h \models F$ means F is *true* in the interpretation I under the variable assignment h. An interpretation M is a *model* of a formula F iff $M, h \models F$ for every variable assignment h. F is *satisfiable* (or *consistent*) iff there is a model of F. If F is not satisfiable, F is called a *contradiction* (or *inconsistent*). F is *tautology* (or *valid*) iff every interpretation is a model of F.

Let Σ be a set of formulas in the language. An interpretation M is a *model* of Σ iff M is a model of every formula in Σ. Σ (logically) *entails* a formula F, denoted $\Sigma \models F$, iff every model of Σ is a model of F. We call F a *logical consequence* of Σ. If F is not a logical consequence of Σ, then we write $\Sigma \not\models F$. In case that $\Sigma = \{G\}$ for some formula G, we simply write $G \models F$. Σ (logically) *entails* a set of formulas Γ, denoted $\Sigma \models \Gamma$, iff every model of Σ is a model of every formula in Γ. If not $\Sigma \models \Gamma$, we write $\Sigma \not\models \Gamma$. Two formulas F and G are said to be (logically) *equivalent*, denoted $F \equiv G$, iff both $F \models G$ and $G \models F$. Similarly, two sets of formulas Σ and Γ are said to be logically *equivalent*, denoted $\Sigma \equiv \Gamma$, iff both $\Sigma \models \Gamma$ and $\Gamma \models \Sigma$.

Let L be the set of all formulas in the language. Then, the notion of logical entailment can be viewed as a binary relation $\models \subseteq 2^L \times 2^L$. It is known that the entailment relation $\models$ satisfies the following features: (reflexivity) $\Sigma_1 \models \Sigma_1$, (transitivity) if $\Sigma_1 \models \Sigma_2$ and $\Sigma_2 \models \Sigma_3$ then $\Sigma_1 \models \Sigma_3$, (monotonicity) if $\Sigma_1 \models G$ then $\Sigma_1 \cup \{F\} \models G$; (cut) if $\Sigma_1 \models F$ and $\Sigma_1 \cup \{F\} \models G$ then $\Sigma_1 \models G$; (deduction) $\Sigma_1 \cup \{F\} \models G$ iff $\Sigma_1 \models F \rightarrow G$; and (contraposition) $\Sigma_1 \cup \{F\} \models G$ iff $\Sigma_1 \cup \{\neg G\} \models \neg F$, for each formulas F and G and each set of formulas Σ_1, Σ_2 and Σ_3. Especially, we denote by $\Box$ the empty set of formulas. Since there is no models of $\Box$, it holds that a set Σ of formulas is inconsistent iff $\Sigma \models \Box$ holds. Accordingly, it holds that for every two sets Σ_1 and Σ_2, $\Sigma_1 \cup \Sigma_2$ is inconsistent iff $\Sigma_1 \cup \Sigma_2 \models \Box$. Using this property as well as the contraposition, we often denote by $\Sigma_1 \models \neg\Sigma_2$ that Σ_1 is inconsistent with Σ_2. Hence, the consistency condition of two sets Σ_1 and Σ_2 can be represented by $\Sigma_1 \not\models \neg\Sigma_2$.

The entailment relation is an important concept in logic-based artificial intelligence. Once we represent the knowledge in a target system with a first-order language, the entailment relation enables us to obtain new statements that the prior knowledge does not refer to. On the other hand, given a set of formulas Σ, we may not be able to find out in finite time whether or not $\Sigma \models F$ holds for some formula F, since the number of possible interpretations is usually infinite. We will review some related issues on this consequence finding problem in Section 2.3. As its introduction, we define several normal forms like clausal forms in next section.

2.2 Normal Forms and Herbrand's Theorem

In the previous section, we reviewed the syntax and semantics in first-order logic. Given a first-order language $\mathcal{L}$, every formula in $\mathcal{L}$ has several alternative representation formalizations. For instance, the formula $\neg((\exists x)f(x))$ is logically equivalent to the formula $(\forall x)f(x)$. In this section, we introduce several standard forms for representing the formulas in the language $\mathcal{L}$.

Definition 2.5 (Prenex normal forms)**.** A formula F is said to be in *prenex normal form* iff the formula F is the form of

$$(Q_1x_1)\cdots(Q_nx_n)(M).$$

where every Q_ix_i $(1 \leq i \leq n)$, is either $(\forall x_i)$ or $(\exists x_i)$, and M is a formula containing no quantifiers. $(Q_1x_1)\cdots(Q_nx_n)$ is called the *prenex* and M is called the *matrix* of the formula F.

It is well known that for every formula F, there exists a formula F' in prenex normal form such that F is logically equivalent to F' and an equivalent formula F' can be obtained by translating F with several equivalent translating operations [52, 7].

We next define so-called *Skolem standard forms*, which was introduced by Davis and Putnam [10]. Every formula F in the language is translated into its prenex normal form $(Q_1x_1)\cdots(Q_nx_n)(M)$ without lose of generality. The matrix M, since it does not contain quntifiers, can be transformed into a *conjunctive normal form.* After this transformation, we eliminate the existential quantifiers in the prenex by using Skolem functions. We often say this as *skolemizing* the formula F, which follows the below operations. Suppose Q_r $(1 \leq r \leq n)$ is an existential quantifier in the prenex $(Q_1x_1)\cdots(Q_nx_n)$. If no universal quantifier appears before Q_r, we choose a new constant c different from other constants occurring in M, replace all x_r appearing in M by c, and delete (Q_rx_r) from the prefix. If $Q_{s_1}, Q_{s_2}\ldots, Q_{s_m}$ $(1 \leq s_1 < s_2 < \cdots < s_m < r)$ are all the universal quantifiers appearing before Q_r, we choose a new function f of arity m different from other functions, replace all x_r in M by $f(x_{s_1}, x_{s_2}, \cdots, x_{s_m})$, and delete (Q_rx_r) from the prefix. After the above process is applied to all the existential quantifiers in the prenex, the last formula we obtain is a *Skolem standard form* of the formula F. The constants and functions used to replace the existential variables are called *Skolem functions*.

Every formula can be put in the Skolem standard form, but not every formula has a standard form which is equivalent to the original formula. For example, using the above translating procedure, the formula $(\exists x)F(x)$ is translated to the Skolem standard form $F(c)$ for a new constant c. The original formula implies that there is an element in the extension of the predicate F. In contrast, the translated standard form formula implies that an element associated with the term c is in the extension of the prediacate F. If the latter is true, then the former is true. However, its inverse does not necessarily hold. Hence, the translation into Skolem standard forms can lose the generality. On the other hand, it can affect the inconsistency property in the original formula. Let F be a formula and F_S a Skolem standard form of F. Then it is known that F is inconsistent iff F_S is inconsistent.

This feature is a strong evidence to use Skolem standard forms in automated theorem proving. The task in theorem proving is to decide whether or not $\Sigma \models F$ for a given set of formulas Σ and a target formula F. If true, $\Sigma \cup \{\neg F\}$ should be inconsistent by the contrapositive property of the entailment relation. Hence, it is sufficient for the original task to check the inconsistency of a Skolem standard form of $\Sigma \cup \{\neg F\}$.

For two Skolem standard form formulas, the difference between them occurs only in the part of matrixes. Their matrixes are formalized in conjunctive normal forms which are alternatively defined using the notion of clauses. A *literal* is an atom or the negation of an atom. A *positive literal* is an atom, a *negative literal* is an atom. A *clause* is a finite disjunction of literals which is often identified with the set of its literals. A clause $\{A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m\}$, where each A_i, B_j is an atom, is also written as $B_1 \wedge \cdots \wedge B_m \supset A_1 \vee \cdots \vee A_n$. A *definite clause* is a clause which contains only one positive literal. A *positive* (*negative*) *clause* is a clause whose disjuncts are all positive (negative) literals. A *Horn clause* is a definite clause or negative clause. A *unit clause* is a clause with exactly one literal. The *empty clause*, denoted $\perp$, is the clause which contains no literals.

A *clausal theory* is a finite set of clauses. Note that a clausal theory S can include tautological clauses. Then, τS denotes the set of non-tautological clauses in S. A clausal theory is *full* if it contains at least one non-Horn clause. A *conjunctive normal form* (CNF) formula is a conjunction of clauses, and a *disjunctive normal form* (DNF) is a disjunction of conjunctions of literals. A clausal theory S is often identified with the conjunction of its clauses. *In this thesis, every variable in a clausal theory S is considered governed by the universal quantifier.* By this convension, a Skolem standard form can be simply represented by a clausal theory.

For the sake of simplicity and preserving inconsistency of original formulas, clausal forms have been commonly used for knowledge representation in automated reasoning with first-order logic.

Let S be a clausal theory. Then, how can we check whether or not S is inconsistent? Based on the primary definition of the entailment relation, it is necessary to check all the interpretations with respect to every possible domains, which cannot be achieved in real. For this problem, Herbrand introduced a specific model, called a *Herbrand model*, and proved an important feature that inconsistency of clausal theories can be checked within the notion of Herbrand models. As a preliminary, we define several key notions in the below. Let S be a clausal theory. The *Herbrand universe*, denoted H_S, is the set of all ground terms in S. Note that if no ground term exists in S, U_S consists of a single constant c, say $H_S = \{c\}$. The *Herbrand base*, denoted B_S, is the set of all ground atoms in S.

Let A_S be the alphabet consisting of the constants, variables, functions, and predicates symbols in S. Then, we denote the first-order language given by the alphabet A_S by $\mathcal{L}_S$. A *Herbrand interpretation* I_H of S is an interpretation wrt the language $\mathcal{L}_S$ and the domain U_S such that $c^{I_H} = c$ for each constant c in $\mathcal{L}_S$, $f^{I_H} = f$ for each function f in $\mathcal{L}_S$, and $p^{I_H} \subseteq U_S^n$ for each predicate p in $\mathcal{L}_S$. Let S be a clausal theory and I_H a Herbrand interpretation of S. I_H is a *Herbrand model* iff I_H satisfies S. It is known that for a clausal theory S, S has a model iff S has a Herband model. This property enables us to focus on only the Herbrand models for checking the inconsistency of S.

Let S be a clausal theory. A *ground instance* of a clause C in S is a clause obtained by replacing variables in C by members of the Herbrand universe U_S of S. If S has a model, by the property of Herbrand models, S has also a Herbrand model. In other other, S is true in some Herbrand interpretation I_H of S for every variable assignment h. Note that each variable assignment h maps the set of variables in S into members in the Herbrand universe U_S. Then, this mapping can be regarded as constructing ground instances of clauses in S. Hence, if S has a model, every finite set of ground instances of clauses in S should be also true in some Herbrand interpretation I_H. By taking the contrapositive of this statement, if there is a finite set S' of ground instances of clauses in S such that S' is false in every Herbrand interpretation I_H (that is, S' is inconsistent), then S does not have any model (that is, S is inconsistent). Interestingly, this inverse also holds, which Herbrand has firstly proved.

Theorem 2.1 (Herbrand's theorem)**.** Let S be a clausal theory. S is inconsistent iff there is a finite set of ground instances of clauses in S.

2.3 Consequence Finding

In the previous section, we introduced clausal forms that are commonly used in Inductive Logic Programming as well as Herbrand's theorem. One importance issue that this theorem brings is the fact that the consistency of a target clausal theory can be systematically decided by checking the consistency of each possible finite set S' of ground instances from S. The consistency of each S' can be determined in the finite number of checking. This propositional approach based on Herbrand's theory is one possible way to realistically achieve the task of automated theorem proving. Given a clausal theory S, the essential problem in automated *deductive* reasoning lies in how we can find the logical consequences of S. In case that S is inconsistent, this problem is reduced to find the refutation, that is, the empty clause. For the consequence finding problem, it is difficult to use the propositional approach since logical consequences are not necessarily ground. Instead, we can use so-called *resolution* principle, which was introduced by Robinson. In this section, we first define key notions such as *substitutions* and *subsumption*, next describe the resolution principle in brief, and lastly introduce the *Subsumption* theorem which shows the completeness of the resolution principle for consequence finding.

Let C and D be two clauses. C *subsumes* D, denoted $C \succeq D$, if there is a substitution θ such that $C\theta \subseteq D$. C *properly* subsumes D if $C \succeq D$ but $D \not\succeq C$. For a clausal theory S, μS denotes the set of clauses in S not properly subsumed by any clause in S. Let S_1 and S_2 be two clausal theories. S_1 *subsumes* S_2, denoted $S_1 \succeq S_2$ iff for any clause $D \in S_2$, there exists a clause $C \in S_1$ such that $C \succeq D$. In particular, S_1 *strongly subsumes* S_2, denoted $S_1 \succeq^{\natural} S_2$ iff $S_1 \succeq S_2$ and for any clause $C \in S_1$, there exists a clause $D \in S_2$ such that $C \succeq D$. the relation $\succeq^{\natural}$ is known as *Plotkin's order* [56]

Let C_1 and C_2 be two clauses (called *parent clauses*) with no variables in common. Let L_1 and L_2 be two literals in C_1 and C_2, respectively. If L_1 and $\neg L_2$ have a most general unifier σ, then the clause

$$(C_1\sigma - \{L_1\sigma\}) \cup (C_2\sigma - \{L_2\sigma\})$$

is called a (*binary*) *resolvent* of C_1 and C_2, denote $Res(C_1, C_2)$. The literals L_1 and L_2 are called the *literals resolved upon.* Let S be a clausal theory and C be a clause. A *derivation* of C from S is a finite sequence of clauses $R_1, \ldots, R_k = C$ such that each R_i is either in S, or is a resolvent of two clauses in $R_1, \ldots, R_{i-1}$. S is *minimal wrt derivation* iff for each clause $C \in S$, there is no D such that a derivation from $S - \{C\}$ can be constructed to D and $D \succeq C$. S is *maximal wrt derivation* iff for each consequence C of S, there exists a clause D in S such that $D \succeq C$.

We then recall the following result, called the *Subsumption theorem.*

Theorem 2.2 ([37, 52])**.** Let S be a clausal theory and C be a clause. Then $S \models C$ iff C is a tautology or there exists a derivation of a clause D from S that subsumes C.

For a clausal theory S, a consequence of S is a clause entailed by S as we defined in the previous section. We denote by $Th(S)$ the set of all consequences of S. The Subsumption theorem states the completeness of the resolution principle for finding any non-tautological consequences in $\mu Th(S)$.

By the Subsumption theorem, the resolution principle is sufficient to compute the non-tautological consequences $\mu Th(S)$ for a target clausal theory S. *SLD resolution* [38] is a well known deduction technique as the key procedure in Prolog. Though SLD resolution can be used to theorem proving, we cannot use it for consequence finding. Besides, though SLD resolution preserves the soundness and completeness for finding refutations, its usage is limited in Horn representation formalization. Inoue [25] proposed to use a restricted but powerful consequence finding technique to overcome this problem. The essential idea lies in so-called *characteristic clauses*, which are "interesting" consequences that users wish to obtain. Characteristic clauses are declaratively defined with a language bias, called a *production field*. Using a relevant production field, we can efficiently restrict the search space for hypothesis finding. We first formally define the key notions as follows:

A production field $\mathcal{P}$ is defined as a pair, $\langle \mathbf{L}, Cond \rangle$, where $\mathbf{L}$ is a set of literals closed under instantiation, and $Cond$ is a certain condition to be satisfied, e.g., the maximum length of clauses, the maximum depth of terms, etc. Note that if $\mathbf{L}$ is closed under instantiation. A production field $\mathcal{P}$ is *stable* if for any two clauses C and D such that C subsumes D, D belongs to $\mathcal{P}$ only if C belongs to $\mathcal{P}$. For instance, the production field $\mathcal{P} = \langle \{p(a)\} \rangle$ is not stable, since there is a clause $p(X)$ such that $p(X)$ does not belong to $\mathcal{P}$, but the ground clause $p(a)$ that is subsumed by $p(X)$ belong to $\mathcal{P}$. Note that there is recent a work to make any unstable production fields stable [62].

Let $\mathcal{R}$ be the set of all predicate symbols appeared in the language. For $\mathbf{R} \subseteq \mathcal{R}$, we denote by $\mathbf{R}^+$ ($\mathbf{R}^-$) the positive (negative) occurences of predicates from $\mathbf{R}$ in the language.

A clause C *belongs* to $\mathcal{P} = \langle \mathbf{L}, Cond \rangle$ if every literal in C belongs to $\mathbf{L}$ and C satisfies $Cond$. For a set S of clauses, the set of consequences of S belonging to $\mathcal{P}$ is denoted by $Th_{\mathcal{P}}(S)$. Then, the characteristic clauses of S wrt $\mathcal{P}$ are defined as:

$$Carc(S, \mathcal{P}) = \mu Th_{\mathcal{P}}(S)$$

Note that $Carc(S, \mathcal{P})$ can, in general, include tautological clauses [25].When

$Cond$ is not specified, $\mathcal{P}$ is simply denoted as $\langle \mathbf{L} \rangle$. If both $\mathbf{L}$ and $Cond$ are not specified, $\mathcal{P}$, denoted as $\langle \mathcal{L} \rangle$, allows all the formulas in the first-order language $\mathcal{L}$. Note that for the production field $\mathcal{P} = \langle \mathcal{L} \rangle$, $Carc(S, \mathcal{P}) = \mu Th(S)$.

Example 2.1. Let a clausal theory S and a (stable) production field $\mathcal{P}$ be as follows:

$$S = \{s(X_0),\ p(X_1) \vee \neg q(X_1) \vee \neg s(Y_1),\ p(X_2) \vee q(X_2) \vee r(X_2)\},$$
$$\mathcal{P} = \langle \{\pm p,\ \pm q,\ \pm r\},\ max\_length \leq 3 \rangle.$$

Note that for a predicate symbol p of arity k, $\pm p$ means the complementary literals $p(X_1, \ldots, X_k)$ and $\neg p(X_1, \ldots, X_k)$ and the condition $max\_length \leq k$ for some natural number k means the number of literals in every characteristic clauses is less than or equal to k. Using the resolution principle, we have three additional consequences derived from S :

$$C_1 = p(X_3) \vee \neg q(X_3),\ C_2 = p(X_4) \vee r(X_4) \vee \neg s(Y_1),\ C_3 = p(X_5) \vee r(X_5).$$

Whereas the number of literals in every clause in S and those three clauses is less than or equal to 3, only C_1, C_3 and the 3rd clause in S belong to $\mathcal{P}$ as the other contains the predicate s that is not allowed to be included. In addition, the following tautological clauses also belong to $\mathcal{P}$:

$$C_4 = p(X_6) \vee \neg p(X_6),\ C_5 = q(X_7) \vee \neg q(X_7),\ C_6 = r(X_8) \vee \neg r(X_8).$$

Then, $Th_{\mathcal{P}}(S)$ consists of those 6 clauses. On the other hand, both the 3rd clause in S and C_2 are properly subsumed by C_3. As a result, the characteristic clauses of S wrt $\mathcal{P}$ are only C_1, C_3, C_4, C_5 and C_6.

When a new clause F is added to a clausal theory, some consequences are newly derived with this additional information. The set of such clauses that belong to the production field are called *new characteristic clauses*. Formally, the *new characteristic clauses* of F wrt S and $\mathcal{P}$ are defined as:

$$NewCarc(S, F, \mathcal{P}) = Carc(S \cup \{F\}, \mathcal{P}) - Carc(S, \mathcal{P}).$$

Example 2.2. Recall Example 2.1. Let a new clause F be $\neg p(Z_1)$. Then, $Th_{\mathcal{P}}(S \cup \{F\})$ consists of $Th_{\mathcal{P}}(S)$ and the following three consequences derived from S together with F:

$$C_7 = \neg q(Z_1) \vee \neg s(Y_1),\ C_8 = q(Z_1) \vee r(Z_1),\ C_9 = \neg q(Z_2),\ C_{10} = r(Z_3).$$

Since C_9 properly subsumes both C_7 and C_8, it holds that $Carc(S \cup \{F\}, \mathcal{P}) = \mu[\{C_9,\ C_{10}\} \cup Th_{\mathcal{P}}(S)]$. Hence, $NewCarc(S, F, \mathcal{P})$ consists of C_9 and C_{10}.

For computation of the (new-)characteristic clauses, Inoue [25] proposed a sound and complete resolution procedure, called *SOL* resolution. SOL resolution constructs derivations from F with respect to S based on three inference rules: *Skip*, *Extension* and *Reduction*. Note that SLD resolution consists of Extension and Reduction operations. Hence, SOL resolution can be regarded as an extension of SLD resolution constructed by adding Skip operation to SLD resolution.

Let S be a clausal theory and F a clause. We call a derivation obtained by SOL resolution as an *SOL-deduction* from $S + F$ and $\mathcal{P}$.

Theorem 2.3 (The soundness and completeness of SOL-deductions [29])**.** The soundness and completeness of SOL-deductions are formally described as follows:

1. Soundness: if a clause D is derived by an SOL-deduction from $S + F$ and $\mathcal{P}$, then D belongs to $Th_{\mathcal{P}}(S \cup \{F\})$.

2. Completeness: if a clause E does not belong to $Th_{\mathcal{P}}(S)$ but belongs to $Th_{\mathcal{P}}(S \cup \{F\})$, then there is an SOL-deduction of a clause D from $S + F$ and $\mathcal{P}$ such that D subsumes E.

In terms of efficient implementation on SOL resolution, there is an available system, called *SOLAR* [29, 51]. SOLAR (SOL for Advanced Reasoning) uses a connection tableau format embedding the three operations to efficiently compute new characteristic clauses.

Example 2.3. We recall Example 2.2. SOLAR constructs a SOL-deduction of $C_{10} = r(Z_3)$ from $S + F$ and $\mathcal{P}$ with the following tableau in Figure 2.1:

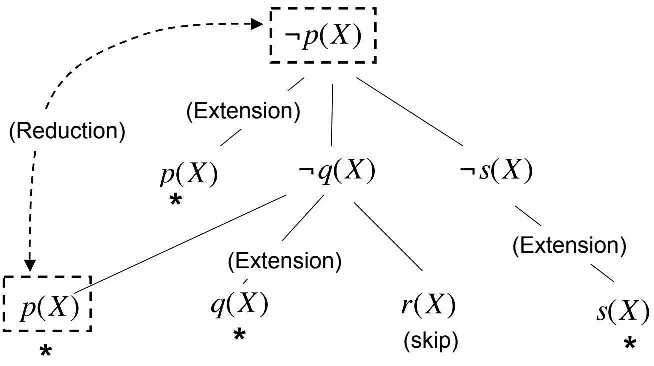


Figure 2.1: Tableau in SOLAR

2.4 Dualization

In this section, we review issues on the *dualization* problem. The dualization problem is a well-known problem in NP-completeness [16]. It needs huge computational costs, whereas hypothesis finding techniques such as CF-induction often need it. Thus, it is essential to use efficient algorithms for real-world applications of those hypothesis-finding techniques that use dualization. After describing basic notion and terminologies, we introduce an efficient algorithm in a specific dualization problem, called *monotone* dualization.

The dualization problem [15, 16, 68] is to compute a CNF formula from the *dual* ψ^d of a given CNF formula ψ. Let ψ be a CNF formula. The *dual* ψ^d is the DNF formula obtained by replacing every conjunction symbol $\wedge$ ($\vee$) with the disjunction symbol $\vee$ ($\wedge$) in ψ. Though there are many CNF formulas equivalent to ψ^d in general, we are used to find a *prime* and *irredundant* CNF formula. A CNF formula S is *prime* if for every $C \in S$, there is no literal $l \in C$ such that $(S - \{C\}) \cup \{C - \{l\}\} \equiv S$. A CNF formula S is *irredundant* if there is no clause $C \in S$ such that $S - \{C\} \equiv S$.

Example 2.4. Let ψ_1 and ψ_2 be as following CNF formulas:

$$\psi_1 = (a \vee b) \wedge (c \vee \neg b) \wedge (a \vee c),$$
$$\psi_2 = (a \vee b) \wedge (a \vee \neg b).$$

ψ_1 is prime, since for every literal l in ψ_1, the clause obtained by dropping l is no longer logically equivalent to ψ_1. On the other hand, ψ_1 is redundant, since the third clause in ψ_1 is the resolvent of the other two clauses. In contrast, ψ_2 is irredundant, since for ever clause C in ψ_2, $\psi_2 - \{C\}$ is no longer logically equivalent to ψ_2. On the other hand, ψ_2 is non-prime, since two literals b and $\neg b$ can be dropped from ψ_2 without losing the logical equivalence.

Let ψ be a CNF formula. ψ is *monotone* iff ψ does not include any negations. We call dualization of monotone CNF formulas as *monotone dualization.* An arbitrary monotone CNF formula ψ satisfies two nice properties: The first is that ψ has its *unique* prime CNF. The second is that if ψ is prime, then ψ is irredundant. These two properties imply that the monotone dualization problem is to finding the unique prime CNF formula. In contrast, a non-monotone CNF formulas does not necessarily have its unique prime-irreducndant CNF, shown as the following example:

Example 2.5. Let three CNF formula ψ_1, ψ_2 and ψ as follows:

$$\begin{aligned}
\psi_1 &= (a \vee b) \wedge (\neg b \vee c) \wedge (\neg c \vee \neg a),\\
\psi_2 &= (a \vee c) \wedge (\neg b \vee \neg a) \wedge (\neg c \vee b),\\
\psi &= \psi_1 \wedge \psi_2.
\end{aligned}$$

Both ψ_1 and ψ_2 are prime and irredundant CNF formulas logically equivalent to ψ. Thus, the prime-irredundant CNF of ψ is not indeed unique.

It is well known that the monotone dualization is equivalent to the minimal hitting sets enumeration [17], which is solvable in quasi-polynominal time [19]. We then here introduce an efficient algorithm [78, 68] for this enumeration problem. To describe the algorithm, we first give the definition of a minimal hitting set of a given family of sets.

Definition 2.6 ((Minimal) Hitting set)**.** Let Π be a finite set and $\mathcal{F}$ be a subset family of Π. A *hitting set* HS of $\mathcal{F}$ is a set such that for every $S \in \mathcal{F}$, $S \cap HS \neq \emptyset$. A set MHS is a *minimal hitting set* of $\mathcal{F}$ if MHS satisfies the following two conditions:

1. MHS is a hitting set of $\mathcal{F}$;

2. For every subset $MHS' \subseteq MHS$, if MHS' is a hitting set of $\mathcal{F}$, then $MHS' = MHS$.

As a property in minimal hitting sets, the following theorem holds:

Theorem 2.4. [78] Let $\mathcal{F}$ be a family of sets. A set E is a minimal hitting set of $\mathcal{F}$ iff for every element $e \in E$, there exists a set F in $\mathcal{F}$ such that

$$E \cap F = \{e\}.$$

Example 2.6. Let a monotone CNF formula ψ be $(a \vee b) \wedge (a \vee c)$ and a subset family $\mathcal{F}_\psi$ $\{\{a,\ b\},\ \{a,\ c\}\}$. Note that each set in $\mathcal{F}_\psi$ can be identified with a conjunction of ψ. We first show the correspondence between the dualization of ψ and the miniminal hitting set enumeration of $\mathcal{F}_\psi$. The dual ψ^d is the DNF $(a \wedge b) \vee (a \wedge c)$. The prime implicants of ψ^d are two clauses a and $b \vee c$. Then, the output of the dualization of ψ is the prime CNF $\phi = a \wedge (b \vee c)$. In contrast, the minimal hitting sets of $\mathcal{F}_\psi$ contains two sets $\{a\}$ and $\{b,\ c\}$. We then notice that each of which can be identified with a conjunction of ϕ.

This $\mathcal{F}_\psi$ follows Theorem 2.4. When we consider the minimal hitting set $E_1 = \{a\}$ of $\mathcal{F}_\psi$, there is in fact a set $F_1 = \{a,\ b\} \in \mathcal{F}_\psi$ such that $E_1 \cap F_1 = \{a\}$. When we consider another minimal hitting set $E_2 = \{b,\ c\}$, for each element b and c, there are two sets $F_1 = \{a,\ b\}$ and $F_2 = \{a,\ c\}$ in $\mathcal{F}_\psi$ such that $E_2 \cap F_1 = \{b\}$ and $E_2 \cap F_2 = \{c\}$, respectively.

An efficient algorithm that we introduce here is based on the principle of *reverse search* [2] for enumeration problems. In reverse search, we define a relevant relation, called a *parent-child relationship* over the solution space (i.e. the minimal hitting sets of a given subset family). Note that this relationship has to satisfy the condition that for every solution, there is no ancestor on the relationship, such that is equal to the original solution. Such a relationship that satisfies this condition can lead a rooted-tree, called a *enumeration tree*, on the solution space. In general, the size of an enumeration tree is too big to store it in memory for searching. Instead, we use a relevant algorithm that enables us to enumerate only the children of any node in the tree. Using this algorithm, we can search all the solutions with the depth-first strategy, while we do not need to store the whole enumeration tree in memory. The parent-child relationship presented in [78] is defined as follows.

Definition 2.7 (Parent-child relationship [78])**.** Let Π be a finite set, $\mathcal{F}_n = \{F_1, \ldots, F_n\}$ a subset family of Π that contains n elements and E_i $(1 \leq i \leq n)$ a minimal hitting set of $\mathcal{F}_i$. Then a pair $(i+1, E_{i+1})$ is a *child* of a pair (i, E_i) if E_{i+1} satisfies the following condition:

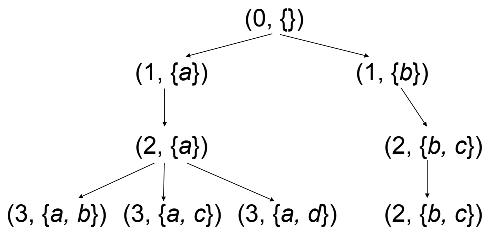
- If E_i is a minimal hitting set of $\mathcal{F}_{i+1}$, then $E_{i+1} = E_i$.
- Else, $E_{i+1} = E_i \cup \{e\}$, where e is an element in F_{i+1} such that $E_i \cup \{e\}$ is a minimal hitting set of $\mathcal{F}_{i+1}$.

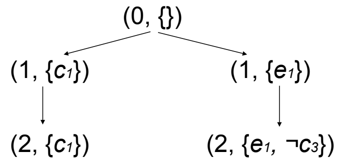
For a child pair $(i+1, E_{i+1})$, its parent (i, E_i) is never equal to the child, since the left number $i+1$ of the child is different from its parent's one (that is i). Analogically, this relationship satisfies that for every child pair, there is no ancestor such that is equal to the original.

Based on this parent-child relationship, the efficient algorithm for enumerating the minimal hitting sets [78, 68] is described as follows:

Global $\mathcal{F}_n = \{F_1, \cdots, F_n\}$
compute(i, mhs, S)/*mhs is a minimal hitting set of $\mathcal{F}_i$**,**
and S is the family of minimal hitting sets of $\mathcal{F}_n$*/
Begin
if i == n then add mhs to S and return;
else if mhs is a minimal hitting set of $\mathcal{F}_{i+1}$ do
    compute($i+1$, mhs, S);
else for every $e \in F_{i+1}$ s.t. $mhs \cup \{e\}$ is a minimal hitting set of $\mathcal{F}_{i+1}$
    do *compute*($i+1$, $mhs \cup \{e\}$, S);
output S and return;
End

Example 2.7. Let a subset family $\mathcal{F}_3$ be $\{\{a, b\}, \{a, c\}, \{b, c, d\}\}$. Then, the minimal hitting sets are computed with the depth-first search on the enumeration tree in the left below figure. As a result, we can obtain the set $S = \{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}\}$ that are the minimal hitting sets of $\mathcal{F}_3$. Next, suppose a family $\mathcal{F}'_2$ of two sets $\{c_1, e_1\}$ and $\{c_1, \neg c_3\}$, which can be regarded as two sets of conjuncts of the DNF formulas $\neg C$ and $\neg E$ in Example 5.3, respectively. Then, the minimal hitting sets of $\mathcal{F}'_2$ are computed using the enumeration tree in the right below figure, and we can obtain the output family $\{\{c_1\}, \{e_1, \neg c_3\}\}$.





Notice that $\mathcal{F}_3$ in Example 2.7 allows negations to be included in inputs.

This enumeration algorithm can be applied in clausal form logic where we often deal with negations of theories. Consider a ground clausal theory S and its negation $\neg S$. By De Morgan's laws, $\neg S$ is logically equivalent to the DNF formula obtained by replacing every literal l in the dual S^d with $\neg l$. Hence, the dualization of S can be computationally regarded as the translation of the DNF formula $\neg S$ into some CNF formula in clausal form logic.

There are several CNF formulas that are logically equivalent to $\neg S$. Hereafter, we introduce three kinds of CNF formulas. Let S be a ground clausal theory $\{C_1, C_2, \ldots, C_n\}$ where C_i $(1 \leq i \leq n) = l_{i,1} \vee l_{i,2} \vee \cdots \vee l_{i,m_i}$. The *complement* of S, denoted by $\overline{S}$, is defined as follows:

$$\overline{S} = \left\{ \quad \neg l_{1,k_1} \vee \neg l_{2,k_2} \vee \cdots \vee \neg l_{n,k_n} \quad \middle| \quad \begin{array}{c} 1 \leq k_1 \leq m_1,\ 1 \leq k_2 \leq m_2, \\ \ldots,\ 1 \leq k_n \leq m_n \end{array} \right\}.$$

Note that $\overline{S}$ is a clausal theory such that $\overline{S} \equiv \neg S$. Accordingly, $\overline{S}$ can be regarded as such a CNF formula that is logically equivalent to $\neg S$. We can also define alternative equivalent clausal theories. The *residue* complement of S, denoted $R(S)$, is the clausal theory obtained by removing any tautological clauses in the complement $\overline{S}$. The *minimal* complement of S, denoted $M(S)$, is the clausal theory obtained by removing any clauses in the complement $\overline{S}$ each of which is properly subsumed by another clause in $\overline{S}$. We may notice that the residue and minimal complements can be represented as $\tau\overline{S}$ and $\mu\overline{S}$, respectively. $\tau\overline{S}$ and $\mu\overline{S}$ are also CNF formulas logically equivalent to $\neg S$. We often denote $\tau\overline{S}$ and $\mu\overline{S}$ as the functions $R(S)$ and $M(S)$, called the *residue* and *minimal* complement of S, respectively. $R^2(S)$ and $M^2(S)$ denotes $R(R(S))$ and $M(M(S))$, respectively.

In terms of computation of the minimal complement, it is interestingly equal to the enumeration of minimal hitting sets. We show this correspondence in the following. Let S be a CNF formula consisting of ground literals. In the following, $\mathcal{F}(S)$ denotes the family of sets $\{C_1,\ C_2,\ \ldots,\ C_n\}$ where C_i $(1 \leq i \leq n)$ is the set of literals in each conjunction of the DNF formula $\neg S$. Similarly, given a DNF formula S, $\mathcal{F}(S)$ denotes the family of sets $\{C_1,\ C_2,\ \ldots,\ C_n\}$ where C_i $(1 \leq i \leq n)$ is the set of literals in each disjunction of the CNF formula $\neg S$. Note that given a ground clausal theory S, $\mathcal{F}(\mathcal{F}(S))$ corresponds to S.

The number of minimal hitting sets of the family $\mathcal{F}$(S) is finite. We then consider the ground clausal theory consisting of all the minimal hitting sets of $\mathcal{F}(S)$, denoted $MHS(S)$. Then $MHS(S)$ is equal to the minimal complement $M(S)$ as follows:

Theorem 2.5. Let S be a ground clausal theory. Then, $M(S) = MHS(S)$.

(*Proof* of $MHS(S) \subseteq M(S)$.) Let E be a minimal hitting set of $\mathcal{F}(S)$. We show $E \in \mu\overline{S}$ since $\mu\overline{S} = M(S)$ by the definition of minimal complements. By Theorem 2.4, for each literal $e_i \in E$ $(1 \leq i \leq n)$, there exists an element $f_i \in \mathcal{F}(S)$ such that $f_i \cap E = \{e_i\}$. We denote by F_E the subset $\{f_1, \ldots, f_n\}$ of $\mathcal{F}(S)$. Then E is constructed by selecting e_i from each element in F_E. In contrast, by the definition of the complement, each clause in $\overline{S}$ is constructed by selecting one literal l from every element in $\mathcal{F}(S)$. Since E is a minimal hitting set of $\mathcal{F}(S)$, for each element f' in $\mathcal{F}(S) - F_E$, $f' \cap E \neq \emptyset$ holds. Hence, E can be constructed by selecting one literal from every element f_i of $\mathcal{F}(S)$ in such a way that if $f_i \in F_E$, then a certain literal $e_i \in E$ is selected, otherwise, any literal $e_i \in f_i \cap E$ is selected. Therefore, $E \in \overline{S}$ holds. Suppose that $E \not\in \mu\overline{S}$. Then there is a clause $D \in \overline{S}$ such that $D \subset E$. Since $D \in \overline{S}$, D is a hitting set of $\mathcal{F}(S)$. However, this contradicts that E is minimal.

Hence, $E \in \mu\overline{S}$ holds.
(*Proof* of $M(S) \subseteq MHS(S)$.) Suppose that there is a clause $D \in M(S)$ such that $D \notin MHS(S)$ (*). Since $D \in \mu\overline{S}$ and $\mu\overline{S} \subseteq \overline{S}$, $D \in \overline{S}$ holds. By the definition of $\overline{S}$, D satisfies that $C \cap D \neq \emptyset$ for every $C \in \mathcal{F}(S)$. Hence, D is a hitting set of $\mathcal{F}(S)$. Accordingly, there is a clause D' in $MHS(S)$ such that $D' \subseteq D$. Since we assume that $D \notin MHS(S)$, $D \neq D'$ holds as $D' \in MHS(S)$. Then, $D' \subset D$ holds. Since $D' \in MHS(S)$ and $MHS(S) \subseteq M(S)$, $D' \in M(S)$ holds. Hence, there is a clause $D' \in M(S)$ such that D' properly subsumes the clause D in $M(S)$. This contradicts the minimality of $M(S)$. Hence, the primary assumption (*) is false. Therefore for every clause $D \in M(S)$, $D \in MHS(S)$ holds. $\square$

Example 2.8. Let S be the clausal theory $\{a \vee \neg b,\ \neg b \vee \neg c,\ \neg b \vee \neg d\}$. Then, $\mathcal{F}(S)$, $MHS(S)$, $\overline{S}$ and $M(S)$ are represented as follows:

$$\begin{aligned}
&\mathcal{F}(S) = \{\{\neg a,\ b\},\ \{b,\ c\},\ \{b,\ d\}\},\\
&MHS(S) = \{\{\neg a,\ c,\ d\},\ \{b\}\},\\
&\overline{S} = \{\neg a \vee b,\ \neg a \vee b \vee d,\ \neg a \vee c \vee b,\ \neg a \vee c \vee d,\ b,\ b \vee d,\ b \vee c,\ b \vee c \vee d\},\\
&M(S) = \{\neg a \vee c \vee d,\ b\}.
\end{aligned}$$

Based on Theorem 2.5, we can use the efficient algorithm for enumerating the minimal hitting sets in order to compute minimal complements.

Chapter 3

Inverse Entailment and CF-induction

In this chapter, we review the previously proposed ILP methods based on the principle of inverse entailment (IE) [44], including *CF-induction* [26] which is sound and complete for finding hypotheses in full-clausal theories. Each IE-based method has its own properties in terms of search strategy and computational efficiency. After describing these properties in Section 3.1, we focus on CF-induction and investigate its inherent possibility for achieving an advanced inference integrating abduction and induction in Section 3.2.

3.1 Hypothesis Finding in Inverse Entailment

We formally give the definition of a hypothesis H in the problem setting of explanatory induction as follows:

Definition 3.1 (Hypothesis)**.** Let B and E be clausal theories, representing a background theory and (positive) examples/observations, respectively. Let H be a clausal theory. Then H is a *hypothesis* wrt B and E if H satisfies that $B \wedge H \models E$ and $B \wedge H$ is consistent. To ensure that the inductive task is not trivial, we assume $B \not\models E$ throughout this thesis. We refer to a hypothesis instead of a hypothesis wrt B and E if no confusion arises.

Hypothesis finding in Definition 3.1 is logically equivalent to seeking a consistent hypothesis H such that $B \wedge \neg E \models \neg H$. Using this alternative condition, IE-based procedures [26, 31, 44, 61, 63, 75, 83] compute a hypothesis H in two steps. First, they construct an intermediate theory F such that F is ground and $B \wedge \neg E \models F$. Second, they generalize its negation $\neg F$ into hypotheses using some inverse relation on entailment such that $H \models \neg F$. Hereafter, we call F the *bridge theory* wrt B and E as follows.

Definition 3.2 (Bridge theory)**.** Let B and E be a background theory and observations, respectively. Let F be a ground clausal theory. Then F is a *bridge theory* wrt B and E if $B \wedge \neg E \models F$. IF no confusion arises, a bridge theory wrt B and E will simply be called a bridge theory.

Every IE-based procedure constructs a bridge theory and generalizes its negation into a hypothesis in its own way.

Progol [44, 46], one of the state of the art ILP systems, uses the technique of *Bottom Generalization*. Its bridge theory F corresponds to the conjunction of ground literals each of which is derived from $B \wedge \neg E$. After constructing $\neg F$, called the *bottom clause*, Progol generalizes it with the inverse relation of subsumption, instead of entailment. Note that recently this generalization operator has been improved by introducing the notion of clause ordering [75]. Progol has been successfully applied to many practical problems so far. Indeed, the two application problems on protein secondary structure predictions and estimation of inhibitory effects on metabolic pathways, which we explained in Chapter 1, have been tested using Progol. On the other hand, it is known that the procedure of Progol is sound, but incomplete for finding hypotheses [82]. Beside, since it is developed based on Prolog, its knowledge representation formalisms are limited to Horn clausal theories. The language bias used in Progol is called *mode declaration*, which can restrict the literals and the appearance positions of common variables in the target hypotheses. One important remark on Progol is that it searches *one* clause that will be included in the output hypothesis for each observation. Hence, if the observations should contain only one clause, the output hypothesis is the form of some single clause. In other words, Progol cannot derive multiple clauses as a hypothesis for each observation.

HAIL [61] constructs so-called *Kernel Sets* to overcome some limitation on Bottom generalization. Each clause C_i in a Kernel Set $\{C_1, \ldots, C_n\}$ is given by $B^i_1 \wedge \cdots \wedge B^i_{m_i} \supset A^i$, where $B \cup \{A^1, \ldots, A^n\} \models E$ and $B \models \{B^1_1, \ldots, B^n_{m_n}\}$. After constructing a Kernel Set, HAIL generalizes it using the inverse relation of subsumption like Progol. Note that a Kernel Set is regarded as the negation of a certain bridge theory F. In other words, they directly construct $\neg F$ by separately computing head and body literals of each clause in $\neg F$. FC-HAIL [63] and XHAIL [60] are extensions of HAIL to use richer knowledge representation formalisms that allow full clausal theories and Nagation as Failure (NAF), respectively. Like Progol, these HAIL series are also sound, but incomplete for finding hypotheses.

We remark there is a recent work to extend Kernel Sets with an iterative procedure [31] in order to generate a multiple clauses hypothesis in response to a single seed example, which was one limitation of Progol. In this work, Kimber et al. [31] proposed so-called *Connected Theory* generalization, which can be viewed as an extension of Kernel Set generalization of HAIL. A Connected Theory $T = T_1 \cup \cdots \cup T_n$ is a Horn clausal theory such that (1) $B \wedge T^+_1 \models E^+$, (2) $B \wedge T^+_{i+1} \models T^-_i$ $(1 \leq i < n)$, (3) $B \models T^-$ and (4) $B \wedge T \not\models \Box$. Note that for a Horn clausal theory T, T^+ (*resp.* T^-) denotes the set of Horn clauses each of which contains only the head literal (*resp.* body literals) of a Horn clause in T. After iteratively constructing a Connected theory, they generalize it with the inverse of subsumption. Hence, a Connected theory, like HAIL, is regarded as the negation of a certain bridge. They develop so-called *Imparo* based on Connected Theory generalization.

The *residue procedure* [83] has been firstly proposed to find hypotheses in full clausal theories. This procedure constructs a bridge theory F consisting of ground instances from $B \wedge \neg E$. It then computes the residue complement $R(F)$, and generalizes it with the inverse of subsumption. In contrast, *CF-induction* [26], which is sound and complete for finding hypotheses in full clausal theories, constructs a bridge theory F consisting of ground instances from characteristic clauses of $B \wedge \neg E$. As we explained in Chapter 2, each characteristic clause is a subsume-minimal consequence of $B \wedge \neg E$ that satisfies a given language bias. In this sense, CF-induction actively approach $\neg H$ from $B \wedge \neg E$ as far as possible by means of the consequence finding technique (i.e. SOL resolution). After constructing a bridge theory F, CF-induction translates $\neg F$ into a CNF formula and generalizes it with inverse entailment. Unlike the other IE-based methods, generalization in CF-induction is based on the inverse relation of entailment. This feature becomes the key aspect to ensure the completeness of CF-induction. On the other hand, CF-induction needs several computationally expensive procedures such as consequence finding, dualization, and generalization based on inverse entailment. Figure 3.1 describes the characteristics for each IE-based ILP method: In the next section, we investigate more detail of CF-induction.

Method	**Formalisms**	**language**	**Bridge theory**	**Generalization relation**	**Valid ness**
Progol [Muggleton, 95; 00; Tamaddoni *et al*, 08]	Horn	Mode declaration	Bottom clause	Subsumption	Sound
Residue Procedure [A. Yamamoto & Fronhofer, 00]	Full	×	Instances from B and -E	Subsumption	Sound
HAIL [Ray et al, 03]	Horn	Mode declaration	Kernel set	Subsumption	Sound
CF-induction [Inoue, 04; Y. Yamamoto *et al*, 08]	Full	Production field	Instances from consequences of B and -E	Entailment	Sound and Complete
FC-HAIL [Ray & Inoue, 07]	Full	Mode declaration	Kernel set	Subsumption	Sound
X-HAIL [Ray, 09]	Horn with NAF	Mode declaration	Kernel set	Subsumption	Sound
Imparo [Kimber *et al*, 09]	Horn	Mode declaration	Connection theory	Subsumption	Sound

Figure 3.1: Characteristics in IE-based Methods

3.2 CF-induction

Computation of CF-induction is based on the principle of inverse entailment. As we described before, recently, IE methods have been developed for full clausal theories to enable the solution of more complex problems in richer knowledge representation formalisms. One such method is CF-induction [26], which has two important benefits: unlike some related systems, such as FC-HAIL [63], CF-induction is complete for finding full clausal hypotheses; and unlike other related systems, such as the Residue Procedure [83], CF-induction can exploit language bias to focus the procedure on some relevant part of the search space specified by the user. In the previous version, we have assumed the production field $\mathcal{P} = \langle \mathbf{L}, max\_length \rangle$ where $\mathbf{L}$ is a set of literals reflecting an *inductive bias* whose literals are the negations of those literals we wish to allow in hypothesis clauses. When no inductive bias is considered, $\mathcal{P}$ is just set to $\langle \mathcal{L} \rangle$, which allows all the formulas in the first-order language $\mathcal{L}$.

Definition 3.3. [Hypothesis wrt B, E and $\mathcal{P}$] Let B, E and $\mathcal{P}$ be a background theory, observations and a production field, respectively. Let H be a clausal theory. H is a *hypothesis wrt B, E and $\mathcal{P}$* iff H is a hypothesis wrt B and E, and satisfies the condition that for every literal l appearing in H, its negation $\neg l$ is in $\mathbf{L}$. If no confusion arises, we simply call a hypothesis wrt B, E and $\mathcal{P}$ as a hypothesis.

Then, for every hypothesis H wrt B, E and $\mathcal{P}$, the following holds:

$$
\begin{aligned}
B \wedge \overline{E_{sk}} &\models Carc(B \wedge \overline{E_{sk}}, \mathcal{P}) \models \neg H, &(3.1)\\
B &\models Carc(B, \mathcal{P}) \qquad \not\models \neg H, &(3.2)
\end{aligned}
$$

where E_{sk} is the ground clausal theory obtained by skolemizing[1] E. The two formulas above follow from the principle of IE and the definition of characteristic clauses. To explain them, we recall Formula (1.5) and the consistency condition[2] of B and H as follows:

$$
\begin{aligned}
B \wedge \neg E &\models \neg H,\\
B &\not\models \neg H.
\end{aligned}
$$

Since $\overline{E_{sk}} \models \neg E$, it holds that $B \wedge \overline{E_{sk}} \models B \wedge \neg E$. Since $B \wedge \neg E \models \neg H$, it also holds that $B \wedge \overline{E_{sk}} \models \neg H$. Then, by Herbrand's theorem, for this hypothesis

[1]See P. 29 for more detail.

[2]See P. 28 for representing the consistency condition of two sets of formulas.

H, there is a ground clausal theory H_g such that H_g consists of ground instances from H and $B \wedge \overline{E_{sk}} \models \neg H_g$. Since any literal in $\neg H$ is included in $\mathbf{L}$ of $\mathcal{P} = \langle \mathbf{L} \rangle$, $\neg H_g$ belongs to $\mathcal{P}$. Then, by the definition of characteristic clauses, $Carc(B \wedge \overline{E_{sk}}, \mathcal{P}) \models \neg H_g$ holds. Since $H \models H_g$, $\neg H_g \models \neg H$ holds. Hence, Formula (3.1) holds. In turn, since $B \models Carc(B, \mathcal{P})$ holds, it must hold $Carc(B, \mathcal{P}) \not\models \neg H$ to satisfy $B \not\models \neg H$. Hence, Formula (3.2) also holds.

Formula (3.1) implies that we can use characteristic clauses to construct bridge theories for IE. Formula (3.2) is used to ensure the consistency of the hypothesis and background theory, which can be achieved by including at least one clause from $NewCarc(B, \overline{E}, \mathcal{P})$ in a bridge theory. In summary, CF-induction uses the following bridge theories:

Definition 3.4 (Bridge theory wrt B, E and $\mathcal{P}$)**.** Let B, E and $\mathcal{P}$ be a background theory, observations and a production field. Let CC be a ground clausal theory. CC is a *bridge theory wrt* B*,* E *and* $\mathcal{P}$ iff CC satisfies the following conditions:

1. Each clause $C_i \in CC$ is an instance of a clause in $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$;
2. At least one $C_i \in CC$ is an instance of a clause in $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$.

If no confusion arises, we alternatively call a bridge theory wrt B, E and $\mathcal{P}$ as a bridge theory of CF-induction.

Theorem 3.1. [26] Let B and E be clausal theories, and $\mathcal{P}$ a production field. Then, for any hypothesis H wrt B, E and $\mathcal{P}$, there exists a bridge formula CC wrt B, E and $\mathcal{P}$ such that $H \models \neg CC$.

This theorem shows that any hypothesis can be computed by constructing and generalizing the negation $\neg CC$ of a set of characteristic clauses CC. After selecting a bridge formula CC, CF-induction computes a clausal theory F obtained by translating $\neg CC$ to CNF. Finally, H is obtained by applying a series of so called *generalizers* to F under the constraint that $B \wedge H$ is consistent. In the current version of CF-induction, there are many such generalizers as follows:

- *Reverse skolemization* [9]: It is used to convert Skolem constants to existentially quantified variables. For instance, when we apply this generalizer to the clausal theory $F = \{\{p(sk_1), q(X, sk_1)\}, \{r(f(sk_2))\}\}$, we can obtain the hypothesis $\{\{p(Y), q(X)\}, \{r(f(Z))\}\}$. Note that this operator is necessary in case that bridge theories contain Skolem constants. Such a case arises if observations E have variables. Because those variables are replaced with Skolem constants to compute $\overline{E_{sk}}$.

- *Anti-instantiation*: It is used to replace ground subterms with variables. For instance, when we apply this generalizer to the clausal theory $F = \{\{p(a, b), q(b, f(f(c)))\}\}$, a hypothesis $\{\{p(a, X), q(X, f(Y))\}\}$ can be obtained by replacing a, b and c with X, Y and Z, respectively. We can consider other applications to obtain alternative hypotheses like $\{\{p(X, Y), q(Y, f(f(Z)))\}\}$ and $\{\{p(X, b), q(b, Y)\}\}$. In this example, we have 2^6 choice points to possibly apply as there are 6 subterms in F. Thus, along with increasing the total number of subterms in F, the possible choice points exponentially increase. In this sense, this generalizer is non-deterministic.

- *Dropping*: It is used to remove any literals from F. For instance, when we apply dropping to the clausal theory $F = \{a \vee b,\ c \vee d\}$, we can obtain a hypothesis $\{a, c \vee d\}$. In general, there are $2^n - 1$ ways to possibly apply dropping for n literals included in F. Then, this generalizer is also non-deterministic.

- *Anti-weakening*: It is used to add some clauses. Let F be a clausal theory and C be an arbitrary clause. Then, it holds in general that $F \cup \{C\} \models F$. Hence, anti-weakening can be soundly applied to F as a generalizer. In case that there is no restriction to added clauses, the number of possible choice points should be infinite. Hence, this generalizer is highly non-deterministic.

- *Inverse resolution* [47]: It applies the inverse of the resolution principle. For instance, when we apply this generalizer to the clausal theory $F = \{\{p(X)\}\}$, we can obtain a hypothesis $\{\{p(X), \neg q(X)\}, \{p(Y), q(Y)\}\}$.

Note that the resolvent of two clauses in the hypothesis corresponds to the clause in F. We may consider a hypothesis $\{\{p(X), r(X)\}, \{\neg r(Y)\}\}$ as another application way. Like $q(X)$ and $r(X)$, there are many (possibly infinite) literals resolved upon. Hence, this generalizer is also highly non-deterministic.

- *Plotkin's least general generalization (LGG)* [55]: This uses the lattice of clauses over the subsumption order. By the property of lattices, for every two clauses C_1 and C_2, there is the least general generalization of C_1 and C_2. Plotkin [55] proposed an algorithm to compute this least generalization. CF-induction allows this algorithm to be used as a generalizer. For instance, when we apply this generalier to $F = \{\{p(a,b), q(a,c)\}, \{p(f(c),b), q(f(c),c)\}\}$, we obtain their least generalization $\{\{p(X,Y), q(X,Z)\}\}$. We may notice that this hypothesis can be alternatively obtained by several times applications of anti-instantiations.

On the one hand, 5 generalizers: reverse skolemization, dropping, anti-instantiation, anti-weakening and LGG are based on the inverse relation of subsumption. In other words, given a clausal theory F, if a theory G is obtained by applying those 5 generalizers to F, it holds that $G \succeq F$. On the other hand, inverse resolution is based on the inverse relation of entailment beyond subsumption. As explained in the previous section, generalization in the other IE-based methods expect for CF-induction is based on the subsumption relation. This fact results in incompleteness of hypothesis finding in those methods, whereas CF-induction needs to treat inverse resolution that is highly non-determinisitc. Using those generalizers, CF-induction computes hypotheses in the following procedure:

Input: A background theory B, observations E
and a production field $\mathcal{P}$.
Output: A hypothesis H wrt B, E and $\mathcal{P}$.

Step 1. Compute $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$.

Step 2. Construct a bridge formula CC wrt B, E and $\mathcal{P}$.

Step 3. Convert $\neg CC$ into a CNF formula F.

Step 4. H is obtained by applying generalizers to F
under the constraint that $B \wedge H$ is consistent.

Several remarks are necessary as follows:

Step 1. CF-induction first computes the complement $\overline{E_{sk}}$ and then uses SOL resolution to obtain $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$.

Step 2. We select one by hand from the possible bridge theories. In general, as the number of clauses in $Carc(B \wedge \overline{E_{sk}})$ increases, the choice points can exponentially increase. In this sense, this procedure is highly non-deterministic.

Step 3. Though the current version does not fix the translation procedure, the (residue or minimal) complement would be available. Thus, in the case of minimal complement, we intend to use the efficient algorithm for the minimal hitting set enumeration described in Section 2.4.

Step 4. If F includes Skolem constants, we apply reverse skolemization to replace all those Skolem constants with variables. We may need to apply a combination of generalizers to obtain target hypotheses. Besides, generalizers do not ensure the consistency of their outputs and the background theory. Then, it becomes necessity to perform the consistency checking in their applications.

Theorem 3.2. [26] Let B, E and $\mathcal{P}$ be a background theory, observations and a production field. Then, the soundness and completeness of CF-induction are described as follows:

1. Soundness: If a clausal theory H is constructed by the above procedure with B, E and $\mathcal{P}$, then H is a hypothesis wrt B, E and $\mathcal{P}$.

2. Completeness: For every hypothesis H wrt B, E and $\mathcal{P}$, H is constructed by the above procedure with B, E and $\mathcal{P}$.

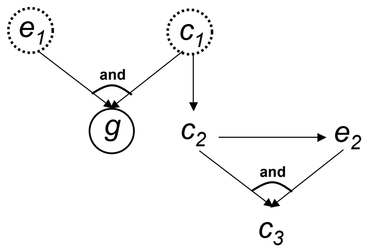
In the following, we give three examples to explain how hypotheses can be derived by CF-induction. The first example deals with the graph completion task in propositional logic.

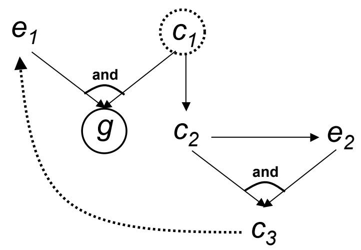
Example 3.1. [Graph completion by CF-induction] Let a background theory B and observations E be as follows:

$$B = \{e_1 \wedge c_1 \supset g,\ c_1 \supset c_2,\ c_2 \supset e_2,\ c_2 \wedge e_2 \supset c_3\},$$
$$E = \{g\}.$$

Suppose a clausal theory $H_1 = \{\{c_1\},\ \{e_1\}\}$. Then, since H_1 logically explain E with B and $B \wedge H_1$ is consistent, H_1 is a hypothesis wrt B and E.

Next, suppose another clausal theory $H_2 = \{\{c_1\},\ \{e_1,\ \neg c_3\}\}$. Though H_2 is more specialized than H_1, it is also a hypothesis wrt B and E since $B \wedge H_2 \models E$ and $B \wedge H_2$ is consistent. The following figures describe the logical relations in the background theory, observations and hypotheses using arrows: In the figures, the observation g is surrounded by a solid circuit and





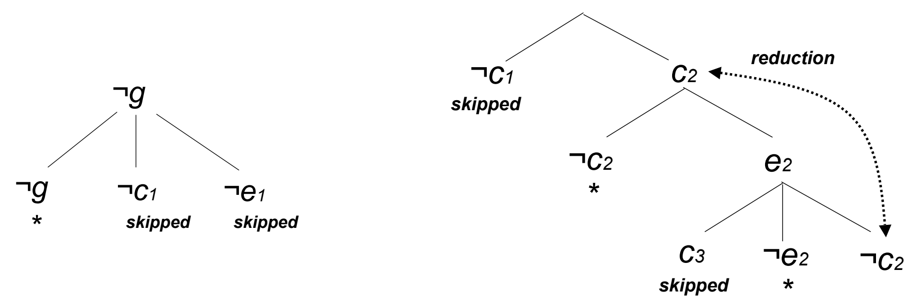
each hypothesis is represented using dotted circuits and lines. H_1 provides two explanations e_1 and c_1 that account for the reason why g occurs (See the left figure). On the other hand, H_2 contains not only one explanation c_1 but also a missing causal relation between c_3 and e_1 (See the right figure). In this sense, H_2 completes unknown causal relation in the prior background theory. For deriving H_1 and H_2 by CF-induction, we set the following production field:

$$\mathcal{P} = \langle\{\neg c_1,\ \neg e_1,\ c_3\}\rangle.$$

For every literal in H_1 and H_2, the set $\{\neg c_1,\ \neg e_1,\ c_3\}$ of $\mathcal{P}$ contains its negation. Then, both H_1 and H_2 are hypotheses wrt B, E and $\mathcal{P}$. Thus, they should be derived by CF-induction. CF-induction first computes the characteristic clauses of $B \wedge \overline{E_{sk}}$ by SOL resolution. Since E is ground, $\overline{E_{sk}}$ is equal to $\overline{E} = \{\{\neg g\}\}$. $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$ and $Carc(B, \mathcal{P})$ are as follows:

$$\begin{aligned} NewCarc(B, \overline{E_{sk}}, \mathcal{P}) &= \{\{\neg c_1,\ \neg e_1\}\}, \\ Carc(B, \mathcal{P}) &= \{\{\neg c_1,\ c_3\}\}. \end{aligned}$$

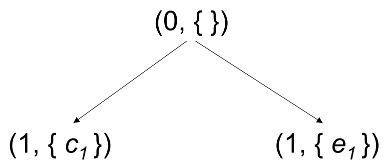
Note that both sets do not include any tautological clauses since $\mathcal{P}$ does not include complementary literals. SOL resolution constructs tableaus for deriving two clauses $\{\neg c_1,\ \neg e_1\}$ and $\{\neg c_1,\ c_3\}$, as shown in the following right and left figures, respectively: By the definition, there are two possible

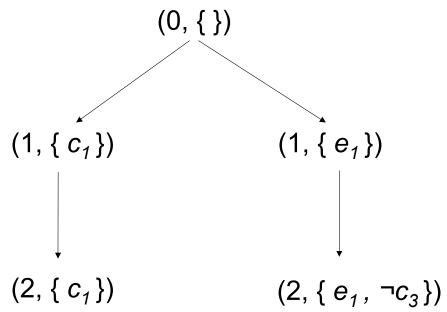


bridge theories CC_1 and CC_2 wrt B, E and $\mathcal{P}$ as follows:

$$\begin{aligned} CC_1 &= \{\{\neg c_1,\ \neg e_1\}\}, \\ CC_2 &= \{\{\neg c_1,\ \neg e_1\},\ \{\neg c_1,\ c_3\}\}. \end{aligned}$$

We next compute two CNF formulas F_1 and F_2 obtained by translating $\neg CC_1$ and $\neg CC_2$. We represent F_1 and F_2 as the minimal complements $M(CC_1)$ and $M(CC_2)$, respectively. The minimal complements can be efficiently computed by the algorithm for minimal hitting set enumeration, explained in Section 2.4: The algorithm constructs two enumeration trees for computing $M(CC_1)$ and $M(CC_2)$, as shown in the following right and left figures, respectively. As a result, we obtain $M(CC_1) = \{\{c_1\}, \{e_1\}\}$





and $M(CC_2) = \{\{c_1\}, \{e_1, \neg c_3\}\}$ that correspond to the target hypotheses H_1 and H_2, respectively. Note that we do not use generalizers, and thus do not need to check the consistency of hypotheses with the background theory.

Whereas H_1 can be obtained by the other IE-based methods in Figure 3.1, H_2 is derived by only residue procedure and CF-induction. As explained in Section 3.1, for one example, Progol searches a single clause as a hypothesis. Since H_2 contains multiple clauses, Progol cannot generate it. In turn, since the body literal c_3 in H_2 cannot derived only by B, HAIL cannot generate H_2. The body literal c_3 becomes true if both c_2 and e_2 should be true, together with B. However, c_2 and e_2 are not derived only by B. Hence, Imparo cannot generates H_2. On the other hand, the residue procedure can do in such a way that we select the bridge theory F consisting of all the clauses in $B \wedge \overline{E_{sk}}$, compute its residue complement $R(F)$ and generalize it based on the inverse relation of subsumption.

We next give an example that deals with the theory completion task on integer number. Unlike the previous example, this example is requires several generalizers to obtain target hypotheses including a non-propositional case.

Example 3.2 (Theory completion by CF-induction)**.** Let a background theory B, an example E be as follows:

$$\begin{array}{rcl} B & = & natural(0) \vee even(0), \\ E & = & natural(s(0)). \end{array}$$

The successor function $s(X)$ denotes the next number of X, that is, $X + 1$. Formally, it is defined as $s^{n+1}(X) = s(s^n(X))$ for $0 \leq n$, and $s^0(X) = X$. In this example, we do not have any general rules on integer number, but know only the fact that 0 is a natural or even number. Given the new fact that $s(0)$ (i.e. 1) is a natural number, we may infer some rules to explain this. Here, we show how CF-induction generates such missing rules. Assume a production field $\mathcal{P}$ as follows:

$$\mathcal{P} = \langle \{natural(X), \neg natural(X), even(X), \neg even(X)\} \rangle.$$

Then, $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$ and $Carc(B, \mathcal{P})$ are as follows:

$$\begin{array}{rcl} NewCarc(B, \overline{E_{sk}}, \mathcal{P}) & = & \{\neg natural(s(0))\}, \\ Carc(B, \mathcal{P}) & = & \{\{natural(0),\ even(0)\}\} \cup Taut, \end{array}$$

where $Taut$ denotes two tautological clauses $\{natural(X), \neg natural(X)\}$ and $\{even(X), \neg even(X)\}$, which are constructed from $\mathcal{P}$. We may notice that $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$ and $Carc(B, \mathcal{P})$ correspond to $\overline{E_{sk}}$ and B, expect for tautological clauses. That is because there is no resolvent from B and $\overline{E_{sk}}$.

Let CC be a clausal theory $(natural(0) \vee even(0)) \wedge \neg natural(s(0))$. Since each clause in CC is a clause in $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$ and the unit clause $\neg natural(s(0))$ in CC is a clause in $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$, CC is a bridge formula wrt B, E and $\mathcal{P}$. The minimal complement of CC is as follows:

$$M(CC) = (natural(0) \supset natural(s(0))) \wedge (even(0) \supset natural(s(0))).$$

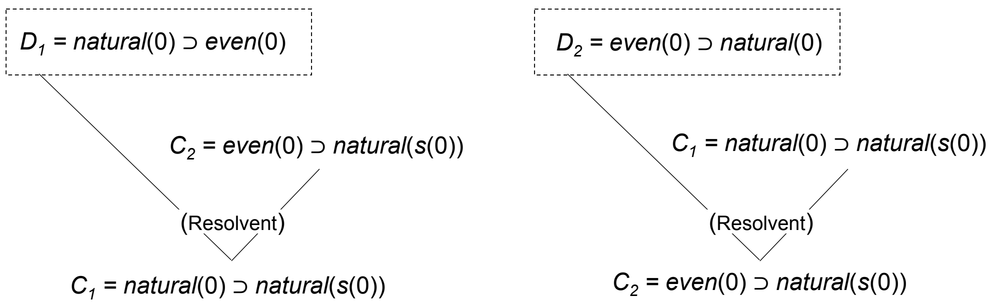
Since $B \wedge M(CC)$ is consistent, $M(CC)$ is a hypothesis wrt B, E and $\mathcal{P}$. Assume that an inverse resolution generaliser is applied to $M(CC)$ in such a way that the clause $C_1 = natural(0) \supset natural(s(0))$ in $M(CC)$ is replaced with the clause $D_1 = natural(0) \supset even(0)$, which is treated as a parent clause of

C_1. This means C_1 is the resolvent of D_1 and $C_2 = even(0) \supset natural(s(0))$ in $M(CC)$. Then the following clausal theory H_1 is constructed:

$$H_1 = (natural(0) \supset even(0)) \land (even(0) \supset natural(s(0))).$$

Since $B \land H_1$ is consistent, H_1 is a hypothesis wrt B, E and $\mathcal{P}$.

The following left (*resp.* right) figures represent the logical relation between D_1 (*resp.* D_2), C_1 and C_2. Both D_1 and D_2 are obtained by applying inverse resolution to C_1 and C_2. We may notice that the difference arises in which clause is assumed to the resolvent.



Next, assume that another inverse resolution generaliser is applied to $M(CC)$ in such a way that the clause C_2 in $M(CC)$ is replaced with the clause $D_2 = even(0) \supset natural(0)$. Then the following clausal theory H_2 is constructed.

$$H_2 = (natural(0) \supset natural(s(0))) \land (even(0) \supset natural(0))$$

Since $B \land H_2$ is consistent, H_2 is also a hypothesis wrt B, E and $\mathcal{P}$. In addition to the above generaliser, if we apply an anti-instantiation generaliser to H_2 in such a way that the ground term 0 occurring in H_2 is replaced with the variable X, then the following theory is obtained:

$$H_3 = (natural(X) \supset natural(s(X))) \land (even(X) \supset natural(X)).$$

Since $B \land H_3$ is consistent, H_3 is also a hypothesis wrt B, E and $\mathcal{P}$.

Every hypothesis H_1, H_2, and H_3 cannot be derived by the other IE-based methods in Figure 3.1. That is because those hypotheses are obtained by applying inverse resolution generalizers, which involve the entailment relation beyond subsumption. As explained before, every IE-based methods except for CF-induction uses the inverse relation of subsumption, instead of entailment, in generalization. This example thus shows that the reduction from entailment to subsumtion may cause the incompleteness in generalization.

Both examples above involve completion problems to find some missing fact or causal relations in the prior background theory. On the one hand, every hypothesis in the first example is a ground explanation, which can be obtained without gereralizers. On the other hand, the last hypothesis in the second example can be regarded as a general rule, rather that an explanation, lacked in the background example. Hence we may view the first and second example are achieved by abduction and induction, respectively. In next example, we show that CF-induction can achieve an advanced inference integrating both abduction and induction.

Example 3.3. [Integrating abduction and induction by CF-induction]

Assume the following background theory B, observations E, and target hypothesis H wrt B and E:

$$\begin{aligned}
B &= \{arc(a,b),\ arc(X,Y) \wedge path(Y,Z) \supset path(X,Z)\}.\\
E &= \{path(a,c)\}.\\
H &= \{arc(b,c),\ arc(X,Y) \supset path(X,Y)\}.
\end{aligned}$$

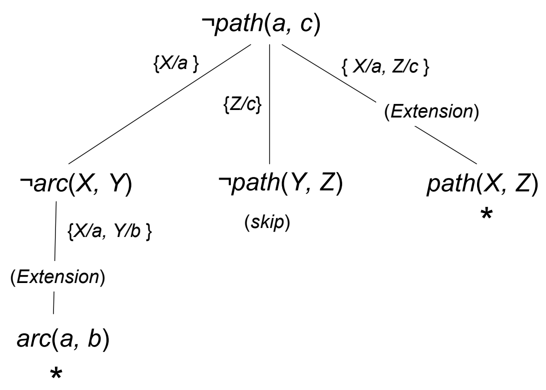
One arc from b to c and one general rule on pathways are missing in B. The task is to find the hypothesis H that completes these missing fact and rule. To complete H, both abduction and induction must involve, but most current ILP systems cannot compute it. This advanced inference has a possibility to be effectively applied to systems biology [88], later.

Let a production field $\mathcal{P}$ be $\langle\{arc(X,Y), \neg arc(X,Y), \neg path(X,Y)\}\rangle$. Using SOL-resolution, the following new characteristic clause C_1 of $\overline{E_{sk}}$ wrt B

and $\mathcal{P}$ and characteristic clause C_2 of B wrt $\mathcal{P}$ are obtained:

$$C_1 = \{\neg path(b,c)\} \in NewCarc(B, \overline{E_{sk}}, \mathcal{P}).$$
$$C_2 = \{arc(X,Y) \vee \neg arc(X,Y)\} \in Carc(B, \mathcal{P}).$$

The following figure is a tableau of SOL resolution for deriving C_1:



Note that C_2 is a tautological clause constructed from $\mathcal{P}$. We then select the following bridge theory CC wrt B and $\mathcal{P}$ consisting C_1 and a ground instance of C_2:

$$CC = \{\neg path(b,c),\ arc(b,c) \vee \neg arc(b,c)\}.$$

The minimal complement of CC is as follows:

$$M(CC) = \{path(b,c) \vee \neg arc(b,c),\ path(b,c) \vee arc(b,c)\}.$$

The target hypothesis H is then derived from $M(CC)$ in such a way that the literal $path(b,c)$ in the clause $path(b,c) \vee arc(b,c)$ is removed by applying a dropping generalizer and then two constants b and c in the clause $path(b,c) \vee \neg arc(b,c)$ are replaced with two variables X and Y, respectively, by applying an anti-instantiation generalizer.

Like Example 3.1 and 3.2, the hypothesis H cannot be obtained by the other IE-based methods in Figure 3.1. B includes one pathway definition that needs non-Horn representation formalisms. Then, it is not straightforward to use Progol, HAIL and Imparo in this example. Though FC-HAIL and residue procedure can be applied, both cannot generate H. In the case of FC-HAIL, it should be at least necessary to derive the body literals $arc(b, c)$ and $arc(b, c)$ in $M(CC)$ from B. However, they cannot be derived only from B. Accordingly, FC-HAIL cannot generate H. In turn, the residue procedure does not allow bridge theories to include any tautological clauses. Thus, the residue procedure cannot also generate H.

As we have described so far, CF-induction is the unique IE-based method that preserves the soundness and completeness for finding hypotheses with respect to a given language bias (i.e. production field) in full clausal theory. Like Example 3.3, this theoretical advantage enables us to obtain complex hypotheses by integrating abduction and induction. In next chapter, we introduce a practical problem in life science that really requires such advanced inference. Lastly, we remark two important issues on implementation and production fields in CF-induction, using the rest of this chapter.

Implementation: The previous version of CF-induction, which we have reviewed here, has been implemented by JAVA. In this implementation, we use SOLAR and the efficient algorithm for minimal hitting set enumeration, shown in Section 2.3 and 2.4, in order to compute the characteristic clauses and translate the negation of a bridge theory into CNF, respectively. Then, Step 1 and Step 3 are automatically performed. On the other hand, in Step 2, users need to select which instances should be included in a bridge theory by hand. Along with construction of bridge theories, Step 3 also need to make users select which generalizers should be used as well as how to apply the selected generalizers. Currently, three generalizers: dropping, anti-instantiation and LGG are embedded in the implementation. Thus, the current system is still far from automatically finding relevant hypotheses. The essential problem lies in that both procedures: construction of bridge theories and generalization into hypotheses are highly non-deterministic. In Chapter 5 and 6, we will deal with this problem to reduce the non-determinism.

Production field: The current version of CF-induction does not allow the production fields to include any conditions like the maximum length of clauses. That is because a production field cannot directly define the syntax of hypotheses. Instead, it provides a condition to be satisfied in the negations of hypotheses. However, it is difficult to translate the negation of a hypothesis back into the original hypothesis.

Example 3.4. Consider a production field $\mathcal{P} = \langle \mathcal{L}, max\_length \leq 2 \rangle$ including the condition of the maximum length. Let a clausal theory H, representing a target hypothesis, be $\{\{a, \neg b\}, \{a, b, c\}, \{c\}\}$. Assume a clausal theory $S = \{\{\neg a,\ \neg c\},\ \{b,\ \neg c\}\}$ that represents $\neg H$ in clausal form and belongs to $\mathcal{P}$. Since $S \equiv \neg H$, $\neg S$ is logically equivalent to H. Hence, it would be possible to obtain the original hypothesis from $\neg S$ with equivalent translating operations. We have already investigated several such translating ways in Section 2.4: the residue and minimal complements, shown as follows:

$$\overline{S} = \{\{a,\ \neg b\},\ \{a,\ c\},\ \{c,\ \neg b\},\ \{c\}\},$$
$$R(S) = \overline{S},\ M(S) = \{\{a,\ \neg b\},\ \{c\}\}.$$

We notice that every equivalent theories $\overline{S}$, $R(S)$ and $M(S)$ is not equal to the original hypothesis H.

As Example 3.4 shows, there are several ways to translate the negation of a hypothesis into clausal theories that are logically equivalent to the original hypothesis. However, these translated theories cannot be syntactically equal to the original form. Thus, even if the syntax of the negations could be efficiently restricted by a production field with conditions like the maximum length, it was not straightforward that this restriction to negations can also affect to the syntax of hypotheses itself. We intend to investigate the issue how we sufficiently define the syntax of hypotheses using production fields, again in Chapter 6.

Chapter 4

Theory Completion using CF-induction: Case Study

In this chapter, we provide a new practical application of inference-based hypothesis finding techniques. Its main task involves theory completion in biological networks. Unlike the previous applications, it requires both abuctive and inductive inference to find missing facts and *general* causal relations in incomplete knowledge on life systems, which can be achieved by CF-induction as we explained in Chapter 3. We first give the problem setting so as to use ILP techniques and then show initial experimental results using CF-induction. Finally, we overview the related work and conclude.

4.1 Introduction

The newly emerging field of *Systems Biology* has been developed toward precise understanding of the whole mechanism of living cells and organisms. *Metabolism* is one of the essential biological systems oriented in this field. It is organized in a complex network of interconnected reactions, called a *metabolic pathway* [11], and its whole behavior results from individual properties of reactions and global properties of the network organization. An important key for understanding this whole metabolic system lies in regulatory mechanism on activities of enzymes catalyzing chemical reactions involved in metabolism. *Metabolic Flux Analysis* (MFA) [74, 70] is a methodology for quantitatively analyzing those enzymatic activities. The *flux* of a reaction, defined by the rate of the reaction, can be regarded as an effective value for indicating the activity of a certain enzyme catalyzing the reaction. There are two kinds of approaches for computing a flux distribution over the reactions. The first approach uses kinetics of chemical reactions, and models the time-series changes of fluxes [40]. These dynamic behaviors can be represented as coupled non-linear differential equations. The second approach introduces a steady-state approximation to the first approach and reconstructs a prior set of equations into the linear formalization by considering the stoichiometry of the chemical reactions [81]. In general, the equations in both approaches cannot be analytically solved. Because, there are a large number of intracellular metabolites involved in the chemical reactions. The fluxes of these reactions cannot be experimentally observed. Thus, the non-linear equations are under-determined. This problem is usually managed in such a way that the equations are numerically simulated *in silico* with several kinds of approximating constraints. Indeed, *elementary mode analysis* [71] and *extreme pathways analysis* [69] have been previously proposed in the second approach, and introduce some relevant optimization functions with respect to the cellular growth maximization or the energy consumption minimization. However, these assumptions can often cause the dissociation with real situation. Moreover, even if these approximation methods can be utilized, a large-scale metabolic pathway cannot be solved only with these methods due to huge computational costs.

Our long-term goal is to identify *master reactions* whose fluxes are relatively high in a metabolic pathway [12]. It is a crucial feature of flux distributions that reactions with fluxes spanning several orders of magnitude coexist under the same conditions [1]. Whereas most metabolic reactions have low fluxes, the overall behavior of metabolism is dominated by several reactions with very high fluxes. Therefore, we can divide activities of enzyme reactions into two kinds of states, that is, an activated state and a non-activated state. If we could know which chemical reactions are in an activated or a non-activated state, it would be helpful to solve the equations using the previously proposed MFA techniques. Because we can reconstruct a prior set of equations into more simplified ones as the non-activated reactions with low fluxes can be ignored.

In this work, we focus on a logic-based approach that enables us to estimate possible reaction states in a metabolic pathway. Our approach introduces the logical viewpoint with respect to causal relations between states of the enzymatic activity influencing a reaction and concentration changes of metabolites involved in the reaction. Based on these causal relations, we quantitatively estimate possible states of enzyme reactions, which logically explain the concentration changes of measurable metabolites obtained from experiments. Computation for this estimation is based on Inductive Logic Programming (ILP) [36]. ILP studies inductive learning with relational representation in first-order predicate logic. As we explained before, the main task of ILP is to find hypotheses that logically explain a set of observations with respect to a background theory. In this ILP setting, we can obtain possible states of enzyme reactions as a hypothesis that logically explains the concentration changes of the measurable metabolites with a background theory. Though there are several ILP systems, we focus on CF-induction.

CF-induction has a unique feature that can integrate inductive and abductive inferences, preserving its soundness and completeness for finding hypotheses in full clausal logic. While both inductive and abductive inferences are used to find hypotheses that account for given observations, their use in applications is quite different. Abduction is applied for finding specific explanations (causes) of observations obtained by using the current background theory. On the other hand, induction is applied for finding general rules that hold universally in the domain, but are missing in the background theory. In our problem, an explanation obtained by abduction corresponds to an estimation of enzyme reaction states. If a background theory is complete with respect to the regulatory mechanism of enzymatic activities, then possible reaction states could be computed only using abduction. However, since background theories are incomplete in general, it is necessary to find such missing rules that represent some unknown control mechanisms using induction. Therefore, it could be a crucial advantage if we could analyze metabolic pathways using both abductive and inductive inferences in CF-induction. We show how CF-induction can work for both estimating possible reaction states and completing missing causal relations using several examples.

The rest of this chapter is organized as follows. Section 2 first explains notions of metabolic pathways and a basic approach for MFA in brief, and next introduces the logical model representing the causal relations between enzymatic activities and concentration changes of metabolites. Section 3 shows experimental results obtained by applications of CF-induction for estimation of possible reaction states from given observations. The examples include the metabolic network of Pyruvate as well as simple topology of a metabolic pathway. Section 4 discusses related work. Section 5 concludes.

4.2 Logical Modeling of Metabolic Flux Dynamics

4.2.1 Metabolic Pathways

Whereas cells have different morphologies and structures and the fact that their roles in the different organisms are varied, their basic functionality is the same. One of those basic activities of cells is to insure their own survival. Its whole activity can be summarized in the two points. First, cells need to find the necessary energy for its activity. This energy is mainly obtained by degradation of mineral or organic molecules. Second, cells need to manufacture simple molecules necessary to their surviving. The former is called *catabolism* and the latter is called *anabolism*. These two great activities are regrouped under the name of metabolism, and result from a great number of mechanisms and biochemical reactions. Most of these reactions, unfolding in a cell, are catalyzed by special molecules, called *enzymes*. Such a large amount of data on metabolism is represented as a network [11], called a metabolic pathway, and has been stored and maintained in a large-scale database, such as KEGG [30].

Recently, the study of metabolic pathways is becoming increasingly important to exploit an integrated, systemic approach for simulating or optimizing cellular properties or phenotypes. One of these significant properties is a metabolic flux defined as the rate of a biochemical reaction, which can be very often utilized to improve production of metabolites in industry [74]. Figure 4.1[1] describes non-linear phenomena in the growth of the yeast *S. cerevisiae*, which has been extensively studied. The microorganism senses the availability of favorite sugars like glucose, and under nutrient-rich environment, it rapidly grows and yields products like ethanol by consuming its favorite sugars (glucose). This state is called, *permentation.* On the other hand, if the nutrient has been finished consuming, it gradually grows using its product (ethanol), instead glucose. This state is called, *oxidation.* It is known that these phenomena arise in dynamic transitions of *master reactions* in metabolism of the yeast, shown in Figure 4.2. Master reactions denote enzyme reactions with high flues in the metabolic pathway. In this sense, the study on analyzing flux distributions has significant impacts on simulation and optimization of (metabolic) bio-based products such as ethanol.

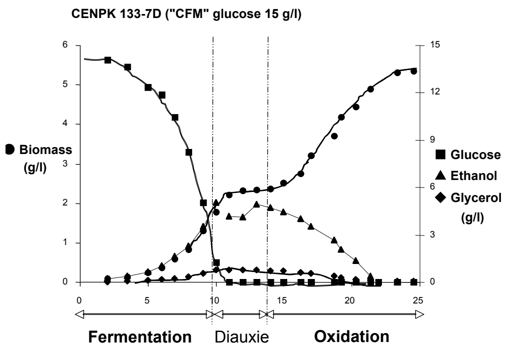


Figure 4.1: Dynamics Emerged in The Growth of Yeast

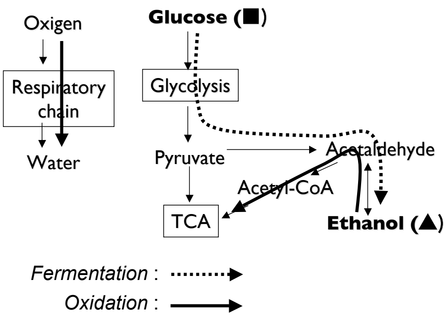


Figure 4.2: Transition of Master Reactions

One basic but powerful approach to understand the steady state fluxes is *metabolite flux balancing*, which is based on the stoichiometric model of the biochemical reactions. Figure 4.3 represents simple topology of a metabolic pathway in a cell, which consists of five metabolites A, B, C, D and E and six reactions, each of which connects two certain metabolites. Each flux is placed on the corresponding reaction in Figure 4.3. Although the concentrations of A, C, D and E are experimentally measurable, the concentration of B cannot

[1]This figure is provided by Prof. G. Goma of INRA CNRS in France.

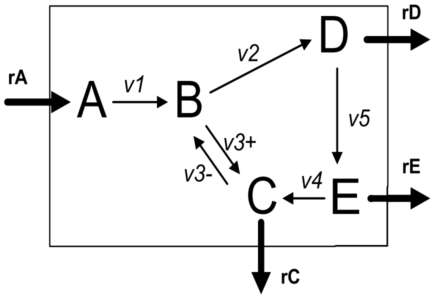


Figure 4.3: System of Mass Balance

be measured. Hence, B is the intracellular metabolite. Based on the enzyme kinetics, the dynamic behavior of the flux of an enzyme reaction can be represented as the following differential equation:

$$\frac{dC_X}{dt} = v_{in} - v_{out} - \mu C_X, \tag{4.1}$$

where C_X is the concentration of a metabolite X, v_{in} (resp. v_{out}) is the sum of fluxes of reactions for producing (resp. consuming) X, and μC_X represents the growth rate of biomass in a cell. If all the metabolites are in the steady state, the left term of Equation (4.1) must be zero since there are no time-series changes of the concentrations, and also, it can be assumed that the dilution of components due to biomass growth (corresponding to the last term of Equation (4.1)) is neglected [70]. This fact means that for each metabolite X, the fluxes consuming X are balanced with the ones producing X in the steady state. Metabolic flux balancing is based on this simple notion. For example, its balancing in Figure 4.3 can be represented as the following linear equations:

$$\begin{gathered} v_1 = rA,\ rD + v_5 = v_2,\ rE + v_4 = v_5, \\ v_2 + v_{3+} = v_{3-} + v_1,\ rC + v_{3-} = v_{3+} + v_4. \end{gathered} \tag{4.2}$$

Then we can analyze the flux distribution based on Equation (4.2) with the measurable fluxes rA, rC, rD and rE. In general, these equations cannot be deterministically solved as the number of unknown values such as $v_1, \ldots, v_5$ corresponding to the fluxes of intracellular enzyme reactions becomes larger than the number of known values corresponding to measurable fluxes. The previously proposed methods such as elementary mode analysis and extreme pathway analysis use optimization functions in order to solve the equations. Those introduced functions are usually constructed by assuming the cellular growth maximization or the energy consumption minimization. However, these assumptions can often cause the dissociation with real situation. In other words, we cannot necessarily determine if the organism indeed grows based on the above assumptions to be expected. Additionally, in the case of a large-scale of metabolic pathway, it is not straightforward to solve the flux distribution with approximation methods due to huge computational cost.

In this work, we propose a new approach that enables us to release those unsure assumptions and reduce the complexity of a given metabolic pathway. One essential feature of enzymatic activities is that all the activities are not necessarily on the same level. There exist enzymes whose activities are about 100 or 1000 times higher than other enzymes. This fact allows us to assume whether each enzyme reaction is in a *relatively* activated state or not. Then, if we could estimate which enzyme reactions are in an activated or a non-activated state, we could simplify the prior metabolic pathway by ignoring those reactions in the non-activated state, which are estimated to have low fluxes. The smaller the target pathway is, the smaller the number of unknown values in the equations obtained from the pathway is. It may imply that the possibility of solving the equations based on the metabolic flux balancing without the previously proposed approximation methods. In our approach, we introduce a logical model that represents causal relations between enzyme reaction states and concentration changes of metabolites. Based on the logical model, we can logically find possible states that can explain the observations which are experimentally observed. In the following, we focus on those causal relations in enzyme reactions.

4.2.2 Regulation of Enzymatic Activities

The cellular metabolic system has a sophisticated mechanism for dynamically controlling the activities of enzymes to meet the needs of a cell. This regulatory mechanism can be represented as causal relations between enzymatic activities and concentration changing of metabolites. Here we consider two simple metabolic pathways: First one consists of two reactions with three metabolites, and second one consists of one reaction with two metabolites. Note that in the following figures, we describe activated and non-activated reactions as (back) circles and slashes over arrows corresponding to reactions, respectively. And also, a upward (resp. downward) arrow represents the increase (resp. decrease) in a metabolite concentration.

Figure 4.4 corresponds to the metabolic pathway consisting of three metabolites X, Y and Z, and two reactions. Figure 4.4 shows that if the concen-

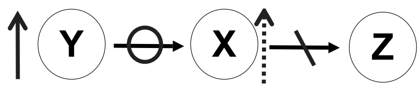


Figure 4.4: First Biological Inference in Metabolic Pathways

tration of Y tends to be increasing at some time, provided that the state of enzyme reaction $Y \rightarrow X$ (resp. $X \rightarrow Z$) is in an activated (resp. non-activated) state, then the concentration of X will also change to be increasing. This causal relation is rational biological inference based on Equation (4.1). Assume that the increase in concentration of X is observed, which is denoted by a dotted arrow in the figures. Then, it will be possible to logically estimate the states of the concentration change of Y and two reactions so that the estimated states cause this concentration change of X. One possible case is that the concentration of Y increases, the reaction $Y \rightarrow X$ is activated and the reaction $X \rightarrow Z$ is not activated. This is because X produced from Y cannot be consumed and generate Z.

Next, we consider Figure 4.5 which represents a metabolic pathway consisting of two metabolites X and Y, and one reaction. Figure 4.5 shows that

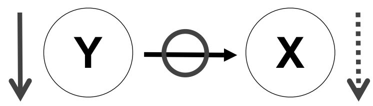


Figure 4.5: Second Biological Inference in Metabolic Pathways

even if the reaction $Y \rightarrow X$ is activated, the concentration of X must decrease as far as the concentration of Y decreases. Accordingly, if we observe the concentration of X decreases, we can logically estimate the concentration of Y decreases and the reaction $Y \rightarrow X$ is activated as one possible case.

As we see in the above, consideration of these causal relations enables us to estimate possible reaction states that explain the concentration changes of measurable metabolites. Two causal relations shown in Figures 4.4 and 4.5 are not sufficient for explaining all the possible cases. In other words, there exist cases that we cannot estimate possible reaction states using these causal relations only. Although it would be possible to construct other causal relations a priori, however it must be crucially difficult to *enumerate* all the causal relations corresponding to the whole regulatory mechanism on enzymatic activities. This problem brings the necessity to complete the current causal relations for estimation in some cases. Hence, we need to simultaneously realize these two tasks, that is, estimation of possible reaction states and completion of missing causal relations.

We represent our problem in the ILP setting. The metabolic pathway topology in Figure 4.3 can be represented as the following clausal theory T consisting of facts:

$$T = \{reac(a,b), reac(b,d), reac(d,e), reac(e,c), reac(b,c), reac(c,b)\},$$

where the literal $reac(X,Y)$ means that there is a reaction between the substrate X and the product Y. Along with the logical representation of topology, we formalize the causal relations in Figures 4.4 and 4.5 as the following two clauses (4.3) and (4.4), respectively:

$$\begin{aligned} &reac(Y,X) \wedge reac(X,Z) \wedge con(Y,up) \\ &\quad \wedge act(Y,X) \wedge \neg act(X,Z) \supset con(X,up), \end{aligned} \tag{4.3}$$

$$\begin{aligned} &reac(Y,X) \wedge con(Y,down) \\ &\quad \wedge act(Y,X) \supset con(X,down), \end{aligned} \tag{4.4}$$

where the literal $con(X,up)$ (resp. $con(X,down)$) means that the concentration of the metabolite X increases (resp. decreases), and the literal $act(X,Y)$ means that the reaction $X \rightarrow Y$ is activated. Note that both (4.3) and (4.4) are non-Horn clauses.

In the ILP setting of our problem, the background theory B consists of the above logical formulas. Along with B, observations E is given as concentration changes of measurable metabolites obtained from experimental results. Using these two inputs B and E, we need to compute hypotheses that not only estimate possible reaction states, but also complete missing causal relations in B. In this chapter, we use CF-induction for these two tasks estimation and completion, which is the unique ILP technique that can realize both abductive and inference inference, simultaneously.

4.3 Experiments

In this section, we show what kinds of hypotheses the current implementation of CF-induction can find using two examples. The simple pathway in the first example corresponds to Figure 4.3, and the metabolic pathway of Pyruvate is used in the second example.

4.3.1 Simple Pathway

Define a background theory B as follows:

$$\begin{aligned} B = T \cup \{&con(a, up), (4.3), (4.4), \\ &\{\neg con(X, up), \neg con(X, down)\}\}, \end{aligned} \tag{4.5}$$

where the rule (4.5) means that concentrations of any metabolites cannot be up and down at the same time. Note here that $con(a, up)$ can be regarded as an input signal to the metabolic system. So is put into B. In the following figures, concentration changes that are assumed to be observed and then included in E are represented as dotted bold arrows. Thus, we let the observations E denoting the measurable concentration changes as follows:

$$E = \{con(d, up), con(c, down), con(e, down)\}.$$

As a hypothesis with respect to B and E, the following clausal theory is considerable:

$$\begin{aligned} H_1 = \{&con(e, down) \leftarrow \neg act(d, e) \wedge act(e, c), \\ &act(a, b), act(e, c), act(b, d), \neg act(b, c), \neg act(d, e)\}. \end{aligned}$$

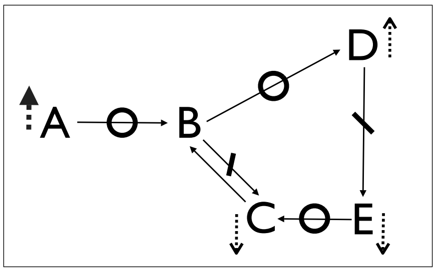


Figure 4.6: First Hypothesis H_1

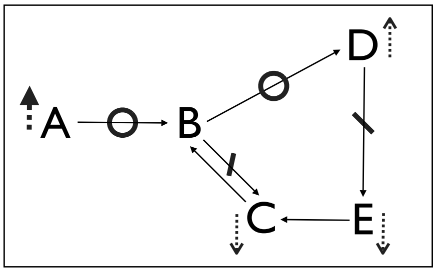


Figure 4.7: Second Hypothesis H_2

Figure 4.6 shows the reaction states in H_1. According to H_1, whereas the reactions $A \rightarrow B$, $B \rightarrow D$ and $E \rightarrow C$ are activated, the reactions $B \rightarrow C$ and $D \rightarrow E$ are not activated. This estimation of reaction states is realized using abduction. On the other hand, we cannot explain the reason why the concentration of E decreases only with abductive hypotheses. For explaining this concentration decreasing of E, H_1 includes a new rule worth considering as a reasonable answer, that is, "the concentration of E decreases if the reaction $D \rightarrow E$ is not activated and the reaction $E \rightarrow C$ is activated".

As another hypothesis, the following clausal theory can be also considered:

$$\begin{aligned} H_2 = \{&con(X, down) \leftarrow \neg act(Y, X) \wedge con(Y, up), \\ &act(a, b), act(b, d), \neg act(b, c), \neg act(d, e)\}. \end{aligned}$$

Figure 4.7 shows the states of reactions in H_2. Compared with H_1, the rule for explaining the concentration change of E is more general, and also H_2 does not say whether the reaction $E - C$ is activated or not. Hence, the hypothesis H_2 logically estimates several reaction states and completes one missing rule on biological inference in metabolic pathways.

These two hypotheses can be indeed computed using the implementation of CF-induction. Let a production field $\mathcal{P}$ be as follows:

$$\mathcal{P} = \langle\{con(X, Y), \neg con(X, Y), \neg act(X, Y), act(X, Y)\}\rangle.$$

Note that, due to computational efficiency, we set the maximum length of characteristic clauses and the maximum search-depth for computing the characteristic clauses as 6 and 4, respectively. Then, CF-induction displays

$NewCarc(B, \overline{E_{sk}}, \mathcal{P})$ and $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$ consisting of 30 and 4 clauses, respectively, as follows.

```
NewCarc:
 (1)[-con(c,down),-con(b,up),-act(b,d),act(d,e),-con(e,down)]}
...
(9)[-con(e,down),-act(e,c),-act(a,b),act(b,c),-act(b,d),act(d,e)]
...
(30)[-con(c,down),-con(d,up),-con(e,down)]

Carc:
(1)[-con(_0,up),-con(_0,down)]
(2)[con(a,up)]
(3)[con(_0,_1),-con(_0,_1)]
(4)[-act(_0,_1),act(_0,_1)]
```

We construct a bridge formula CC_1 in such a way that both two instances of the 4th clause in $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$ and the 9th clause in $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$ are manually selected. The display of CF-induction is as follows:

```
CC:[[-con(e,down),-act(e,c),-act(a,b),act(b,c),-act(b,d),act(d,e)],
    [-act(d,e),act(d,e)],
    [-act(e,c),act(e,c)]]
```

The system automatically computes the minimal complement $M(CC_1)$ of CC_1 as follows.

```
-CC:[[con(e,down),act(d, e),-act(e, c)],
     [act(e,c),act(d,e)],
     [act(a,b),act(d,e),-act(e,c)],
     [-act(b,c),act(d,e),-act(e,c)],
     [act(b,d),act(d,e),-act(e,c)],
     [-act(d,e),act(e,c)],
     [-act(d,e),-act(e,c)]]
```

Next, we apply an dropping generalizer to $M(CC_1)$ in such a way that relevant 9 literals from 18 literals in $M(CC_1)$ are dropped. Then, the clausal theory corresponding to H_1 is constructed. After automatically checking the consistency of H_1 with B, the system successfully outputs H_1.

```
Hypothesis:[[con(e,down),act(d,e),-act(e,c)],
            [act(e,c)],
            [act(a,b)],
            [-act(b,c)],
            [act(b,d)],
            [-act(d,e)]]
```

Test example	($depth = 6$, $max\_length = 6$)		
	$con(c, down)$	$con(d, up)$	$con(e, down)$
N_H	21	22	22
N_P	8	2	7
Ratio [%]	38	9	32
Time [msec]	6915	11360	10751

Table 4.1: Performance of abductive explanations in CF-induction

Next, we set the maximum length of characteristic clauses and the maximum search-depth for computing the characteristic clauses as 6 and 6. Then, $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$ (resp. $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$) increases to 137 (resp. 84) clauses. We construct another bridge formula CC_2 in such a way that one clause in $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$, two clauses and two instances of a clause in $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$ are selected. After computing the minimal complement $M(CC_2)$ of CC_2 , we apply both a dropping and an anti-instantiation generalizers to $M(CC_2)$ in such a way that relevant 15 literals are dropped from 27 literals in $M(CC_2)$ and 4 constants are replaced by variables. Then, the clausal theory corresponding to H_2 is constructed.

Both H_1 and H_2 need inductive inference to be constructed. It becomes necessary to apply inductive inference for completing missing general rules. Another role of inductive inference is to find such hypotheses that have high predictive accuracy to unknown observations, which are generated by applying generalizers like anti-instantiation. Indeed, if abductive inference is only used, the predicate accuracy tend to become low, as we describe in the below.

Table 4.1 shows the predicate accuracy of abductive hypotheses each of which is a set of ground unit clauses, such as $\{act(a, b),\ \neg act(b, c)\}$, obtained by CF-induction. Since the negation of every abductive hypothesis is the form of ground clause, every subsumption-minimal abductive hypothesis can be derived by CF-induction in such a way that a bridge theory CC only contains one ground instance of a clause in $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$, and outputs the minimal complement $M(CC)$ without applying any generalizers [26]. We then evaluated the predictive accuracy of those minimal abductive hypotheses using *a leave-one-out strategy* [5].

First, we select one clause C in observations E as a test example. Second, for each new characteristic clause D in $NewCarc(B, \overline{E - \{C\}}, \mathcal{P})$, we check whether or not the minimal abductive hypothesis $M(\{D\})$ derives the test example C with respect to B, that is, $B \wedge M(\{D\}) \models \{C\}$. In case of the simply pathway problem, we assume E is ground, and thus every new characteristic clause in $NewCarc(B, \overline{E - \{C\}}, \mathcal{P})$ is also ground.

N_H and N_P in Table 4.1 denote the number of minimal abductive hypotheses with respect to B, $E - \{C\}$ and $\mathcal{P}$ and the number of those hypotheses that can also explain the test example C. This experimental result shows that CF-induction can generate abductive hypotheses such that accurately predict each (unseen) test example. However, we may notice that the ratio of N_P to N_H is not so high. It means that though several abductive hypotheses succeeded with the prediction of a unseen test example, most of them could not do. This fact shows the necessity of not only abductive but also inductive inference to improve the predictive accuracy of hypotheses.

4.3.2 Metabolic Pathway of Pyruvate

Next, we consider the metabolic pathway of Pyruvate (see Figure 4.8). The logical representation of topology in Figure 4.8 is as follows:

$$\begin{aligned} T' = \{&terminal(ethanol), reac(pyruvate, acetylcoa),\\ &reac(pyruvate, acetaldehyde), reac(glucose, glucosep),\\ &reac(glucosep, pyruvate), reac(acetaldehyde, acetate),\\ &reac(acetate, acetylcoa), reac(acetaldehyde, ethanol)\},\end{aligned}$$

where the predicate $terminal(X)$ means that there is no reaction where X is consumed. If the metabolite X is terminated and there is activated reaction that produces X, then the concentration of X must increase. However, this consequence cannot be derived only using the previous causal rule (4.3) which concerns with the concentration increase of a metabolite. Thus, we construct the following new causal rules obtained by incorporating the con-

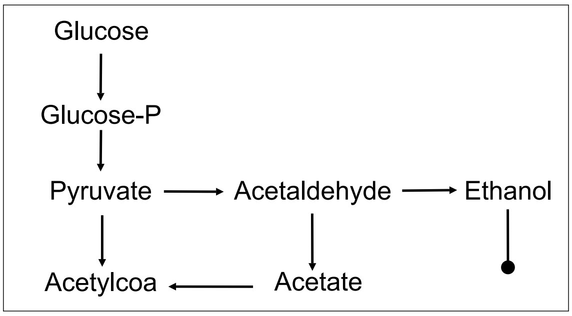


Figure 4.8: Metabolic Pathway of Pyruvate

cept of "*terminal*" with the rule (4.3):

$$reac(X, Z) \land \neg act(X, Z) \supset blocked(X), \tag{4.6}$$

$$terminal(X) \supset blocked(X), \tag{4.7}$$

$$reac(Y, X) \land act(Y, X) \land blocked(X) \supset con(X, up), \tag{4.8}$$

where the predicate $blocked(X)$ means that the metabolite X cannot be consumed, and the predicate $terminal(X)$ means that the metabolite X is terminated. Define a background theory B as follows:

$$B = T' \cup \{(4.6), (4.7), (4.8), con(glucose, up)\}.$$

Next, we input the following observations E:

$$E = \{con(ethanol, up), con(pyruvate, up)\}.$$

Then, the following clausal theories H_3 and H_4 are considerable as hypotheses with respect to B and E.

$$\begin{aligned} H_3 = \{&act(glucosep, pyruvate), \\ &\neg act(pyruvate, acetylcoa), act(acetaldehyde, ethanol)\}. \\ H_4 = \{&act(X, Y) \land con(X, up) \supset con(Y, up), \\ &act(glucosep, pyruvate), act(acetaldehyde, ethanol), \\ &act(pyruvate, acetaldehyde), act(glucose, glucosep)\}. \end{aligned}$$

Figure 4.9 and 4.10 show the states of reactions in H_3 and H_4, respectively. Compared with H_3, H_4 includes a new general rule concerning the mechanism of how concentrations of metabolites increase. Both H_3 and H_4 can be

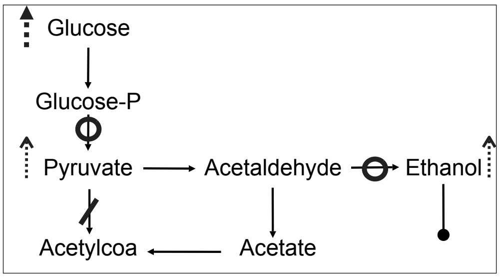


Figure 4.9: Third Hypothesis H_3

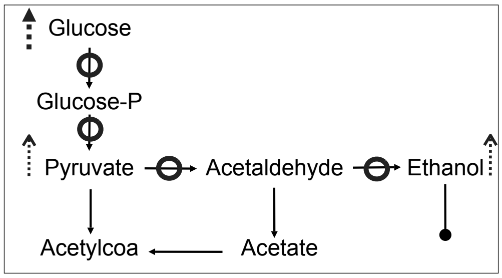


Figure 4.10: Forth Hypothesis H_4

generated using CF-induction. Let a production field $\mathcal{P}$ be as follows:

$$\mathcal{P} = \langle \{con(X,Y), \neg con(X,Y), \neg act(X,Y), act(X,Y)\}\rangle.$$

We set the maximum length of characteristic clauses and the maximum search-depth as 6 and 4, respectively. Then, CF-induction displays the following $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$ and $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$, which consist of 6 and 3 clauses, respectively.

```
NewCarc:
(1)[-act(acetaldehyde,ethanol),-con(pyruvate,up)]
(2)[-con(ethanol,up),-act(glucosep,pyruvate),
    act(pyruvate,acetylcoa)]
(3)[-con(ethanol,up),-act(glucosep,pyruvate),
    act(pyruvate,acetaldehyde)]
(4)[-act(acetaldehyde,ethanol),-act(glucosep,pyruvate),
    act(pyruvate,acetylcoa)]
(5)[-act(acetaldehyde,ethanol),-act(glucosep,pyruvate),
    act(pyruvate,acetaldehyde)]
(6)[-con(ethanol,up),-con(pyruvate,up)]

Carc:
(1)[con(glucose,up)]
(2)[-act(_0,_1),act(_0,_1)]
(3)[con(_0,_1),-con(_0,_1)]
```

We first manually construct the bridge theory CC_3 which contains only the 4th clause in $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$. The system automatically computes the minimal complement of CC_3, which corresponds to H_3. Note that no generalizer is necessary to generate H_3. We next construct the bridge formula CC_4 for finding H_4 which contains the two 1st and 5th clauses in $NewCarc(B, \overline{E_{sk}}, \mathcal{P})$ and the 1st clause, four instances of the 2nd clause and one instance of the 3rd clause in $Carc(B \wedge \overline{E}, \mathcal{P})$. Note that CC_4 totally contains 8 clauses. After computing the minimal complement of CC_4, CF-induction outputs H_4 using both a dropping and an anti-instantiation generalizers.

4.4 Related Work

Tamaddoni-Nezhad et al. studied estimation of inhibitory effects on metabolic pathways using ILP [76]. Inhibition of enzyme functions plays an important role on dynamically controlling enzymatic activities on metabolism. They have also introduced the logical modeling of metabolic pathways and firstly showed the possibility of application of ILP to qualitatively analyzing metabolic pathways with their logical model on inhibitory effects to concentration changes of metabolites. Here, we refer to three points as the difference, compared with their work.

The first point is the difference of main goal that we try to solve using ILP techniques. Our long-term goal is more precise modeling of the flux distribution corresponding to dynamics on enzymatic activities. Previously proposed techniques in MFA that quantitatively analyzes the flux distribution can have several limitations involved in embedded approximation conditions as well as computational costs in the case of large-scale of metabolic pathways. Using ILP techniques, we can logically estimate possible reaction states in metabolic pathways, which enable us to reconstruct the prior pathway into more simplified one by removing enzyme reactions with low fluxes. In other words, we intend to reduce the complexity of metabolic pathways. In contrast, they have focused on the inhibitory effects of particular toxins affected to objects to be examined, which are used as drugs.

The second point is the difference of ILP techniques applied to problems. They have used Progol5.0 [46] that is one of the successful ILP systems and can also compute both inductive and abductive hypotheses. Compared with Progol5.0, CF-induction preserves the soundness and completeness for finding hypotheses and can use not only Horn but also non-Horn clauses in the knowledge representation formalisms. In [76], they have evaluated the predictive accuracy of both abductive and inductive hypotheses obtained by Progol5.0 in the metabolic pathway consisting of 76 enzyme reactions and 31 metabolites. The data set of this metabolic pathway is available on their website. Then, it will be interesting that we evaluate the hypotheses obtained by CF-induction using the data set. Note that in an initial experiment with this data set, CF-induction computes 66 (subsumption-minimal) abductive hypotheses including the unique output of Progol5.0 when we set the maximum search-depth as 5 and put the maximum length of a production field as 15. The following example is concerning with an inductive hypothesis introduced in [76]. We show how CF-induction can compute the same inductive hypothesis as the one obtained by Progol5.0 using the next example.

Example 4.1. Define a background theory B and an observation E as follows:

$$\begin{aligned} B = \{&reac(X,Y,Z) \wedge inh(X,Y,Z) \supset con(X,up), \\ &reac(s,e,p), class(e,c)\}, \\ E = \ &con(s,up). \end{aligned}$$

The predicate $reac(X,Y,Z)$ includes the new term Y which denotes an enzyme catalyzing the reaction $X \rightarrow Z$. The predicate $inh(X,Y,Z)$ means the reaction between a substrate X and a product Z, catalyzed by an enzyme Y, is inhibited. Note that those reactions that are inhibited might be regarded as non-activated reactions. However, the activated state does not necessarily always correspond to the non-inhibited state. Because there are other factors expects for inhibition, which make enzyme reactions inactivated. We thus need to distinguish the "inhibited" state with the "non-activated" state. Along with the predicate $inh(X,Y,Z)$, the predicate $class(X,Y)$ is newly introduced. It means that an enzyme X belongs to an enzyme class Y. In [76], the following inductive hypothesis H is introduced:

$$H = inh(s,X,p) \leftarrow class(X,c).$$

This hypothesis can be derived by CF-induction. Suppose the following production field $\mathcal{P}$:

$$\mathcal{P} = \langle \{inh(X,Y,Z), \neg inh(X,Y,Z), class(X,Y)\} \rangle.$$

$$
\begin{array}{ll}
\mathbf{B}\text{:} & con(X, up) \leftarrow reac(X, Y, Z) \wedge inh(X, Y, Z), \quad reac(s, e, p), \;\; class(e, c) \\
\mathbf{E}\text{:} & con(s, up) \\
\hline
 & (abduction) \quad inh(s, e, p) \\
\hline
 & (induction) \qquad\qquad\qquad inh(s, X, p) \leftarrow class(X, c)
\end{array}
$$

Figure 4.11: Integrating Abduction and Induction for Finding Hypotheses

$NewCarc(B, \overline{E_{sk}}, \mathcal{P})$ and $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$ are computed as follows:

$$
\begin{aligned}
NewCarc(B, \overline{E_{sk}}, \mathcal{P}) &= \{\neg inh(s, e, p), \neg con(s, up)\}, \\
Carc(B \wedge \overline{E_{sk}}, \mathcal{P}) &= \{class(e, c), \neg inh(s, e, p)\}.
\end{aligned}
$$

Let a bridge formula CC be the clausal theory $\{\neg inh(s, e, p), class(e, c)\}$. Then, the minimal complement of CC is as follows:

$$M(CC) = inh(s, e, p) \leftarrow class(e, c).$$

If we apply an anti-instantiation generalizer to $M(CC)$ in such a way that the ground term e appearing in F is replaced with the variable X, then the hypothesis H can be generated.

The process of computing the above inductive hypothesis can be sketched in Figure 4.11. An inductive hypothesis is constructed based on a certain abductive explanation. Hence, it is necessary to generate abductive hypotheses in advance for constructing inductive hypotheses. Indeed, Progol5.0 first computes an abductive hypothesis, and second find inductive hypotheses using the abductive explanation. In other words, Progol5.0 performs abduction and induction processes, step by step. On the other hand, CF-induction can realize them with one process. This difference of those mechanisms for integrating abduction and induction is the last and crucial point. In our problem setting, we need to not only estimate possible reaction states but also complete missing general rules in the prior background. CF-induction can realize these two tasks simultaneously using both abduction and induction.

There are several works that have studied on applications of ILP techniques to biology [32, 94], as we described in Chapter 1. King et al. showed that hypothesis finding techniques can help to reduce the experimental costs for predicting the functions of genes in their project, called *robot sceientist* [32]. Besides, Zupan et al. have developed a system based on abduction, which enables us to find new relations from experimental genetic data for completing the prior genetic networks [94].

4.5 Summary

We have studied a logic-based method for estimating possible states of enzyme reactions. Since this method can help us to understand which enzyme reactions are activated, it can be potentially used in MFA, in that it is important for optimization or improvement of production and to identify master reactions that have high fluxes. In this work, we have showed how CF-induction can realize not only estimation of possible reaction states, but also completion of the current causal relations.

On the other hand, it is still far from our long-term goal that we construct a new technique integrating our logic-based analysis with previously proposed MFA methods. As an important remark, we point out the issue on completeness and non-determinism in hypothesis fining techniques. Systems like Progol can deterministically generate a hypothesis. However, since it does not ensure completeness in hypothesis finding, some important hypotheses might fail to be generated. In contrast, CF-induction ensures the completeness, and thus can achieve such an advanced inference that simultaneously integrates abduction and induction. However, since CF-induction consists of several highly non-deterministic procedures, it is not straightforward to apply to practical problems that have to deal with a huge number of data. In next chapter, we treat this issue and investigate how the non-determinism in generalization of IE-based ILP methods can be logically reduced.

Chapter 5

From Inverse Entailment to Inverse Subsumption

As we explained in Chapter 3, modern explanatory ILP methods like Progol, Residue procedure, CF-induction, HAIL and Imparo use the principle of Inverse Entailment (IE). It is based on the fact that the negation of a hypothesis is derived from a prior background theory and the negation of examples. IE-based methods commonly compute a hypothesis in two steps: by first constructing an intermediate theory and next by generalizing its negation into the hypothesis with the inverse relation of entailment. In this chapter, we focus on the sequence of intermediate theories that constructs a derivation from the background theory and the negation of examples to the negation of the hypothesis. We then show the negations of those derived theories in a sequence are represented with inverse subsumption. Using our result, inverse entailment can be reduced into inverse subsumption, while it preserves the completeness for finding hypotheses.

5.1 Introduction

Given a background theory B and observations E, the task of explanatory induction is to find a hypothesis H such that $B \wedge H \models E$ and $B \wedge H$ is consistent [26]. By the principle of Inverse Entailment (IE) [44], this is logically equivalent to finding a consistent hypothesis H such that $B \wedge \neg E \models \neg H$. This equivalence means that the inductive hypothesis H can be computed by deducing its negation $\neg H$ from B and $\neg E$. We represent this derivation process as follows:

$$B \wedge \neg E \models F_1 \models \cdots \models F_i \models \cdots \models F_n \models \neg H \tag{5.1}$$

where each F_i ($1 \leq i \leq n$) denotes a clausal theory. IE-based methods [26, 31, 44, 61, 63, 75, 83] compute a hypothesis H in two ways: by first constructing an intermediate theory F_i in Formula (5.1) and next generalizing its negation $\neg F_i$ into the hypothesis H with the inverse relation of entailment. The logical relation between $\neg F_i$ and H is obtained from the contrapositive of Formula (5.1) as follows:

$$\neg(B \wedge \neg E) \mathrel{=\!\!|} \neg F_1 \mathrel{=\!\!|} \cdots \mathrel{=\!\!|} \neg F_i \mathrel{=\!\!|} \cdots \mathrel{=\!\!|} \neg F_n \mathrel{=\!\!|} H \tag{5.2}$$

where $\mathrel{=\!\!|}$ denotes the inverse relation of entailment. Hereafter, we call it the *generalization relation*. In brief, IE-based methods first use the entailment relation to construct F_i with Formula (5.1), and then switch to the generalization relation to generate the hypothesis H with Formula (5.2) (See Figure 5.1).

$$\overline{B \wedge \neg E \models F_1 \models \cdots \models F_i} \quad \text{(Generalization relation)}$$
$$\neg F_i \mathrel{=\!\!|} \cdots \mathrel{=\!\!|} \neg F_n \mathrel{=\!\!|} H$$

Figure 5.1: Hypothesis Finding Based on Inverse Entailment

The inverse relation of entailment ensures the completeness in finding hypotheses from an intermediate theory by Formula (5.2). However, the generalization procedures with this relation need a variety of different generalizers, such as dropping and inverse resolution [47]. As shown in Chapter 3, there are several such operators and each one can be applied in many different ways, which lead to a large number of choice points. For this reason, systems like Progol [44, 75] and HAIL [61, 63] use subsumption due to computational efficiency, though their generalization procedures may become incomplete. On the other hand, systems like CF-induction [26] use entailment to find any hypothesis, though they need to handle a huge search space.

In this chapter, we show inverse subsumption is an alternative generalization relation to ensure the completeness for finding hypotheses (See Figure 5.2).

$$\overline{B \land \neg E \models F_1 \models \cdots \models F_i} \quad \text{(Generalization relation)}$$
$$\neg F_i \preceq \cdots \preceq \neg F_n \preceq H$$

Figure 5.2: Hypothesis Finding Based on Inverse Subsumption

Our result is used to reduce the non-deteriminism in generalization without losing the completeness. The key idea lies in that for two ground clausal theories S and T such that $S \models T$, $\neg S$ and $\neg T$ translated into CNF are represented with inverse subsumption. This feature is applied to the logical relation $F_i \models \neg H$ where F_i is an intermediate theory and H is a hypothesis in Formula (5.1). We then obtain its alternative relation $\neg F_i \preceq H$ represented by inverse subsumption. Since F_i is a CNF formula, there are several ways to represent $\neg F_i$ in CNF.

This chapter focuses on two CNF formulas translated $\neg F_i$ into CNF, that is, *residue* and *minimal* complements[1], respectively. We show inverse subsumption with residue and minimal complements ensure completeness of generalization in Section 5.2 and 5.3, respectively. In Section 5.4, we conclude.

5.2 Inverse Subsumption with Residue Complements

Our approach is based on the fact that for two ground clausal theories S and T such that $S \models T$, the logical relation between $\neg S$ and $\neg T$ translated in CNF is represented by inverse subsumption. We intend to apply this feature to Formula (5.1), later. Since $\neg S$ and $\neg T$ are DNF formulas, there are several ways to represent $\neg S$ and $\neg T$ in CNF. In this section, we use the residue complement and consider the logical relation between $R(S)$ and $R(T)$, which is represented primarily by the following theorem:

Theorem 5.1 ([83])**.** Let S and T be two clausal theories such that T is ground and both S and T do not include tautological clauses. If $S \models T$, there is a finite subset U of ground instances from S such that $R(T) \succeq R(U)$.

By Theorem 5.1, the following, which deals with the ground case, holds:

[1]See P. 42.

Proposition 5.1 ([90])**.** Let S and T be two ground clausal theories such that S and T do not include tautological clauses. If $S \models T$, then $R(T) \succeq R(S)$.

We first recall the following lemma to prove Proposition 5.1.

Lemma 5.1 ([83])**.** For ground clausal theories S and T that do not include tautological clauses, $T \subset S$ implies $R(T) \succeq R(S)$.

Using Lemma 5.1 as well as Theorem 5.1, Proposition 5.1 is proved:

Proof. By Theorem 5.1, there is a ground theory U such that $U \subseteq S$ such that $R(T) \succeq R(U)$. By Lemma 5.1, $R(U) \succeq R(S)$ holds. Hence, $R(T) \succeq R(S)$ holds. □

We apply this proposition to the logical relation $F \models \neg H$ where F is a bridge theory[2] and H is a ground hypothesis. We represent $\neg H$ using the residue complement $R(H)$. Suppose that F does not include any tautological clauses. Then, by Propositon 5.1, $R^2(H) \succeq R(F)$ holds. In other words, $R^2(H)$, which is logically equivalent to H, can be obtained from $R(F)$ using inverse subsumption. This property is described as the following theorem:

Theorem 5.2 ([90])**.** Let F be a bridge theory such that F do not include tautological clauses, and H a hypothesis such that $F \models \neg H$. Then, there is a hypothesis H^* such that $H^* \equiv H$ and $H^* \succeq R(F)$.

Proof. By Herbrand theorem, there is a ground clausal theory H_g such that $H \succeq H_g$ and $F \models \neg H_g$. Since $\neg H_g \equiv R(H_g)$ holds, $R^2(H_g) \succeq R(F)$ holds by Proposition 5.1. Assume the clausal theory $H^* = H \cup R^2(H_g)$. Since $H \succeq H_g$ and $H_g \equiv R^2(H_g)$, $H \models R^2(H_g)$ holds. Accordingly, $H \models H^*$ holds. Hence, $H^* \equiv H$ holds. Since $R^2(H_g) \succeq R(F)$ and $H^* \supseteq R^2(H_g)$, $H^* \succeq R(F)$ holds. □

However, every target hypothesis is not necessarily obtained from the residue complement by inverse subsumption, as described in the below examples.

[2]We refer to P.46 for the formal definition of bridge theories.

Example 5.1. Let B, E and H be a background theory, observations and a target hypothesis as follows:

$$B = \{p(a)\}.\ E = \{p(f(f(a))\}.$$
$$H = \{\{p(a) \supset p(f(a)),\ p(f(a)) \supset p(f(f(a)))\}\}.$$

Let F be the clausal theory $\{p(a), \neg p(f(f(a)))\}$. Since $F = B \cup \neg E$, F is a bridge theory wrt B and E such that $F \models \neg H$. $R(F)$ is $\{p(a) \supset p(f(f(a)))\}$. Then we notice that $R(F)$ is not subsumed by H. Indeed, $R(F)$ is the resolvent of two clauses in H. Then, we may need to apply an inverse resolution operator to $R(F)$ for obtaining the target hypothesis H.

Example 5.2. We recall Example 3.3 in Chapter 3 as follows:

$$B = \{arc(a,b),\ arc(X,Y) \wedge path(Y,Z) \supset path(X,Z)\}.$$
$$E = \{path(a,c)\}.$$
$$H = \{arc(b,c),\ arc(X,Y) \supset path(X,Y)\}.$$

Let F be the clausal theory as follows:

$$F = \{arc(a,b),\ arc(a,b) \wedge path(b,c) \supset path(a,c),\ \neg path(a,c)\}.$$

Since F is the set of ground instances from $B \wedge \neg E$, F is a bridge theory wrt B and E. Assume the following ground hypothesis H_g consisting of ground instances of H:

$$H_g = \{arc(b,c),\ arc(X,Y) \supset path(X,Y)\}.$$

Since $F \wedge H_g$ is inconsistent, F satisfies the condition of Theorem 5.2. Hence, there is a hypothesis H^* such that $H^* \equiv H$ and $H^* \succeq R(F)$. $\overline{F}$ and $R(F)$ are as follows:

$$\begin{array}{rcl} \overline{F} & = & \{\{\neg arc(a,b),\ arc(a,b),\ path(a,c)\}, \\ & & \{\neg arc(a,b),\ path(b,c),\ path(a,c)\}, \\ & & \{\neg arc(a,b),\ \neg path(a,c),\ path(a,c)\}\}. \\ R(F) & = & \{\{\neg arc(a,b),\ path(b,c),\ path(a,c)\}\} \end{array}$$

We then consider the following clausal theory as H^*:

$$H^* = H \cup \{path(b,c)\}.$$

H^* is equivalent to H, since the added clause $C = \{path(b,c)\}$ can be derived from H. We may notice that C subsumes the clause in $R(F)$. Then, $H^* \succeq R(F)$ holds. Hence, there is in fact a hypothesis H^* such that $H^* \equiv H$ and $H^* \succeq R(F)$. However, the target hypothesis H itself does not subsume $R(F)$. Instead, we may notice that the clause C is a resolvent of two clauses in H. Then, it is necessary to use an inverse resolution operator for obtaining the target hypothesis H.

This problem is caused by the fact that $R^2(H) = H$ cannot necessarily hold for a ground clausal theory H. The key idea in Theorem 5.2 lies in the logical relation $R^2(H) \succeq R(F)$. Hence, if $R^2(H) = H$ should not hold, H cannot be obtained from $R(F)$ using inverse subsumption. We thus need some CNF formula $F(H)$ for representing the negation of a hypothesis H such that $F(F(H)) = H$.

5.3 Inverse Subsumption with Minimal Complements

We here investigate an alternative generalization using minimal complements. We first show a key property that minimal complements satisfy.

5.3.1 Fixed-Point Theorem on Minimal Complements

Theorem 5.3 ([89])**.** Let S be a ground clausal theory. Then $M^2(S) = \mu S$ holds.

Proof. By Theorem 2.5, $M^2(S) = MHS^2(S)$ holds. By Lemma 5.2, it holds that $MHS^2(S) = \mu S$. Hence, $M^2(S) = \mu S$ holds. □

The above proof uses the below lemma as well as Theorem 2.4 and Theorem 2.5, which concerned a property on minimal hitting sets and a relation between minimal hitting sets and minimal complements, respectively.

Lemma 5.2 ([89])**.** Let S be a ground clausal theory. Then $MHS^2(S) = \mu S$ holds.

(*Proof* of $\mu S \subseteq MHS^2(S)$.)
We show every clause in μS is a minimal hitting set of $\mathcal{F}(MHS(S))$. By the definition of $MHS(S)$, for every clause $D \in MHS(S)$, D satisfies that $D \cap C \neq \emptyset$ for every set $C \in \mathcal{F}(S)$. In other words, for every set $C \in \mathcal{F}(S)$, C satisfies that $D \cap C \neq \emptyset$ for every clause $D \in MHS(S)$. Then every set $C \in \mathcal{F}(S)$ is a hitting set of $MHS(S)$. Let C' be the set of negations of literals in C. Since $C \in \mathcal{F}(S)$, $C' \in \mathcal{F}(\mathcal{F}(S))$ holds. Since C is a hitting set of $MHS(S)$, C' is a hitting set of $\mathcal{F}(MHS(S))$. Accordingly, every set $C \in \mathcal{F}(\mathcal{F}(S))$ is a hitting set of $\mathcal{F}(MHS(S))$. Since the family $\mathcal{F}(\mathcal{F}(S))$ corresponds to S, it holds that every clause $C \in S$ is a hitting set of $\mathcal{F}(MHS(S))$. Since $\mu S \subseteq S$, every clause $C \in \mu S$ is a hitting set of $\mathcal{F}(MHS(S))$.

Suppose that there is a clause $C \in \mu S$ such that C is not a *minimal* hitting set of $\mathcal{F}(MHS(S))$. Then

(*) there is a literal $l \in C$ such that $C - \{l\}$ is a hitting set of $\mathcal{F}(MHS(S))$.

For every clause $C_i \in \mu S$, if $C_i \neq C$ then there is a literal $l_i \in C_i$ such that $l_i \notin C$. We then consider those literals $E = \{l_1,\ l_2,\ \ldots,\ l_n\}$, where each l_i is a literal of $C_i \in \mu S - \{C\}$ such that l_i is not included in C. Note that $E \cap C = \emptyset$ holds. On the other hand, for any literal $l \in C$, the intersection of $E \cup \{l\}$ and each clause in μS is not empty. Hence, $E \cup \{l\}$ is a hitting set of μS. Accordingly, $E \cup \{l\}$ is also a hitting set of S. Then there is a minimal hitting set $E' \in \mathcal{F}(MHS(S))$ such that $E' \subseteq E \cup \{l\}$. Since $E' \subseteq E \cup \{l\}$ and $E \cap C = \emptyset$, it holds that $E' \cap (C - \{l\}) = \emptyset$. However, this contradicts that $C - \{l\}$ is a hitting set of $\mathcal{F}(MHS(S))$ since $E' \in \mathcal{F}(MHS(S))$. Then the assumption (*) is false. Hence, every clause C in μS is a minimal hitting set of $\mathcal{F}(MHS(S))$. In other words, for every $C \in \mu S$, $C \in MHS^2(S)$ holds.

(*Proof* of $MHS^2(S) \subseteq \mu S$.)
Let D be a clause in $MHS^2(S)$. Suppose that there is a clause $C \in \mu S$ such

that $C \subset D$. Since $\mu S \subseteq MHS^2(S)$, $C \in MHS^2(S)$ holds. This contradicts with the minimality of $MHS^2(S)$. Hence, for every clause $C \in \mu S$, $C \not\subset D$. In other words, for ever clause $C \in \mu S$, $C = D$ or $C \not\subseteq D$. Suppose that

$$(*) \text{ for any clause } C \in \mu S,\ C \neq D \text{ holds.}$$

Then, $C \not\subseteq D$ holds. Accordingly, for every clause $C_i \in \mu S$, there is a literal $l_i \in C_i$ such that $l_i \notin D$. We consider the finite set $E = \{l_1,\ l_2,\ \ldots,\ l_n\}$ where each literal l_i corresponds to such the above literal that is not included in D. Note that $E \cap D = \emptyset$. On the other hand, the intersection of E and each clause in μS is not empty. Hence, E is a hitting set of μS. Accordingly, E is a hitting set of S. Then, there is a minimal hitting set E' of S such that $E' \subseteq E$. Since $MHS(S)$ is the set of minimal hitting sets of $\mathcal{F}(S)$, $\mathcal{F}(MHS(S))$ corresponds to the set of minimal hitting sets of $\mathcal{F}(\mathcal{F}(S))$. Since $\mathcal{F}(\mathcal{F}(S)) = S$, $\mathcal{F}(MHS(S))$ is the set of minimal hitting sets of S. Hence, we have $E' \in \mathcal{F}(MHS(S))$. Since $E' \subseteq E$ and $E \cap D = \emptyset$, $E' \cap D = \emptyset$ holds. Note here that since $D \in MHS^2(S)$, D is a minimal hitting set of $\mathcal{F}(MHS(S))$. However this contradicts that there is the set $E' \in \mathcal{F}(MHS(S))$ such that $E' \cap D = \emptyset$. Then the assumption (*) is false. Hence, there is a clause $C \in \mu S$ such that $C = D$. Therefore, $D \in \mu S$ holds. □

Theorem 5.3 can be regarded as a fixed-point theorem on the function M computing the minimal complement. Unlike residue complements, $M^2(S)$ corresponds with S in case that S is subsume-minimal. Thus, minimal complements may not cause the problem in residue complements that they cannot necessarily obtain a target hypothesis using inverse subsumption, as described in Section 5.3.

Example 5.3. Let S be the clausal theory $\{a \vee b,\ b \vee c,\ \neg c\}$. Then, $\overline{S}$, $R(S)$, $R^2(S)$, $M(S)$ and $M^2(S)$ are as follows. In fact, $M^2(S) = S$ holds, whereas

$R^2(S)$ does not. Note that $M(S)$ contains a tautological clause.

$$\overline{S} = \{\neg a \vee \neg b \vee c,\ \neg a \vee \neg c \vee c,\ \neg b \vee c,\ \neg b \vee \neg c \vee c\}.$$
$$R(S) = \{\neg a \vee \neg b \vee c,\ \neg b \vee c\}. \quad R^2(S) = \{a \vee b,\ \neg c \vee a,\ b,\ \neg c \vee b,\ \neg c\}.$$
$$M(S) = \{\neg a \vee \neg c \vee c,\ \neg b \vee c\}.\ M^2(S) = \{a \vee b,\ b \vee c,\ \neg c\}.$$

On the other hand, minimal complements do not necessarily satisfy the logical relation that $M(T) \succeq M(S)$ for ground clausal theories S and T such that $S \models T$. We recall Example 5.1. Whereas $F \models M(H)$ holds, $M^2(H)$, which is equal to H by Theorem 5.3, does not subsume $M(F)$. This is because minimal complements can include tautological clauses that residue complements never have. Indeed, Proposition 5.1, which shows the logical relation between $R(T)$ and $R(S)$, does not allow tautological clauses to be included in S and T. In next subsection, we extend Proposition 5.1 so as to deal with tautological clauses.

5.3.2 Inverting Deductive Operations with Tautologies

Theorem 5.4 ([90])**.** Let S and T be ground clausal theories such that $S \models T$ and for every tautological clause $D \in T$, there is a clause $C \in S$ such that $C \succeq D$. Then, $\tau M(T) \succeq \tau M(S)$. holds.

We use the following deductive operators for proving this theorem:

Definition 5.1 (Deductive operators [83])**.** Let S and T be clausal theories. Then T is *directly-derivable* from S if T is obtained from S by one of the following three operators:

1. (resolution) $T = S \cup \{C\}$, where C is a resolvent of two clauses $D_1, D_2 \in S$.
2. (subsumption) $T = S \cup \{C\}$, where C is subsumed by some clause $D \in S$.
3. (weakening) $T = S - \{D\}$ for some clause $D \in S$.

We write $S \vdash_r T$, $S \vdash_s T$, $S \vdash_w T$ to denote that T is directly derivable from S by resolution, subsumption, weakening, respectively. $\vdash_X^*$ is the reflexive and transitive closure of $\vdash_X$, where X is one of the symbols r, s, w. Alternatively, $S \vdash_X^* T$ if T follows from S by application of zero or more $\vdash_X$.

We now show that entailment between theories can be established by applying these operators in a particular order.

Theorem 5.5 ([92])**.** Let S be a clausal theory and T be a clausal theory without any tautological clauses. If $S \models T$, then there are two clausal theories U, V such that

$$S \vdash_r^* U \vdash_s^* V \vdash_w^* T.$$

Proof. Let $T = \{C_1, \ldots, C_n\}$. Then $S \models C_i$ for each clause C_i in T. By the Subsumption Theorem there is a derivation $R_1^i, \ldots, R_{m_i}^i$ from S of a clause $R_{m_i}^i$ that subsumes C_i. Hence, it is sufficient to let $U = S \cup \{R_j^i : 1 \leq i \leq n, 1 \leq j \leq m_i\}$ and $V = U \cup T$. $\square$

Using Theorem 5.5, we obtain the following lemma that allows tautological clauses to be included in S and T:

Lemma 5.3 ([90])**.** Let S and T be ground clausal theories such that $S \models T$ and for every tautological clause $D \in T$, there is a clause $C \in S$ such that $C \succeq D$. Then there are two ground clausal theories U and V such that

$$S \vdash_r^* U \vdash_s^* V \vdash_w^* T.$$

Proof. We denote two sets of tautological clauses of S and T as $Taut_S$ and $Taut_T$, respectively. For every clause $D \in Taut_T$, there is a clause $C \in Taut_S$ such that $C \succeq D$. Then there is a ground clausal theory V_t such that $Taut_S \vdash_s^* V_t \vdash_w^* Taut_T$. By Lemma 5.5, there are ground clausal theories U and V such that $S - Taut_S \vdash_r^* U \vdash_s^* V \vdash_w^* T - Taut_T$. Therefore, it holds that

$$S \vdash_r^* U \cup Taut_S \vdash_s^* V \cup V_t \vdash_w^* T. \ \square$$

Example 5.4. We recall the two bridge theories and the two target (ground) hypotheses in Example 5.1 and 5.5, respectively. Figure 5.3 corresponds to Example 5.1. In Figure 5.3, the minimal complement $M(H)$ of the target hypothesis H is derived from the bridge theory F with a tautological clause (See the dotted surrounding parts) using the subsumption and weakening operators. Figure 5.4 corresponds to Example 5.5. In Figure 5.4, the minimal complement $M(H)$ of the target hypothesis H is derived from the bridge theory F with a tautological clause using the three deductive operators.

F_1 with $Taut$		V		$M(H_1)$
$p(a)$		$p(a)$		$p(a) \vee p(f(a))$
$\neg p(f^2(a))$	$\vdash_s^3$	$p(a) \vee p(f(a))$	$\vdash_w^2$	$\neg p(f^2(a)) \vee p(a)$
		$\neg p(f^2(a)) \vee p(a)$		$\neg p(f^2(a)) \vee \neg p(f(a))$
$\boxed{\neg p(f(a)) \vee p(f(a))}$		$\neg p(f^2(a))$		$\boxed{\neg p(f(a)) \vee p(f(a))}$
		$\neg p(f^2(a)) \vee \neg p(f(a))$		
		$\boxed{\neg p(f(a)) \vee p(f(a))}$		

Figure 5.3: Deductive Operations in Example 5.1

Based on Lemma 5.3, Theorem 5.4 can be proved by showing $\tau M(T) \succeq \tau M(S)$ in each case that $S \vdash_X T$ holds where X is one of the symbols r, s, w.

Lemma 5.4 ([90])**.** Let S and T be two ground clausal theories such that $S \vdash_r T$. Then, $\tau M(T) \succeq \tau M(S)$ holds.

Proof. Since $S \vdash_r T$, we write T as $S \cup \{C\}$ where C is a resolvent of two clauses C_1 and C_2 in S. Since C_1 and C_2 are ground, the resolvent C is represented as $(C_1 - \{l\}) \cup (C_2 - \{\neg l\}$ for some literal l in C_1. Let C', C'_1 and C'_2 be three sets such that C' (*resp.* C'_1 and C'_2) consists of the negations of literals in C (*resp.* C_1 and C_2). Note that C'_1 and C'_2 are included in $\mathcal{F}(S)$ since $C_1, C_2 \in S$. Let D be a clause in $\tau MHS(S)$. D is a hitting set of $\mathcal{F}(S)$. Then $D \cap C'_1 \neq \emptyset$ and $D \cap C'_2 \neq \emptyset$ hold. Suppose that D is not a hitting set of $\mathcal{F}(S \cup \{C\})$ (*). Then $D \cap C' = \emptyset$ should hold. Since $C' = (C'_1 - \{\neg l\}) \cup (C'_2 - \{l\})$, $D \cap C' = ((C'_1 - \{\neg l\}) \cap D) \cup ((C'_2 - \{l\}) \cap D)$ holds. Since $D \cap C' = \emptyset$, it should hold that $(C'_1 - \{\neg l\}) \cap D = \emptyset$ and

$$
\begin{array}{lclcl}
F_2 \text{ with } Taut & & U & & V \\
\hline
arc(a,b) & & arc(a,b) & & arc(a,b) \\
arc(a,b) \wedge arc(b,c) & & arc(a,b) \wedge arc(b,c) & & arc(a,b) \wedge arc(b,c) \\
\quad \supset path(a,c) & & \quad \supset path(a,c) & & \quad \supset path(a,c) \\
\neg path(a,c) & \vdash_r^2 & \neg path(a,c) & \vdash_s^1 & \neg path(a,c) \\
\boxed{\neg arc(b,c) \vee arc(b,c)} & & arc(b,c) \supset path(a,c) & & arc(b,c) \supset path(a,c) \\
 & & \neg arc(b,c) & & \neg arc(b,c) \\
 & & \boxed{\neg arc(b,c) \vee arc(b,c)} & & \neg arc(b,c) \vee \neg path(b,c) \\
 & & & & \boxed{\neg arc(b,c) \vee arc(b,c)}
\end{array}
$$

$$
\begin{array}{lcl}
V & & M(H_2) \\
\hline
arc(a,b) & & \\
arc(a,b) \wedge arc(b,c) & & \\
\quad \supset path(a,c) & \vdash_w^5 & \neg arc(b,c) \vee \neg path(b,c) \\
\neg path(a,c) & & \boxed{\neg arc(b,c) \vee arc(b,c)} \\
arc(b,c) \supset path(a,c) & & \\
\neg arc(b,c) & & \\
\neg arc(b,c) \vee \neg path(b,c) & & \\
\boxed{\neg arc(b,c) \vee arc(b,c)} & &
\end{array}
$$

Figure 5.4: Deductive Operations in Example 5.5

$(C_2' - \{l\}) \cap D = \emptyset$. Since $D \cap C_1' \neq \emptyset$ and $D \cap C_2' \neq \emptyset$, it holds that $D \cap \{\neg l\} \neq \emptyset$ and $D \cap \{l\} \neq \emptyset$. Then, D has complementary literals $\neg l$ and l. It contradicts that D is not a tautological clause since $D \in \tau MHS(S)$. Thus, the assumption (*) is false. Hence D is a hitting set of $\mathcal{F}(S \cup \{C\})$. Accordingly, there is a clause $E \in MHS(S \cup \{C\})$ such that $E \succeq D$. Since D is not tautology, E is also not tautology, that is, $E \in \tau MHS(S \cup \{C\})$ holds. By Theorem 2.5, it holds that $\tau MHS(S) = \tau M(S)$ and $\tau MHS(S \cup \{C\}) = \tau M(S \cup \{C\})$. Therefore, for each clause $D \in \tau M(S)$, there is a clause $E \in \tau M(S \cup \{C\})$ such that $E \succeq D$. $\square$

Lemma 5.5 ([90])**.** Let S and T be two ground clausal theories such that $S \vdash_s T$. Then, $\tau M(T) \succeq \tau M(S)$ holds.

Proof. Since $S \vdash_s T$, we write T as $S \cup \{C\}$ where C is a clause such that

$D \succeq C$ for some clause $D \in S$. Since D and C are ground, $D \subseteq C$ holds. Let C' and D' be two sets such that C' (*resp.* D') consists of the negations of literals in C (*resp.* D). Note that D' is included in $\mathcal{F}(S)$ since $D \in S$. Let E be a clause in $\tau MHS(S)$. E is a hitting set of $\mathcal{F}(S)$. Then $E \cap D' \neq \emptyset$ holds. Since $D \subseteq C$, $D' \subseteq C'$ holds. Since $D' \subseteq C'$ and $E \cap D' \neq \emptyset$, $E \cap C' \neq \emptyset$ holds. Hence E is a hitting set of $\mathcal{F}(S \cup \{C\})$. Then there is a clause $E' \in MHS(S \cup \{C\})$ such that $E' \succeq E$. Since $E \in \tau MHS(S)$, E is not tautology. Accordingly, E' is also not tautology, that is, $E' \in \tau MHS(S \cup \{C\})$ holds. By Theorem 2.5, it holds that $\tau MHS(S) = \tau M(S)$ and $\tau MHS(S \cup \{C\}) = \tau M(S \cup \{C\})$. Therefore, for each clause $E \in \tau MHS(S)$, there is a clause $E' \in \tau MHS(S \cup \{C\})$ such that $E' \succeq E$. □

Lemma 5.6 ([90])**.** Let S and T be two ground clausal theories such that $S \vdash_w T$. Then, $\tau M(T) \succeq \tau M(S)$ holds.

Proof. Since $S \vdash_w T$, we write T as $S - \{C\}$ where C is a clause in S. Let E be a clause in $\tau MHS(S)$. E is a minimal hitting set of $\mathcal{F}(S)$. Since $T \subset S$, $\mathcal{F}(T) \subset \mathcal{F}(S)$ holds. Then E is a hitting set of $\mathcal{F}(T)$ since $\mathcal{F}(T) \subset \mathcal{F}(S)$. Hence there is a clause $E' \in MHS(T)$ such that $E' \succeq E$. Since E is not tautology, E' is also not tautology, that is, $E' \in \tau MHS(T)$ holds. By Theorem 2.5, it holds that $\tau MHS(S) = \tau M(S)$ and $\tau MHS(T) = \tau M(S)$ holds. Therefore, for each clause $E \in \tau M(S)$, there is a clause $E' \in \tau M(T)$ such that $E' \succeq E$. □

Using Lemma 5.3, 5.4, 5.5 and 5.6, Theorem 5.4 is proved as follows:
(*Proof* of Theorem 5.4.) By Lemma 5.3, there are two ground clausal theories U and V such that

$$S \vdash_r^* U \vdash_s^* V \vdash_w^* T.$$

By Lemma 5.4, $\tau M(U) \succeq \tau M(S)$ holds. By Lemma 5.5, $\tau M(V) \succeq \tau M(U)$ holds. By Lemma 5.6, $\tau M(T) \succeq \tau M(V)$ holds. Hence, the following formula holds:

$$\tau M(T) \succeq \tau M(V) \succeq \tau M(U) \succeq \tau M(S). \ \square$$

Theorem 5.4 enables us to construct an alternative generalization procedure using minimal complements.

5.3.3 Deriving Hypotheses with Induction Fields

To describe the hypotheses that can be found by this, we first introduce the following language bias, called an *induction field*:

Definition 5.2 (Induction field)**.** An *induction field*, denoted by $\mathcal{I}_{\mathcal{H}}$, is defined as $\langle \mathbf{L} \rangle$ where $\mathbf{L}$ is a finite set of literals to be appeared in ground hypotheses. A ground hypothesis H_g *belongs to* $\mathcal{I}_{\mathcal{H}} = \langle \mathbf{L} \rangle$ if every literal in H_g is included in $\mathbf{L}$.

We next define the target hypotheses using the notion of an induction field $\mathcal{I}_{\mathcal{H}}$, together with a bridge theory F as follows:

Definition 5.3 (Hypothesis wrt $\mathcal{I}_{\mathcal{H}}$ and F)**.** Let H be a hypothesis. H is a *hypothesis* wrt $\mathcal{I}_{\mathcal{H}}$ and F if there is a ground hypothesis H_g such that H_g consists of instances from H, $F \models \neg H_g$ and H_g belongs to $\mathcal{I}_{\mathcal{H}}$.

Now, the generalization procedure based on inverse subsumption with minimal complements is as follows:

Definition 5.4. Let B, E and $\mathcal{I}_{\mathcal{H}} = \langle \mathbf{L} \rangle$ be a background theory, observations and an induction field, respectively. Let F be a bridge theory wrt B and E. A clausal theory H is derived by *inverse subsumption with minimal complements* from F wrt $\mathcal{I}_{\mathcal{H}}$ if H is constructed as follows.

Step 1. $Taut(\mathcal{I}_{\mathcal{H}}) := \{\neg A \vee A \mid A \in \mathbf{L} \text{ and } \neg A \in \mathbf{L}\}$;
Step 2. Compute $\tau M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$;
Step 3. Construct a clausal theory H satisfying the condition:

$$H \succeq \tau M(F \cup Taut(\mathcal{I}_{\mathcal{H}})). \qquad (3)$$

Inverse subsumption with minimal complements ensures the completeness for finding hypotheses wrt $\mathcal{I}_{\mathcal{H}}$ and F, by way of (3).

Main Theorem ([90]). Let B, E and $\mathcal{I}_{\mathcal{H}}$ be a background theory, observations and an induction field, respectively. Let F be a bridge theory wrt B and E. For every hypothesis H wrt $\mathcal{I}_{\mathcal{H}}$ and F, H is derived by inverse subsumption with minimal complements from F wrt $\mathcal{I}_{\mathcal{H}}$.

Proof. It is sufficient to prove the following lemma. □

Lemma 5.7 ([90])**.** Let B, E and $\mathcal{I_H}$ be a background theory, observations and an induction field, respectively. Let F be a bridge theory wrt B and E. For every hypothesis H wrt $\mathcal{I_H}$ and F, H satisfies the following condition:

$$H \succeq \tau M(F \cup Taut(\mathcal{I_H})).$$

Proof. Since H is a hypothesis wrt $\mathcal{I_H}$ and F, there is a ground hypothesis H_g such that $H \succeq H_g$, $F \models \neg H_g$ and H_g belongs to $\mathcal{I_H}$. Since $\neg H_g \equiv M(H_g)$, $F \models M(H_g)$ holds. Accordingly, $F \cup Taut(\mathcal{I_H}) \models M(H_g)$ holds. Since H_g belongs to $\mathcal{I_H}$, every literal in H_g is included in $\mathcal{I_H}$. Then, for every tautological clause $D \in M(H_g)$, there is a clause $C \in Taut(\mathcal{I_H})$ such that $C \succeq D$. By Theorem 5.4, $\tau M^2(H_g) \succeq \tau M(F \cup Taut(\mathcal{I_H}))$ holds. Since $\mu H_g = M^2(H_g)$ by Theorem 3, $\tau\mu H_g \succeq \tau M(F \cup Taut(\mathcal{I_H}))$ holds. Since $H_g \supseteq \tau\mu H_g$, $H_g \succeq \tau\mu H_g$ holds. Hence, $H \succeq \tau M(F \cup Taut(\mathcal{I_H}))$ holds. □

Example 5.5. We show how a target hypothesis is derived by inverse subsumption with minimal complements using Example 3.3 on pathway completion:

$B = \{arc(a,b),\ arc(X,Y) \wedge path(Y,Z) \supset path(X,Z)\}$. $E = \{path(a,c)\}$.
$\mathcal{I_H} = \langle\{arc(b,c),\ \neg arc(b,c),\ path(b,c),\ \neg path(b,c)\}\rangle$.
$H = \{arc(b,c),\ arc(X,Y) \supset path(X,Y)\}$.

One arc from b to c and one rule on pathways are missing in B. The task, as we explained in Chapter 3, is to find the hypothesis H that completes these missing fact and rule. To complete H, both abduction and induction must involve, but most current ILP systems cannot compute it. Let F be the clausal theory $\{arc(a,b),\ arc(a,b) \wedge path(b,c) \supset path(a,c),\ \neg path(a,c)\}$. Since F is the set of ground instances from $B \wedge \neg E$, F is a bridge theory wrt B and E. Since there is a ground hypothesis $H_g = \{arc(b,c),\ arc(b,c) \supset path(b,c)\}$ such that H_g consists of instances from H, $F \models \neg H_g$ and H_g belongs to $\mathcal{I_H}$, H is a hypothesis wrt $\mathcal{I_H}$ and F. Then, H could be derived by inverse subsumption with minimal complements. We first compute $Taut(\mathcal{I_H})$.

Then, $Taut(\mathcal{I}_{\mathcal{H}})$ is the set $\{\neg arc(b,c) \lor arc(b,c),\ \neg path(b,c) \lor path(b,c)\}$. After adding $Taut(\mathcal{I}_{\mathcal{H}})$ to F, we compute $\tau M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$ represented as follows.

$$\begin{array}{l} \{\ \neg arc(a,b) \lor path(b,c) \boxed{\lor\, arc(b,c)} \lor path(a,c), \\ \quad \neg arc(a,b) \lor \boxed{\neg arc(b,c) \lor path(b,c)} \lor path(a,c)\ \}. \end{array}$$

We then notice that H subsumes $\tau M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$ (See the dotted surrounding parts). Therefore, H can be derived by inverse subsumption with minimal complements. In contrast, Since $R(F)$ is $\{\neg arc(a,b) \lor path(b,c) \lor path(a,c)\}$, H does not subsume the residue $R(F)$. Hence, H cannot be obtained from the residue complement, whereas the minimal complement can do with inverse subsumption.

5.4 Discussion

In this section, we compare inverse subsumption with minimal complements with the previous approach based on inverse entailment. We recall Example 5.5. In this example, we assumed the following bridge theory:

$$F = \{arc(a,b),\ arc(a,b) \land path(b,c) \supset path(a,c),\ \neg path(a,c)\}.$$

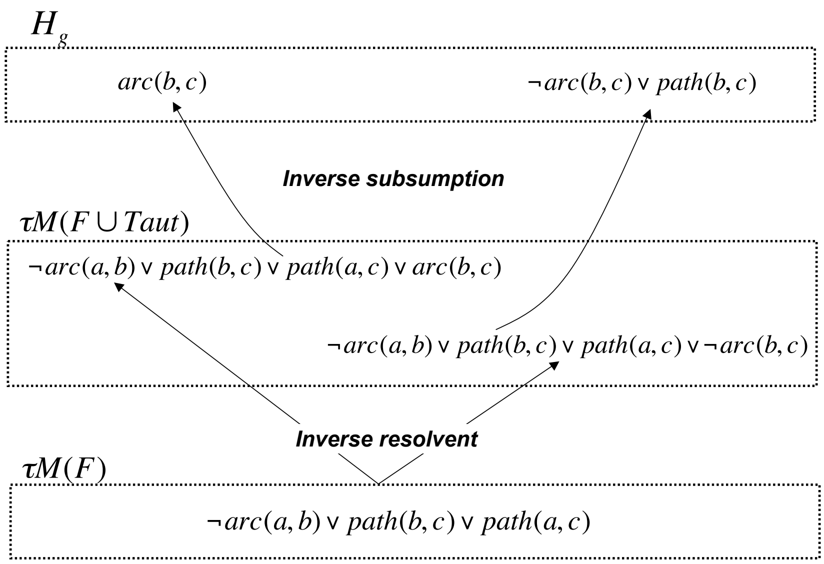
The residue and minimal complement $\tau M(F)$ is as follows:

$$\tau M(F) = \neg arc(a,b) \lor path(b,c) \lor path(a,c).$$

We may notice that $\tau M(F)$ is the resolvent of two clauses C_1 and C_2 in $\tau M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$:

$$C_1 = \neg arc(a,b) \lor path(b,c) \lor path(a,c) \lor arc(b,c).$$
$$C_2 = \neg arc(a,b) \lor path(b,c) \lor path(a,c) \lor \neg arc(b,c).$$

This fact means that adding a tautological clause to the bridge theory F plays a role on an operation of inverse resolution. To deeply view the correspondence between adding tautologies and inverting resolution, we recall



Example 3.2 as follows:

$$\begin{aligned} B &= \{natural(0) \vee even(0)\}. \\ E &= \{natural(s(0))\}. \\ H &= \{natural(0) \supset natural(s(0)),\ even(0) \supset natural(0)\}. \end{aligned}$$

We assume the following induction field $\mathcal{I_H}$ that H belongs to:

$$\mathcal{I_H} = \langle\{natural(0),\ \neg natural(0),\ natural(s(0)),\ \neg even(0),\ natural(s(0))\}\rangle.$$

We assume the same bridge theory $F = \{natural(0) \vee even(0),\ \neg natural(s(0))\}$ in Example 3.2. The residue and minimal complement $\tau M(F)$ is as follows:

$$\tau M(F) = \{\neg natural(0) \vee natural(s(0)),\ \neg even(0) \vee natural(s(0))\}.$$

As we explained in Section 3, the hypothesis H can be derived from $\tau M(F)$ using an inverse resolution operator in such a way that the clause $C_1 = \neg even(0) \vee natural(s(0))$ in $\tau M(F)$ is replaced by a parent clause $C_2 = \neg even(0) \vee natural(0)$ such that C_1 is the resolvent of two clauses C_2 and another clause $C_3 = \neg natural(0) \vee natural(s(0))$ in $\tau M(F)$. See Figure 5.5

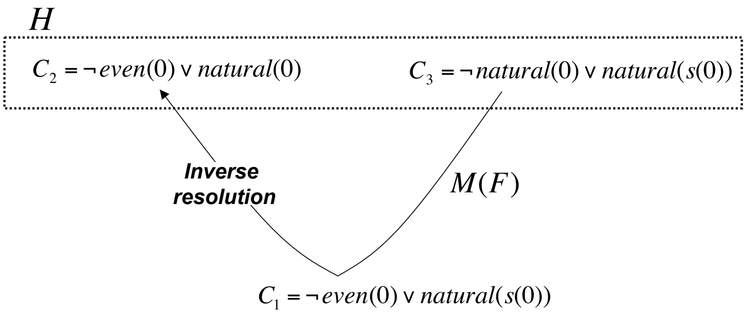


Figure 5.5: Example of Inverse Resolution

which sketches the logical relation between three clauses C_1, C_2 and C_3. In the case of inverse subsumption with minimal complements, we first compute $\tau M(F \cup Taut(\mathcal{I}_\mathcal{H}))$ as follows:

$$\begin{aligned}\tau M(F \cup Taut(\mathcal{I}_\mathcal{H})) &= \{\neg natural(0) \vee natural(s(0)), \\ &\qquad \neg even(0) \vee natural(0) \vee natural(s(0))\}.\end{aligned}$$

We may notice that the clause $\neg natural(0) \vee natural(s(0))$ in $\tau M(F \cup Taut(\mathcal{I}_\mathcal{H}))$ corresponds to C_3 in H, and another clause $C_4 = \neg even(0) \vee natural(0) \vee natural(s(0))$ is subsumed by C_2 in H. Hence, $H \succeq \mu M(F \cup Taut(\mathcal{I}_\mathcal{H}))$ holds. In other words, H can be obtained from $\mu M(F \cup Taut(\mathcal{I}_\mathcal{H}))$ only by using inverse subsumption, strictly speaking, by dropping the literal $natural(s(0))$ from C_4. We also note that the clause C_1 in $\tau M(F)$ is the resolvent of C_3 and C_4 in $\tau M(F \cup Taut(\mathcal{I}_\mathcal{H}))$. Hence, $\tau M(F \cup Taut(\mathcal{I}_\mathcal{H}))$ can be obtained by applying an inverse resolution operator to $\tau M(F)$. This fact implies that adding tautologies to F in fact plays a role inverting resolution. Figure 5.6 describes the logical relation between H, $\tau M(F \cup Taut(\mathcal{I}_\mathcal{H}))$ and $\tau M(F)$.

There is a well-known operator, called $V-operator$ [47], which performs inverting resolution. This operator is given for Horn clauses as follows:

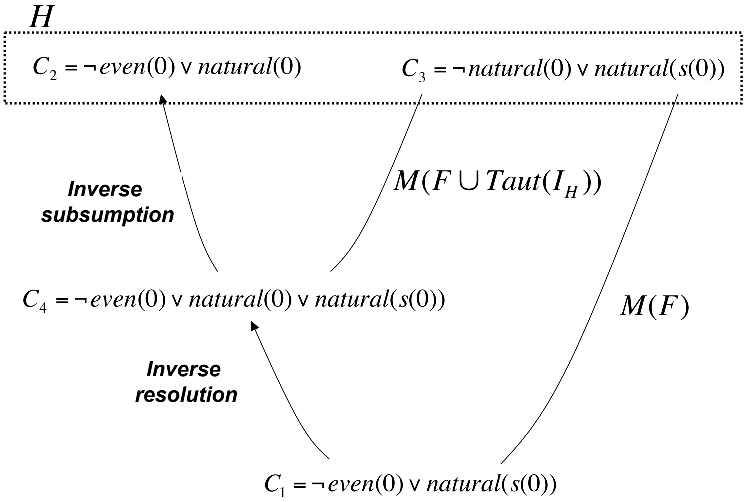


Figure 5.6: Inverting Resolution by Adding Tautologies

Input: Two Horn clauses $C_1 = L_1 \vee C_1'$ and R, where $C_1' \succeq R$.
Output: A Horn clause C_2, such that
R is an instance of a resolvent of C_1 and C_2.

Step 1. Choose a substitution θ_1 such that $C_1'\theta_1 \subseteq R$.

Step 2. Choose an L_2 and C_2' such that $L_1\theta_1 = \neg L_2\theta_2$ and $C_2'\theta_2 = R - C_1'\theta_1$, for some θ_2.

Step 3. Let $C_2 = L_2 \vee C_2'$.

The following figure describes the logical relation between two input clauses C_1 and R and the output clause C_2.

Example 5.6. Let $C_1 = P(X) \vee \neg Q(X)$ and $R = P(f(Y)) \vee \neg Q(Y)$. We here assume that $L_1 = P(X)$. Then, $C_1' = \neg Q(X)$ holds.

1. There is only one substitution θ_1 such that $C_1'\theta_1 \subseteq R$, namely $\theta_1 = \{X/Y\}$.

2. L_2 and C_2' should be such that, for some θ_2, $L_1\theta_1 = P(Y) = \neg L_2\theta_2$ and $R - C_1'\theta_1 = P(f(Y)) = C_2'\theta_2$. We here choose $\theta_2 = \{Z/Y\}$. Then, it holds that $L_2 = \neg P(Z)$ and $C_2' = P(f(Z))$.

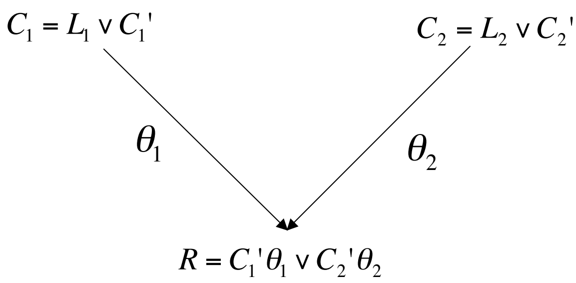


Figure 5.7: Setting for V-operator

3. Let $C_2 = L_2 \vee C_2' = \neg P(Z) \vee P(f(Z))$.

There are several ways to select L_1 and θ_2. The V-operator is then non-deterministic algorithm.

The hypothesis H in Figure 5.5 and 5.6 can be obtained by applying a V-operator to $M(F)$. On the other hand, $M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$ itself cannot be obtained using V-operator, though $M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$ should be also obtained by inverse resolution. In this sense, V-operator is sound but incomplete in performing inverse resolution. Previously, it has not been studied how inverse resolution can be completely performed due to its huge computational complexity. On the other hand, using our result to logically reduce it into inverse subsumption, inverse resolution is embedded by add tautological clauses associated with a given induction field to the bridge theory.

Example 5.7. Let B, E and H be a background theory, observations and a hypothesis as follows:

$$\begin{aligned}
B = \quad & \{plus(0,0,0)\}. \\
E = \quad & \{plus(s(0), s(0), s^2(0))\}. \\
H = \quad & \{plus(X,Y,Z) \supset plus(s(X), Y, s(Z)), \\
& plus(X,Y,Z) \supset plus(X, s(Y), s(Z))\}.
\end{aligned}$$

Note that the predicate $plus(X,Y,Z)$ means $X + Y = Z$. For instance, E says $1+1 = 2$. H corresponds to two missing rules that define the predicate

$plus$. Let F be the following bridge theory wrt B and E:

$$F = \{plus(0,0,0),\ \neg plus(s(0),s(0),s^2(0))\}.$$

Note that F is equal to $B \cup \neg E$. $\tau M(F)$ is the clause $\neg plus(0,0,0) \vee plus(s(0),s(0),s^2(0))$. For deriving H from $\tau M(F)$, we need to perform inverting resolution in some way. However, V-operator cannot be applied to F, since F contain only one clause. Besides, any systematic operator that enables to derive the target H has not been proposed so far. In contrast, our approach based on inverse subsumption can give an insight to perform sufficient inverse resolution by introducing an induction field $\mathcal{I}_{\mathcal{H}}$. For instance, let $\mathcal{I}_{\mathcal{H}}$ be as follow:

$$\begin{aligned}\mathcal{I}_{\mathcal{H}} = \langle\{\ & plus(0,0,0),\ \neg plus(0,0,0),\\ & plus(s(0),s(0),s^2(0)),\ \neg plus(s(0),s(0),s^2(0)),\\ & plus(s(0),0,s(0)),\ \neg plus(s(0),0,s(0))\}\rangle.\end{aligned}$$

One atom $plus(s(0),0,s(0))$ is newly introduced in $\mathcal{I}_{\mathcal{H}}$, which does not appear in B and E. Then, $\tau M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$ is as follows:

$$\begin{aligned}\{&\neg plus(0,0,0) \vee plus(s(0),s(0),s^2(0)) \vee \neg plus(s(0),0,s(0)),\\ &\neg plus(0,0,0) \vee plus(s(0),s(0),s^2(0)) \vee plus(s(0),0,s(0))\}.\end{aligned}$$

We notice that $\tau M(F)$ is the resolvent of two clauses in $\tau M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$. Since the target hypothesis H subsumes $\tau M(F \cup \tau Taut(\mathcal{I}_{\mathcal{H}}))$, it can be obtained by dropping operators and anti-instantiation operators.

As we have described here, there has not been any proposal to inverse resolution operators that ensure completeness for finding hypotheses so far. For this reason, it was not straightforward to formally describe the search space that hypothesis finding based on inverse entailment actually imposes. Through this chapter, we have clarified that hypothesis finding based on inverse entailment can be reduced into inverse subsumption from the minimal complement of a bridge theory F and the tautological clauses $Taut(\mathcal{I}_{\mathcal{H}})$. Suppose that every target hypothesis H wrt F and $\mathcal{I}_{\mathcal{H}}$ is irredundant. In other words, $H - \{C\}$ is no longer a hypothesis wrt F and $\mathcal{I}_{\mathcal{H}}$ for every clause $C \in H$. In this case, we do not need to apply any anti-weankening operators adding arbitrary clauses to $M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$. Every ground hypothesis can be obtained only by applying some dropping operators to $M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$. Hence, under this specific condition that the target hypotheses are ground and irredundant, the number of possible hypotheses should be at least upper bounded by $2^n - 1$ where n is the number of clauses in $M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$.

In this way, if we consider some typical assumption to the target hypotheses, it would be possible to analyze two search spaces of inverse subsumption and inverse entailment by comparing the number of possible hypotheses obtained by each approach. For this comparison, we first give the formal definition of inverse resolution operator as follows:

Definition 5.5 (Inverse resolution operator)**.** Let $\mathcal{I}_{\mathcal{H}} = \langle \mathbf{L} \rangle$ be an induction field and S a ground clausal theory belonging to $\mathcal{I}_{\mathcal{H}}$. A ground clausal theory T is *obtained by inverse entailment from S wrt $\mathcal{I}_{\mathcal{H}}$* if T is of the form:

$$T = (S - \{C\}) \cup \{(D_1 \cup \{l\}),\ (D_2 \cup \{\neg l\})\},$$

where D_1 and D_2 are two clauses such that $D_1 \cup D_2 = C$, and l and $\neg l$ are complementary literals that are not contained in C but in $\mathbf{L}$.

Example 5.8. Let S be the clausal theory $\{\{a\}\}$ and $\mathcal{I}_{\mathcal{H}}$ be $\langle \{a,\ b,\ \neg b,\ c,\ \neg c\} \rangle$. There are six possible clausal theories obtained by at one time applying an inverse resolution operator to S as follows:

$$T_1 = \{\{a,\ b\},\ \{\neg b\}\}.\ T_2 = \{\{b\},\ \{a,\ \neg b\}\}.\ T_3 = \{\{a,\ b\},\ \{a,\ \neg b\}\}.$$
$$T_4 = \{\{a,\ c\},\ \{\neg c\}\}.\ T_5 = \{\{c\},\ \{a,\ \neg c\}\}.\ T_3 = \{\{a,\ c\},\ \{a,\ \neg c\}\}.$$

Assume that S contains only one clause C that has n literals. Each literal in C can be included in either D_1, D_2 or both D_1 and D_2. That is why there are 3^n ways to apply an inverse resolution operator at one time for each complementary literal in $\mathcal{I}_{\mathcal{H}}$. Hence, the total number of possible ways to apply an inverse resolution operator at one time should be $3^n \times m$ where m is the number of pairs of complemental literals in a given induction field.

Based on this inverse resolution operator, we investigate two search spaces that inverse subsumption and inverse entailment impose.

Example 5.9. Let a bridge theory F be $\{\{a,\ b,\ c\},\ \{c,\ \neg b\}, \{\neg a\}\}$ and an induction field $\mathcal{I_H}$ be $\langle\{a,\ \neg a,\ b,\ \neg b,\ c\}\rangle$. In the following, we assume the target hypotheses as the ground hypotheses wrt $\mathcal{I_H}$ and F. $\tau M(F)$ and $\tau M(F \cup Taut(\mathcal{I_H}))$ are as follows:

$$\begin{aligned} \tau M(F) &= \{\{\neg c,\ a\}\}. \\ \tau M(F \cup Taut(\mathcal{I_H})) &= \{\{\neg c,\ a,\ b\},\ \{\neg c,\ a,\ \neg b\}\}. \end{aligned}$$

In case of inverse subsumption, since a target hypothesis H is ground, H can be obtained by applying dropping operators to $M(F \cup Taut(\mathcal{I_H}))$. If we should drop either b or $\neg b$, we obtain the single clause $\{\neg c,\ a\}$. There are $2^2 - 1$ ways to apply dropping operators to this clause. If we should drop neither b nor $\neg b$, there are possibly $2^4 - 1$ dropping operations. Totally, we can possibly consider $2^2 - 1 + 2^4 - 1 = 18$ hypotheses.

In another case of inverse entailment, we focus on a ground hypothesis H such that $H \models M(F)$. By Subsumption Theorem, there exists a derivation of a clause D from H that subsumes $M(F)$. Since D is ground, D should be either $\{\neg c\}$, $\{a\}$ or $\{\neg c,\ a\}$. For each case, we consider the number of possible hypotheses obtained by applying inverse resolution operators. For instance, there are six clausal theories obtained by at one once applying an inverse resolution operator to the clause $\{\neg c\}$ as follows:

$$\begin{aligned} &\{\{\neg c,\ \neg a\},\ \{a\}\},\ \{\{\neg c,\ a\},\ \{\neg a\}\},\ \{\{\neg c,\ a\},\ \{\neg c,\ \neg a\}\}, \\ &\quad \{\{\neg c,\ \neg b\},\ \{b\}\},\ \{\{\neg c,\ b\},\ \{\neg b\}\},\ \{\{\neg c,\ b\},\ \{\neg c,\ \neg b\}\}. \end{aligned}$$

Figure 5.8 shows the possible ways to apply inverse resolution operators. Since there are two pairs of complementary literals in $\mathcal{I_H}$, we can apply this operator at most twice. There are possibly $3 + 6 + 9 = 18$ and $7 + 270 + 42 = 319$ clausal theories obtained at one operation and second operations, respectively. Totally, we can possibly consider $1+3+18+319 = 341$ hypotheses using inverse resolution. In the case of inverse subsumption, it

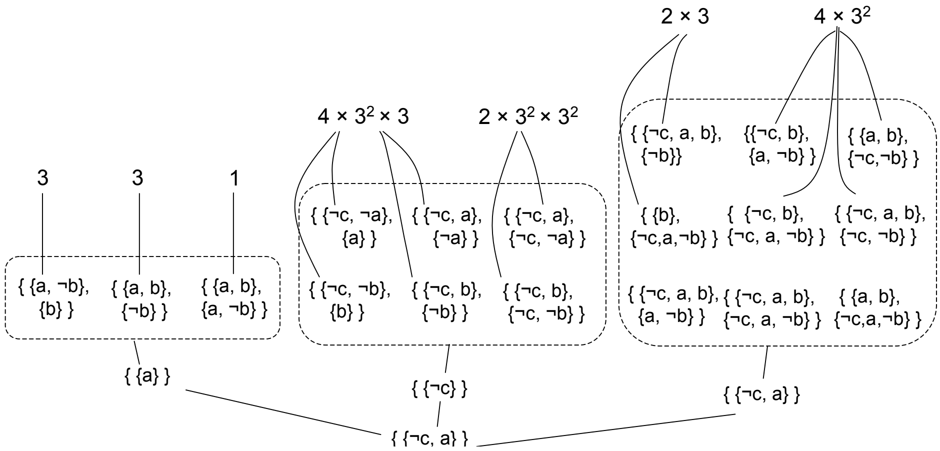


Figure 5.8: Possible Hypotheses Obtained by Inverse Entailment

is sufficient to consider only 18 hypotheses. On the other hand, inverse entailment has to deal with 341 hypotheses. This difference lies in that most of hypotheses obtained by inverse entailment are redundant. For instance, the clausal theory $\{\{\neg c, a\},\ \{\neg c,\ \neg a\}\}$ in Figure 5.8 is redundant, since it contains the original clause $\{\neg c, a\}$.

Simple applications of inverse resolution tend to produce redundant theories that is logically equivalent to the original theory. As shown in the above example, this problem can causes the difference of search space between inverse entailment and inverse subsumption.

5.5 Summary

This chapter has shown that inverse subsumption is an alternative generalization relation to ensure completeness for finding hypotheses. This result can be applied to each IE-based procedures. Generalization in Progol [44, 75], HAIL [61, 63] and Imparo [31] described in Section 3.1, are based on inverse subsumption, instead of entailment, whereas it has not been clarified so far whether or not this logical reduction makes generalization incomplete. For this open problem, we have showed that inverse subsumption can ensure the completeness only by adding tautological clauses associated with a language bias to a bridge theory. The generalization of Residue procedure [83] corresponds to inverse subsumption with residue complements, which has been studied in Section 5.3. In Section 5.4, we have discussed about the correspondence between inverting resolution and adding tautological clauses and showed that the task of V-operator, which has been previously proposed for inverse resolution, can be achieved by using inverse subsumption. We have also analyzed the search space in each case of inverse entailment and inverse subsumption with minimal complement using a concrete example. As a result, we have showed that the search space in inverse subsumption with minimal complements could be indeed less than the one in inverse entailment.

CF-induction uses inverse entailment as the generalization relation. Hence, the generalization of CF-induction can be reduced into inverse subsumption with minimal complements. In next chapter, we investigate the generalization procedure of CF-induction and logically reconstruct it into a more simplified form based so as to reduce the non-determinism in CF-induction.

Chapter 6

Logical Reconstruction in CF-induction

6.1 Introduction

CF-induction preserves soundness and completeness of finding hypotheses in full clausal theory. Compared with the other IE-based methods, CF-induction has three important benefits. Unlike Progol [44], HAIL [61] and Imparo [31], it enables the solution of more complex problems in richer knowledge representation formalisms beyond Horn logic. The motivating problem, provided in Chapter 4, is indeed described in non-Horn clausal logic. Unlike FC-HAIL [63], CF-induction is complete for finding full clausal hypotheses. In Chapter 4, using this feature, we have shown that CF-induction can integrate abduction and induction in biological inference of metabolic pathways. Unlike the residue procedure [83], CF-induction can exploit language bias to specify the search space so as to focus the procedure on some relevant part.

The derivation process in CF-induction can be described as follows: Note

$$\overline{B \wedge \neg E \models \cdots \models CC} \searrow \xrightarrow{\text{(Generalization operations)}}$$
$$\neg CC \;=\!\!|\; U_1 \;=\!\!|\; \cdots \;=\!\!|\; U_n \;=\!\!|\; H$$

Figure 6.1: Hypothesis Finding in CF-induction

that CC is a bridge theory of CF-induction. After constructing a bridge theory CC, its negation translated into CNF is generalized using a variety of generalizers, as shown in Chapter 3. Any combination of generalizers can be also soundly applied as another generalizer. Indeed, a particular hypothesis H requires several different generalizers in some specific order like the generalization operations in Figure 6.1. This fact that these generalizers can be sequenced in many different ways makes generalization of CF-induction highly non-deterministic.

Our objective is to reduce the non-determinism of CF-induction in order to make the procedure easier to apply in real-world applications. In this chapter, we concentrate on simplifying the process of obtaining hypotheses from a bridge theory. In particular, we propose two approaches for simplification of CF-induction. The first approach is based on the observation that the negated hypothesis $\neg H$ can be computed deductively from CC, i.e.,

$$CC \models V_1 \models V_2 \models \cdots \models V_{m-1} \models V_m \models \neg H. \tag{6.1}$$

In the interests of computational efficiency, it is convenient to work in clausal form logic. For instance, $\neg H$ can be represented as the minimal complement of H_{sk} in which any existentially quantified variables in $\neg H$ are substituted by ground Skolem constants. Therefore, the first approach works in three steps: first G is deduced from CC; then it is negated; and finally anti-instantiation is used to result in the final hypothesis H.

To derive G from CC, we introduce a new deductive operator, which can be regarded as simplifying the existing operators of subsumption, resolution and weakening that we have introduced in Chapter 5. This new operator, called the *γ-operator*, warrants the insertion of literals into the clauses of CC. We show that this single deductive operator (followed by negation and anti-instantiation) is sufficient to preserve the soundness and completeness of CF-induction. This new approach, called *CF-induction with γ-operator*, simplifies the original CF-induction in the sense that it only requires one deductive operator (γ-operator) and one generalization operator (anti-instantiation) to be used in the construction of H from CC.

The second approach is based on inverse subsumption with minimal subsumption. Using the result in Chapter 5, any hypothesis can be inductively derived from the minimal complement of $CC \cup Taut$, i.e.,

$$\tau M(CC) \preceq R_1 \preceq R_2 \preceq \cdots \preceq R_{k-1} \preceq R_k \preceq \mu H. \tag{6.2}$$

To use Formula 6.2, we need to logically connect production fields with induction fields that will be extended so as to describe the syntax of target hypotheses.

In this chapter, we also investigate another non-deterministic procedure: construction of bridge theories. Since a bridge theory in CF-induction is defined as a subset of the characteristic clauses, the number of choice points can exponentially increase in accordance with the number of characteristic clauses. This combinatorial explosion makes the construction of bridge theories highly non-deterministic. For this problem, we propose a deterministic way, while it preserves completeness for finding hypotheses in CF-induction.

The rest of this chapter is organized as follows. Section 6.2 describes the first approach based on deductive operations. Section 6.3 describes the second approach based on inverse subsumption. In Section 6.3, we also show a deterministic way to construct bridge theories, and Section 6.4 compares two approaches using several examples. Section 6.5 concludes.

6.2 CF-induction with Deductive Operations

Our motivation in this section is to develop a simplified generalization procedure for CF-induction that uses fewer operators while preserving its soundness and completeness for finding hypotheses. We present one way to simplify the generalization process by computing generalizations deductively. Our approach is motivated by recalling the following deductive operators, which have been introduced in Chapter 5.

1. (resolution) $T = S \cup \{C\}$, where C is a resolvent of clauses $D_1, D_2 \in S$.
2. (subsumption) $T = S \cup \{C\}$, where C is subsumed by some clause $D \in S$.
3. (weakening) $T = S - \{D\}$ for some clause $D \in S$.

Two special cases of the subsumption operator can be further distinguished by the following two operators.

2a (instantiation) $T = S \cup \{D\sigma\}$ for some clause $D \in S$ and substitution σ.

2b (expansion) $T = S \cup \{C\}$, where C is a superset of some clause $D \in S$.

We write $S \vdash_r T$, $S \vdash_s T$, $S \vdash_w T$, $S \vdash_\alpha T$ and $S \vdash_\beta T$ to denote that T is directly derivable from S by resolution, subsumption, weakening, instantiation and expansion, respectively. $\vdash_X^*$ is the reflexive and transitive closure of $\vdash_X$, where X is one of the symbols r, s, w, α, β.

We mention the following two properties concerned with reordering of deductive operators.

Proposition 6.1 ([92])**.** Let U_1 and U_2 be clausal theories. If $U_1 \vdash_s^* U_2$, then there exist two clausal theories V_1 and V_2 such that $U_1 \vdash_\alpha^* V_1 \vdash_\beta^* V_2 \vdash_w^* U_2$.

Proof. By Definition 5.1, there exist a substitution σ_i and a clause $D_i \in U_1$ such that $D_i\sigma_i \subseteq C_i$, for each clause $C_i \in U_2 - U_1$ $(1 \leq i \leq n)$. Let T be a finite set $\bigcup_{i=1}^{n}\{D_i\sigma_i\}$. Let V_1 be $U_1 \cup T$ and V_2 be $V_1 \cup \{C_1, \ldots, C_n\}$. Then it holds that $U_1 \vdash_\alpha^* V_1$ and $V_1 \vdash_\beta^* V_2$. Moreover, since it holds that $U_2 \subseteq V_2$, it holds that $V_2 \vdash_w^* U_2$. $\square$

Proposition 6.2 ([92])**.** Let S, U and V be clausal theories. If $S \vdash_w^* U \vdash_s^* V$, then there exists a clausal theory U' such that $S \vdash_s^* U' \vdash_w^* V$.

Proof. Let U' be the clausal theory $S \cup (V - U)$. Since $U \vdash_s^* V$, there exists a clause $D_i \in U$ such that $D_i \succeq C_i$ for each clause $C_i \in V - U$. Since $S \vdash_w^* U$, $U \subseteq S$ holds. Since $D_i \in U$, $D_i \in S$ holds. Therefore U' is obtained from S using the subsumption operator, that is, $S \vdash_s^* U'$. Next we show that $V \subseteq U'$ holds. Since $U' = S \cup (V - U)$, $V - U \subseteq U'$ holds. And also, $S \subseteq U'$ holds. Since $S \vdash_w^* U$, $U \subseteq S$ holds. Accordingly, $V \cap U \subseteq U'$ holds, since $V \cap U \subseteq U$. Since $V - U \subseteq U'$ and $V \cap U \subseteq U'$, $V \subseteq U'$ holds. Hence $U' \vdash_w^* V$ holds. $\square$

We use the above ordering results as well as Theorem 5.5 to show how the number of generalization operators used in CF-induction can be reduced. Section 6.2.1 shows this result in the case of ground hypotheses and Section 6.2.2 shows it in the general case.

6.2.1 Logical Relation Between Bridge Theories and Ground Hypotheses

First, we show that resolution and instantiation can be incorporated into the selection of CC. We show this with the following two lemmas.

Lemma 6.1 ([92])**.** Let CC be a bridge theory wrt B, E and $\mathcal{P}$, and U be a clausal theory. If $CC \vdash_r^* U$, then there exist a clausal theory V and a bridge theory CC' wrt B, E and $\mathcal{P}$ such that $CC' \vdash_s^* V \vdash_w^* U$.

Proof. The proof is in two parts.

(a) First we prove that, for each clause $C_i \in U$ $(1 \leq i \leq n)$, there exists a clause $D_i \in Carc(B \wedge \overline{E}, \mathcal{P})$ such that $D_i \succeq C_i$.

By the definition of characteristic clauses, for each clause $K \in Th_{\mathcal{P}}(B \wedge \overline{E})$ there exists a clause $M \in Carc(B \wedge \overline{E}, \mathcal{P})$ such that $M \succeq K$. Hence it is sufficient to show that every clause $C_i \in U$ is included in $Th_{\mathcal{P}}(B \wedge \overline{E})$. Since $CC \models U$ and $B \wedge \overline{E} \models CC$, it holds that every clause $C_i \in U$ is a consequence of $B \wedge \overline{E}$. Then it remains to show that every clause $C_i \in U$ belongs to $\mathcal{P}$, which is done by mathematical induction on the number n of the applications of $\vdash_r$ for deriving U from CC. In the following, we write $CC \vdash_r^n U$ to denote that U is derived from CC by n applications of $\vdash_r$.

Base step: If $n = 0$ then $U = CC$ and it trivially follows that every clause in U belongs to $\mathcal{P}$.

Induction step: If $n = k+1$ for some $k \geq 0$, then it holds that $CC \vdash_r^k U' \vdash_r U$ where U' is a clausal theory. By the induction hypothesis, it holds that every clause in U' belongs to $\mathcal{P}$. Moreover, it follows that $U = U' \cup \{R\}$ for some resolvent R of two clauses in U'. Since every clause in U' belongs to $\mathcal{P}$ and $\mathcal{P}$ is closed under instantiation, the resolvent R also belongs to $\mathcal{P}$. Thus every clause in U belongs to $\mathcal{P}$ and so part (a) holds.

(b) Now we show how to construct the theories CC' and V. Start by defining the theory $T = \bigcup_{i=1}^{n}\{D_i\}$ using the clauses $D_i \in Carc(B \wedge \overline{E}, \mathcal{P})$ constructed above. Now define the bridge theory $CC' = CC \cup T$ and the theory $V = CC' \cup U$. Since for each clause $C_i \in U$ there exists $D_i \in T$ such that $D_i \succeq C_i$, $CC' \vdash_s^* V$ holds. Since $U \subseteq V$, $V \vdash_w^* U$ holds. Hence $CC' \vdash_s^* V \vdash_w^* U$ holds. Each clause in CC' is an instance of a clause in $Carc(B \wedge \overline{E}, \mathcal{P})$. Since $CC \subseteq CC'$ and CC is a bridge theory, there exists a clause in CC' which is an instance of a clause in $NewCarc(B, \overline{E}, \mathcal{P})$. Then CC' is a bridge theory. Therefore there exist a bridge theory CC' and a clausal theory V such that $CC' \vdash_s^* V \vdash_w^* U$. □

Lemma 6.2 ([92])**.** Let CC be a bridge theory wrt B, E and $\mathcal{P}$, and U be a clausal theory. If $CC \vdash_\alpha^* U$, then U is a bridge theory wrt B, E and $\mathcal{P}$.

Proof. Every clause in U is an instance of a clause in $Carc(B \wedge \overline{E}, \mathcal{P})$. Since $U \supseteq CC$, there exists a clause $C_i \in U$ such that C_i is an instance of a clause from $NewCarc(B, \overline{E}, \mathcal{P})$. Therefore, U is a bridge theory. □

Then, using Lemmas 6.1 and 6.2, we can show the following theorem, which establishes the logical relation between bridge theories and ground hypotheses.

Theorem 6.1 ([92])**.** Let H be a ground hypothesis wrt B, E and $\mathcal{P}$. Then there exist a bridge formula CC wrt B, E and $\mathcal{P}$ and a clausal theory V such that $CC \vdash_\beta^* V \vdash_w^* \overline{H}$.

Proof. First, we consider the case that $\overline{H}$ has no tautological clauses. Then, by Theorem 3.1, there exists a bridge theory CC such that $CC \models \neg H$. Since H is ground, $\overline{H} \equiv \neg H$ holds. Thus $CC \models \overline{H}$ holds. By Theorem 5.5, there exist clausal theories V_1 and V_2 such that $CC \vdash_r^* V_1 \vdash_s^* V_2 \vdash_w^* \overline{H}$. By Lemma 6.1, there exist a clausal theory V_3 and a bridge formula CC'' such that $CC'' \vdash_s^* V_3 \vdash_w^* V_1 \vdash_s^* V_2 \vdash_w^* \overline{H}$. By Proposition 6.2, there exists a clausal theory V_4 such that $CC'' \vdash_s^* V_3 \vdash_s^* V_4 \vdash_w^* V_2 \vdash_w^* \overline{H}$. Thus $CC'' \vdash_s^* V_4 \vdash_w^* \overline{H}$. By Proposition 6.1, there exist clausal theories V_5 and V_6 such that $CC'' \vdash_\alpha^* V_5 \vdash_\beta^* V_6 \vdash_w^* V_4 \vdash_w^* \overline{H}$. By Lemma 6.2, V_5 is a bridge theory, and by letting $CC' = V_5$, it holds that $CC' \vdash_\beta^* V_6 \vdash_w^* \overline{H}$.
Next, in the case that $\overline{H}$ contains tautological clauses, we let $T = \{D_1, \ldots, D_n\}$ denote the set of the tautological clauses in $\overline{H}$. Then, since every clause $D_i \in T$ $(1 \leq i \leq n)$ is a consequence of $B \wedge \overline{E}$ and D_i belongs to $\mathcal{P}$, for

each $D_i \in T$ there exists a clause $C_i \in Carc(B \wedge \overline{E}, \mathcal{P})$ such that $C_i \succeq D_i$. Let S be the clausal theory $\bigcup_{i=1}^{n}\{C_i\}$. Then, it holds that $S \vdash_s^* S_1 \vdash_w^* T$ where $S_1 = S \cup T$. Now, since the clausal theory $\overline{H} - T$ has no tautological clauses, there exist a bridge theory CC and a clausal theory V_7 such that $CC \vdash_s^* V_7 \vdash_w^* \overline{H} - T$. Then, it holds that $CC \cup S \vdash_s^* V_7 \cup S_1 \vdash_w^* \overline{H}$. Since the clausal theory $CC \cup S$ satisfies Definition 3.4, the theorem holds. □

Next, we introduce a new operator, which can be regarded as concatenating weakening and expansion.

Definition 6.1. [γ-operator] Let S and T be clausal theories. T is *directly γ-derivable* from S iff T is obtained from S under the following condition:

$$T = (S - \{D\}) \cup \{C_1, \ldots, C_n\}$$
$$\text{for some } n \geq 0 \text{ where } C_i \supseteq D \text{ for all } 1 \leq i \leq n.$$

Analogously to Definition 5.1, we write $S \vdash_\gamma T$ iff T is directly γ-derivable from S and $\vdash_\gamma^*$ is a reflexive and transitive closure of $\vdash_\gamma$.

Theorem 6.2 ([92])**.** Let H be a ground hypothesis wrt B, E and $\mathcal{P}$. Then, there exists a bridge formula CC wrt B, E and $\mathcal{P}$ such that $CC \vdash_\gamma^* \overline{H}$.

Proof. Since H is a ground hypothesis, by Theorem 6.1, there exist a bridge theory $CC = \{C_1, \ldots, C_n\}$ and a clausal theory U such that $CC \vdash_\beta^* U \vdash_w^* \overline{H}$. Let F_{C_i} be the clausal theory $\{C \mid C \in \overline{H} \textit{ and } C_i \subseteq C\}$, for each clause $C_i \in CC$ $(1 \leq i \leq n)$. Then, by Definition 6.1, for each clause $C_i \in CC$, $\{C_i\} \vdash_\gamma^* F_{C_i}$ holds. Accordingly, $CC \vdash_\gamma^* \bigcup_{i=1}^{n} F_{C_i}$ holds. Hence, it is sufficient to show that $\overline{H} = \bigcup_{i=1}^{n} F_{C_i}$. Since $F_{C_i} \subseteq \overline{H}$ for every clause $C_i \in CC$, it holds that $\bigcup_{i=1}^{n} F_{C_i} \subseteq \overline{H}$. Conversely, since $CC \vdash_\beta^* U$, for every clause $D \in U$, there exists a clause $C_i \in CC$ such that $C_i \subseteq D$. Also since $U \vdash_w^* \overline{H}$, it holds that $\overline{H} \subseteq U$. Then, for every clause $D \in \overline{H}$, there exists a clause $C_i \in CC$ such that $C_i \subseteq D$, that is, $D \in F_{C_i}$. This means that $\overline{H} \subseteq \bigcup_{i=1}^{n} F_{C_i}$. Hence, it holds that $\overline{H} = \bigcup_{i=1}^{n} F_{C_i}$. Therefore, it holds that $CC \vdash_\gamma^* \overline{H}$. □

6.2.2 Deriving Non-Ground Hypotheses

We generalize the result of the previous section to non-ground hypotheses. We show that any hypothesis can be obtained from a bridge theory by applying the γ-operator followed by anti-instantiation.

Definition 6.2. Let B and E be clausal theories and $\mathcal{P} = \langle \mathbf{L} \rangle$ be a production field. A clausal theory H is derived by *CF-induction with γ-operator* from B, E and $\mathcal{P}$ iff H is constructed as follows:

Input: A background theory B, observations E
and a production field $\mathcal{P}$.
Output: A hypothesis H wrt B, E and $\mathcal{P}$.

Step 1. Construct a bridge theory CC wrt B, E and $\mathcal{P}$.

Step 2. Construct a clausal theory G such that $CC \vdash_{\gamma}^{*} G$.

Step 3. Compute the complement $\overline{G}$ of G.

Step 4. H is obtained by applying anti-instantiation to $\overline{G}$,
such that (1) $B \wedge H$ is consistent,
(2) H contains no Skolem constants, and
(3) for every literal L in H, $\neg L$ belongs to $\mathcal{P}$.

Several remarks are necessary for Definition 6.2.

Step 2. Even if G satisfies $CC \vdash_{\gamma}^{*} G$ for some bridge theory CC, any output H obtained from G cannot satisfy the conditions of Definition 3.1 unless $\overline{G} \wedge B$ is consistent and G belongs to $\mathcal{P}$. In this respect, the constraints of H at Step 4 are introduced to guarantee the soundness of H.

Step 3. The complement of G can theoretically include redundant clauses such as tautological clauses and clauses properly subsumed by other clauses. Accordingly it might be sufficient to use minimal and residue complements, instead of complements.

Step 4. Anti-instantiation allows us to replace subterms in $\overline{G}$ with variables. For example, for the clause $p(a) \vee q(a)$, it is possible to construct $p(X) \vee$

$q(Y)$ obtained by replacing the constant a in $p(a)$ and $q(a)$ with two variables X and Y, respectively. In this way there are many possibilities to apply anti-instantiation for clauses.

We now give soundness and completeness results for CF-induction with γ-operator.

Theorem 6.3 ([92])**.** [Soundness] Let B, E and H be clausal theories, and $\mathcal{P}$ be a production field. If H is derived by CF-induction with γ-operator from B, E and $\mathcal{P}$, then H is a hypothesis wrt B, E and $\mathcal{P}$.

Proof. Suppose a clausal theory H is derived by CF-induction with γ-operator from B, E and $\mathcal{P}$. Then, by Definition 6.2, there exist a bridge theory CC and a clausal theory G such that H is derived by applying anti-instantiation to $\overline{G}$ and $CC \vdash_{\gamma}^{*} G$. By Definition 3.4, it holds that $B \wedge \overline{E} \models CC$. By Definition 6.1, it holds that $CC \models G$. Accordingly, it holds that $B \wedge \overline{E} \models G$. Since H is derived by applying anti-instantiation to $\overline{G}$, $H \models \overline{G}$ holds. Since $\overline{G} \models \neg G$, $H \models \neg G$ follows. Equivalently, $G \models \neg H$. Therefore, it holds that $B \wedge \overline{E} \models \neg H$. Then, it holds that $B \wedge H \models \neg\overline{E}$. Since, from Step 4 of Definition 6.2, H contains no Skolem constants from $\overline{E}$, it holds that $B \wedge H \models E$. Hence, it holds that H is a hypothesis wrt B, E and $\mathcal{P}$, since, from Step 4 of Definition 6.2, $B \wedge H$ is consistent and for every literal L appearing in H, $\neg L \in \mathbf{L}$. □

Theorem 6.4 ([92])**.** [Completeness] Let B, E and H be clausal theories, and $\mathcal{P}$ be a production field. If H is a hypothesis wrt B, E and $\mathcal{P}$, then there exists a theory $H^* \equiv H$ that is derived by CF-induction with γ-operator from B, E and $\mathcal{P}$.

Proof. Suppose H is a hypothesis wrt B, E and $\mathcal{P}$. By Theorem 3.1, there is a bridge theory CC wrt B, E and $\mathcal{P}$ such that $CC \cup H$ is unsatisfiable. Using Herbrand's theorem, there are two finite sets CC' and H' such that CC' (resp. H') is a finite set of ground instances of CC (resp. H) and $CC' \cup H'$ is unsatisfiable. In this case, H' can be chosen in such a way that for every clause C in H, there is an instance C' of C such that $C' \in H'$, and

also, CC' can be chosen in such a way that CC' contains at least one instance of a clause in $NewCarc(B, \overline{E}, \mathcal{P})$. Then, H can be obtained by applying an anti-instantiation generalizer to H'. We prove that H' is a ground hypothesis wrt B, $\overline{\overline{E}}$, and $\mathcal{P}$. That is, we will show that (1) $B \wedge H' \models \overline{\overline{E}}$, (2) $B \wedge H' \not\models \Box$ and (3) $\neg L \in \mathbf{L}$ for every literal L appearing in H'.
Proof of (1): CC is a bridge theory wrt B, E and $\mathcal{P}$. Since every clause in CC' is an instance of a clause in CC, CC' satisfies the first condition of Definition 3.4. Also, CC' contains at least one clause C' such that C' is an instance of a clause from $NewCarc(B, \overline{E}, \mathcal{P})$. Then CC' satisfies the second condition of Definition 3.4. Hence, CC' is also bridge theory wrt B, E and $\mathcal{P}$. Thus $B \wedge \overline{E} \models CC'$ holds. By $CC' \models \neg H'$, $B \wedge \overline{E} \models \neg H'$ holds. Then $B \wedge H' \models \neg\overline{E}$ holds. By $\neg\overline{E} \equiv \overline{\overline{E}}$, $B \wedge H \models \overline{\overline{E}}$ holds.
Proof of (2): If it holds that $B \wedge H' \models \Box$, then it must hold that $B \wedge H \models \Box$, by $B \wedge H \models B \wedge H'$. It contradicts the fact H is a hypothesis.
Proof of (3): Since H is a hypothesis wrt B, E and $\mathcal{P}$, for every literal L appearing in H, $\neg L \in \mathbf{L}$ holds. Then it holds that $\neg L \in \mathbf{L}$ for every literal L appearing in H', since $\mathcal{P} = \langle \mathbf{L} \rangle$ is closed under instantiation.

Now, since H' is a ground hypothesis wrt B, $\overline{\overline{E}}$ and $\mathcal{P}$, there exists a bridge theory CC'' wrt B, $\overline{\overline{E}}$ and $\mathcal{P}$ such that $CC'' \vdash_{\gamma}^{*} \overline{H'}$ by Theorem 6.2. Since $\overline{\overline{\overline{E}}} \equiv \overline{E}$, it holds that for every clause C in $Carc(B \wedge \overline{\overline{\overline{E}}}, \mathcal{P})$, C is contained in $Carc(B \wedge \overline{E}, \mathcal{P})$. Then, it also holds that for every clause C in CC'', C is contained in $Carc(B \wedge \overline{E}, \mathcal{P})$. Hence, CC'' is also bridge theory wrt B, E and $\mathcal{P}$. Then, from Step 1 of Definition 6.2, CC'' can be constructed in a CF-induction with γ-operator from B, E and $\mathcal{P}$. Moreover, $\overline{H'}$ can be also constructed from CC'' at Step 2. Since H' is ground, it holds that H' is logically equivalent to $\overline{\overline{H'}}$ computed from $\overline{H'}$ at Step 3. Recall that H is obtained by applying anti-instantiation to H'. Therefore a formula H^* is obtained at Step 4 with the application of anti-instantiation to $\overline{\overline{H'}}$ such that $H^* \equiv H$. $\Box$

Example 6.1. Recall Example 3.2. Let CC be the following bridge theory, which appears in Example 3.2.

$$CC = (natural(0) \vee even(0)) \wedge \neg natural(s(0)).$$

Assume that a γ-operator is applied to CC so that $\neg natural(s(0))$ is replaced with the two clauses $\neg natural(s(0)) \vee \neg even(0)$ and $\neg natural(s(0)) \vee natural(0)$, then the following clausal theory G_1 is constructed:

$$\begin{aligned} G_1 &= (natural(0) \vee even(0)) \\ &\wedge (\neg natural(s(0)) \vee \neg even(0)) \\ &\wedge (\neg natural(s(0)) \vee natural(0)). \end{aligned}$$

Then, we can obtain the complement $\overline{G_1}$ of G_1, which is logically equivalent to F'_1 in Example 3.2. Next assume that another γ-operator is applied to CC so that $\neg natural(s(0))$ is replaced with the two clauses $\neg natural(s(0)) \vee even(0)$ and $\neg natural(s(0)) \vee \neg natural(0)$, then the following clausal theory G_2 is constructed:

$$\begin{aligned} G_2 &= (natural(0) \vee even(0)) \\ &\wedge (\neg natural(s(0)) \vee even(0)) \\ &\wedge (\neg natural(s(0)) \vee \neg natural(0)) \end{aligned}$$

Then, the complement $\overline{G_2}$ of G_2 is logically equivalent to F'_2 in Example 3.2. Accordingly, we can obtain a clausal theory, which is logically equivalent to F'_3 in Example 3.2 by applying an anti-instantiation generalizer to $\overline{G_2}$. In this way, the inverse resolution generalizer can be realized with applications of the γ-operator.

6.2.3 Related Work

The γ-operator can be regarded as a particular *downward refinement operator* [3, 34, 41, 52] for the $\vdash^*_\gamma$ order, which is closely related to the subsumption order. Let S and T be clausal theories such that $S \vdash^*_\gamma T$. Then $S \succeq T$ holds. Compared with the subsumption order, one important feature of γ-operator lies in restraint of the operation of instantiation, which leads to a large number of choice points. There are certain desirable properties that a "good" downward refinement operator should satisfy and we intend to study which of these the γ-operator satisfies.

We can reduce generalization under the entailment relation in the previous version to generalization under the γ-operator. It is based on the notion that any series of processes of inductive operations on the inverse relation of entailment between the negation of a bridge theory and a hypothesis connects a certain series of processes of deductive operations on entailment between a bridge theory and the negation of hypothesis. Accordingly, there are two sides where we can grasp generalization processes. Yamamoto and Fronhöefer [84] and Yamamoto [83] first have studied the connection between two clausal theories related by entailment and negation. It will be interesting to consider about the relation between two clausal theories ordered by the γ-operator instead of entailment and these negations.

6.3 CF-induction with Inverse Subsumption

This section investigates how CF-induction can compute target hypotheses based on inverse subsumption with minimal complements. Inverse subsumption ensures the completeness of finding hypotheses with respect to a given induction field. An induction field is used to describe possible literals appeared in ground hypotheses. In contrast, a production field used in CF-induction describes possible literals appeared in the *negations* of hypotheses. We thus need to connect induction fields with production fields.

6.3.1 Logical Relation between Production Fields and Induction Fields

We first extend the definition of induction fields, which has been defined in Section 5.3.3, into a richer representation formalization so as to describe not only literals to be possibly included, but also the syntactical structure of ground hypotheses like the number and length of clauses.

Definition 6.3. [Extended induction field]An *extended* induction field $\mathcal{I}_{\mathcal{H}}$ is a triple $\langle \mathbf{L}, num, len \rangle$ where $\mathbf{L}$ is the set of literals to be appeared in ground hypotheses, and num and len are the maximum number and length of clauses in ground hypotheses, respectively. A ground hypothesis H_g *belongs* *to* $\mathcal{I_H} = \langle \mathbf{L}, num, len \rangle$ iff every literal in H_g is included in $\mathbf{L}$, the number of clauses in H_g is less than or equal to num and for every clause $C \in H_g$, the length of C is less than or equal to len. In the following, we simply refer to an induction field, instead of an extended induction field, if no confusion arises.

Note that the induction field is regarded as an extension of the previous language bias obtained by adding the information on the number and length. Based on this language bias, we define target hypotheses as follows.

Definition 6.4. [Hypothesis wrt B, E and $\mathcal{I_H}$] Let B, E and $\mathcal{I_H}$ be a background theory, observations and an induction field, respectively. A clausal theory H is a *hypothesis* wrt B, E and $\mathcal{I_H}$ iff there is a ground hypothesis H_g wrt B and E such that H_g consists of ground instances from H and H_g belongs to $\mathcal{I_H}$.

We then reconstruct the conditions of bridge theories by reflecting the induction field as follows:

Definition 6.5. [Bridge theory wrt B, E and $\mathcal{I_H}$] Let B, E and $\mathcal{I_H} = \langle \mathbf{L}, num, len \rangle$ be a background theory, observations and an induction field. A clausal theory CC is a *bridge theory wrt* B, E *and* $\mathcal{I_H}$ iff CC is a bridge theory wrt B, E and the production field $\mathcal{P_I} = \langle \overline{\mathbf{L}},\ max\_length \leq num \rangle$ such that CC contains at most len^{num} clauses.

Note that $\overline{\mathbf{L}}$ is the set of negations of literals in $\mathbf{L}$. If no confusion arises, a "bridge theory wrt B, E and $\mathcal{I_H}$" will simply be called a "bridge theory".

Theorem 6.5 ([89])**.** Let B, E and $\mathcal{I_H}$ be a background theory, observations and an induction field. Then, for any hypothesis H wrt B, E and $\mathcal{I_H}$, there exists a bridge theory CC wrt B, E and $\mathcal{I_H}$ such that $H \models \neg CC$.

Proof. We write $\mathcal{I_H} = \langle \mathbf{L},\ num,\ len \rangle$. Since H is a hypothesis wrt B, E and $\mathcal{I_H}$, there is a ground hypothesis H_g wrt B and E such that $H \succeq H_g$ and H_g belongs to $\mathcal{I_H}$. Assume the production field $\mathcal{P}_1 = \langle \overline{\mathbf{L}} \rangle$. Then, H_g belongs to $\mathcal{P}_1$. Hence, there is a bridge theory CC wrt B, E and $\mathcal{P}_1$ such that $CC \models \neg H_g$. Since H_g is ground, $\neg H_g \equiv \overline{H_g}$ holds. Since the maximum number and length of clauses in H_g are num and len, respectively, it holds that every clause in $\overline{H_g}$ is less that or equal to num and $\overline{H_g}$ contains at most len^{num} clauses. Since $B \wedge \neg E_{sk} \models CC$ and $CC \models \overline{H_g}$, $B \wedge \neg E_{sk} \models \overline{H_g}$ holds. Hence, for each clause C in $\overline{H_g}$, there is a clause D in $Carc(B \wedge \overline{E_{sk}}, \mathcal{P}_1)$ such that $D \succeq C$. Since the length of the clause D is less than or equal to num, D belongs to the production field $\mathcal{P}_{\mathcal{I}}$. $\Box$

6.3.2 Deriving Hypotheses with Induction Fields

In the original CF-induction, the bridge theory was selected by hand. Assume that we have n characteristic clauses of $B \wedge \neg E$. The number of possible bridge theories is briefly 2^n. It is difficult to sufficiently choose one from a large number of possible theories. Here, we show an incremental way for constructing bridge theories that is deterministic and also preserving the soundness and completeness in CF-induction. First, we recall Theorem 5.4. Let S be a ground clausal theory and C a clause in S. By Theorem 5.4, $\tau M(S - \{C\}) \succeq \tau M(S)$ holds.

Let B, E and $\mathcal{I}_{\mathcal{H}}$ be a background theory, observations and an induction field. Let the list L be $\langle C_n^1, C_n^2, \ldots, C_n^m, C_c^1, C_c^2, \ldots, C_c^r \rangle$ where C_n^i $(1 \leq i \leq m)$ and C_c^j $(1 \leq j \leq r)$ is an instance of a clause in $Carc(B \wedge \neg E, \mathcal{P}_{\mathcal{I}})$ and $NewCarc(B, \neg E, \mathcal{P}_{\mathcal{I}})$, respectively. Note that since the set $\overline{\mathbf{L}}$ of $\mathcal{P}_{\mathcal{I}} = \langle \overline{\mathbf{L}}, max\_length \leq num \rangle$ contains a finite number of ground literals, the number of (new) characteristic clauses should be also finite.

Assume the bridge theory CC_k consisting of the clauses from C_n^1 to C_x^k in the list L where if $k \leq m$, the symbol x is equal to the symbol n, otherwise, x is equal to the symbol c. Then, $CC_k \supset CC_{k-1}$ holds. By Theorem 5.4,

$$\tau M(CC_{k-1}) \succeq \tau M(CC_k) \text{ holds.}$$

This fact implies that the residue and minimal complement of the previous CC_{k-1} can be obtained by residue and minimal complement of the current CC_k using inverse subsumption. Hence, every hypothesis obtained from CC_{k-1} can be also obtained from CC_k. In other words, given a sufficient number k, any hypothesis can be obtained by inverse subsumption from the residue and minimal complement of $\tau M(CC_k)$.

Based on this notation, we can consider an alternative hypothesis finding procedure of CF-induction that computes hypotheses wrt B, E and $\mathcal{I}_{\mathcal{H}}$ using inverse subsumption with minimal complements as follows:

Definition 6.6. Let B and E be clausal theories and $\mathcal{I}_{\mathcal{H}}$ be an extended induction field. A clausal theory H is derived by *CF-induction with inverse subsumption* from B, E and $\mathcal{I}_{\mathcal{H}}$ iff H is constructed as follows:

Input: A background theory B, observations E
and an extended induction field $\mathcal{I}_{\mathcal{H}} = \langle \mathbf{L},\ num,\ len \rangle$.
Output: A hypothesis H wrt B, E and $\mathcal{I}_{\mathcal{H}}$.

Step 1. Translate $\mathcal{I}_{\mathcal{H}}$ into $\mathcal{P}_{\mathcal{I}} = \langle \overline{\mathbf{L}},\ max\_length \leq num \rangle$.

Step 2. Compute $Carc(B \wedge \overline{E_{sk}}, \mathcal{P}_{\mathcal{I}}) = \langle C_1,\ C_2,\ \ldots,\ C_n \rangle$.

Step 3. $i := 0$ and $CC_0 := \emptyset$.

Step 4. $CC_{i+1} := CC_i \cup \{C_{i+1}\}$.

Step 5. Compute $\tau M(CC_{i+1})$.

Step 6. If necessary, return a clausal theory H such that

(1) H belongs to $\mathcal{I}_{\mathcal{H}}$,

(2) $H \succeq \tau M(CC_{i+1})$, and

(3) $B \wedge H$ is consistent.

Else, $i := i + 1$ and go to Step 4.

6.4 Comparison

In Section 6.2 and 6.3, we have studied two approaches for deriving a hypothesis H from a bridge theory CC in CF-induction. In the first approach based on deductive operations, we first search the complement $\overline{H}$ such that $CC \vdash_{\gamma}^{*} \overline{H}$, and then compute the original H by translating $\neg\overline{H}$ into a CNF formula. It would be sufficient for this dualization task to use minimal complements, instead of complements, since minimal complements enables to obtain the original hypothesis (that is, $M(M(H)) = \mu H$) by Theorem 5.3.

In the second approach based on inverse subsumption, we directly search a hypothesis H such that $H \succeq \tau M(CC)$. Then, a question concerning the difference between two approached would naturally occur. Which approach should we use to find hypotheses?

Both approaches are based on the (theory) subsumption relation. For clausal theories S and T, S subsumes T iff there is a clause C in S such that $C \succeq D$ for every clause D in T. Note that there is not necessarily a clause D in T such that $C \succeq D$ for every clause C in S. In other words, if S subsumes T, then $S \cup U$ also subsumes T for any clausal theory U. Then, the first and second approaches have to consider weakening and anti-weakening operations, respectively, in order to ensure the completeness for finding hypotheses. This fact makes both operations with the subsumption relation non-deterministic.

The difference lies in their target theories to be searched: On the one hand, the first approach focuses on the minimal complement of a hypothesis $M(H)$. On the other hand, the second approach focuses on a hypothesis H itself. Figure 6.2 describes their search strategies:

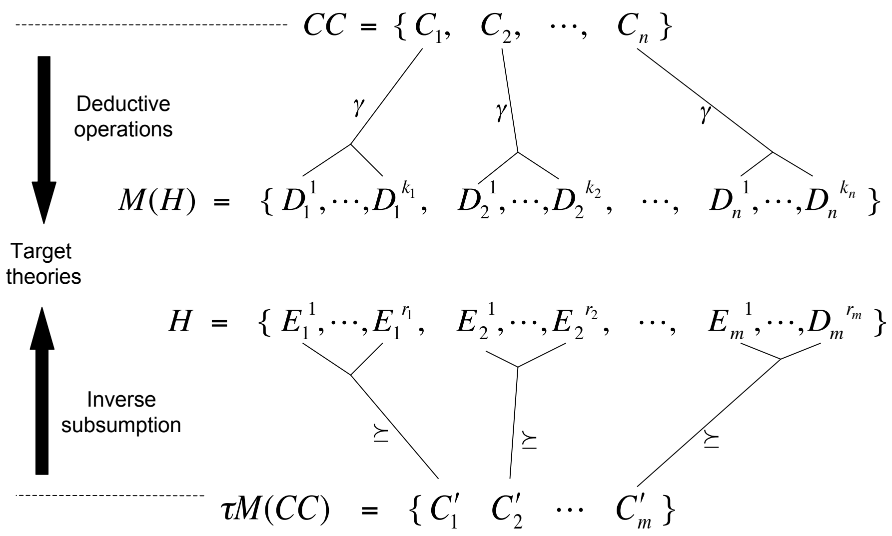


Figure 6.2: Search Strategies in Two Approaches

Figure 6.2 may give an insight to the above question: For a target hypothesis H, If $M(H)$ should be *close* to CC, it is sufficient to use the first approach. Else if H should be *close* to $\tau M(CC)$, then the second approach is sufficient. We give the following example to describe this notation.

Example 6.2. Let a background theory B and observations E be as follows:

$$B = \{a,\ b \supset c\}.\ E = \{d\}.$$

Let CC be a bridge theory $\{a,\ \neg b \vee c,\ \neg d\}$. We first assume a target hypothesis $H_1 = \{d \vee b,\ \neg a \vee \neg c\}$. $\tau M(CC)$ and $M(H_1)$ are as follows:

$$\tau M(CC) = \{\neg a \vee b \vee d,\ \neg a \vee \neg c \vee d\}.$$
$$M(H_1) = \{\neg d \vee a,\ \neg d \vee c,\ \neg b \vee a,\ \neg b \vee c\}.$$

Note that both $CC \vdash_{\gamma}^{*} M(H_1)$ and $H \succeq \tau M(CC)$ hold. However, the number of operations for deriving the target theory is different from each other. Using γ-operator, we can derive $M(H_1)$ from CC in such a way that a in CC is expanded to the clause $\neg b \vee a$ and replaced by it and $\neg d$ in CC is expanded to two clauses $\neg d \vee a$ and $\neg d \vee c$ and replaced by them. Then, the number of expanding operations should be three. On the other hand, using inverse subsumption, we can derive H_1 from $\tau M(CC)$ in such a way that the literal $\neg a$ of the clause $\neg a \vee d \vee b$ and the literal d of the clause $\neg a \vee \neg c \vee d$ are dropped. Then, the number of dropping operations should be two. Accordingly, in terms of this target hypothesis, it would be sufficient to use inverse subsumption, since the number of operations is lower.

We next assume another target hypothesis $H_2 = \{\neg a \vee b,\ \neg c \vee d,\ \neg c \vee b\}$. $M(H_2)$ is then $\{a \vee c,\ \neg b \vee c,\ \neg b \vee \neg d\}$. Note that $CC \vdash_{\gamma}^{*} M(H_2)$ and $H_2 \succeq \tau M(CC)$ hold. Using γ-operator, $M(H_2)$ is derived from CC in such a way that two clauses a and $\neg d$ in CC are expanded to the clause $a \vee c$ and the clause $\neg b \vee \neg d$, and replaced by them, respectively. Then, the number of the expanding operations should be two. On the other hand, using inverse subsumption, H is derived from $\tau M(CC)$ in such a way that d of the clause $\neg a \vee b \vee d$ and $\neg a$ of the clause $\neg a \vee \neg c \vee d$ are dropped and the clause $\neg c \vee b$ is added to $\tau M(CC)$. The number of dropping operations should be two. It is equal to the number of deductive operations. However, we have to apply an anti-weakening operator for deriving the clause $\neg c \vee b$. As we described before, anti-weakening operators adding arbitrary clauses are highly non-deterministic. Then, γ-operator efficiently works for deriving H_2.

As we saw in this example, which approach to be used depends on the target hypotheses. The first approach tends to derive somewhat complex hypotheses like H_2, for which we need anti-weakening operators in the case of inverse subsumption. Note that the clause $\neg c \vee b$ in H_2, which is derived by an anti-weakening operator, is not necessarily used to explain E with B. In other words, only using the other two clauses in H_2, E can be explained. Hence, the clause $\neg c \vee b$ does not directly concern with the explanation. The second approach with inverse subsumption is difficult to find such redundant clauses that are not used to explain E. That is because we need to ensure the consistency of the hypotheses with the background theory. Using anti-weakening operators, we can arbitrarily add any clauses to $\tau M(CC)$. However, we have to check if the added theory is consistent with B. We may consider that the difficulty of consistency checking is the same as the first approach based on deductive operations. Actually, the first approach can do it more easily.

We recall the consistency condition between B and H that $B \not\models \neg H$. Since $M(H)$ is logically equivalent to $\neg H$ in the ground case, it is sufficient to satisfy $B \not\models M(H)$. If $B \models M(H)$, $Carc(B, \mathcal{P_I}) \succeq M(H)$ holds. Accordingly, if $Carc(B, \mathcal{P_I}) \not\succeq M(H)$ should hold, H is consistent with B. By the initial condition of bridge theories in CF-induction, any bridge theory includes at least one instance of a clause in $NewCarc(B, \overline{E}, \mathcal{P_I})$. We denote by C this instance. If $M(H)$ should keep C, then H is consistent with B, since $Carc(B, \mathcal{P_I}) \not\succeq M(H)$ holds. Else, let S_C be a clausal theory such that $S_C \subseteq M(H)$ and $\{C\} \vdash_\gamma S_C$. S_C denotes the clauses obtained by applying some γ-operator to C. If $Carc(B, \mathcal{P_I}) \not\succeq S_C$ holds, $Carc(B, \mathcal{P_I}) \not\succeq M(H)$ also holds, since $M(H)$ contains S_C. Hence, it is sufficient to take care about the γ-operation to the clause C for the consistency of H with B. In contrast, if we apply the same way to the second approach using inverse subsumption, we have to additionally compute $M(H)$ and then check if every clause in $M(H)$, instead of some part C_S, is subsumed by a clause in $Carc(B, \mathcal{P_I})$.

From on the above consideration, the first approach with deductive operations would be useful to derive *active hypotheses* such that refer to rules that are not used to explain observations but are at least consistent with the background theory. On the other hand, the second approach with inverse subsumption efficiently works in finding *compressed hypotheses* whose description lengths tend to be short. That is because dropping operators enable us to easily make the description length of the current theory shorter.

In the inductive learning point of view, we are used to seek more compressed descriptions based on the principle of Occum's razor. Then, the second approach may be straightforward to this principle. In contrast, the first approach interestingly takes the risk that hypotheses can contain some *extra* rules that are not necessary to explain observations. In some cases, this efficiently works for giving users some unexpected insights to the prior background theory.

Example 6.3. Let a background theory B and observations E as follows:

$$B = \{female(s) \vee male(s)\}.\ E = \{human(s)\}.$$

Consider a bridge theory $F = \{female(s) \vee male(s),\ \neg human(s)\}$. We can construct $M(H)$ by applying γ-operator to F in such a way that $\neg human(s)$ is expanded to two clauses $\neg human(s) \vee female(s)$ and $\neg human(s) \vee male(s)$ and replaced by them. $M(H)$ is as follows:

$$\{female(s) \vee male(s),\ \neg human(s) \vee female(s),\ \neg human(s) \vee male(s)\}.$$

Then, we obtain the following H by computing $M(M(H))$:

$$\{\neg female(s) \vee human(s),\ \neg male(s) \vee human(s),\ \neg female(s) \vee \neg male(s)\}.$$

In addition, if we apply an anti-instantiation operator to H, we also obtain another hypothesis represented as follows:

$$\{female(X) \supset human(X),\ male(X) \supset human(X),$$
$$\neg female(X) \vee \neg male(X)\}.$$

We notice that the last clause $\neg female(X) \vee \neg male(X)$ does not involve in explaining E. Instead, it can be regarded as an integrity constraint on the predicates $female(X)$ and $male(X)$ that is consistent with B.

6.5 Summary

In this chapter, we have studied that the generalization procedure of CF-induction can be logically simplified while preserving its soundness and completeness. In Section 6.2, we introduced the γ-operator whose task is removing some clause D in an input clausal theory and adding a set of clauses $C_1, \ldots, C_n$ for some $0 \leq n$ where each clause C_i is a super set of D. We also showed that the γ-operator and ant-instantiation are sufficient to ensure the completeness of CF-induction. In Section 6.3, we have proposed an alternative approach based on inverse subsumption with minimal complements. Induction fields have been extended so as to describe the syntax of target hypotheses. We have clarified the logical relation between induction fields and production fields that enable us to integrate inverse subsumption in Chapter 6 with generalization of CF-induction. We have also studied the issue how bridge theories are deterministically constructed. As a result, we proposed an incremental way to add clauses one by one to the current bridge theory. Compared with the previous version of CF-induction, the difference is that our proposal can deterministically construct a bridge theory and any hypothesis wrt B, E and $\mathcal{I}_{\mathcal{H}}$ is derived based on the inverse subsumption, instead of entailment. In this way, the non-determinism of CF-induction has been logically reduced, while it ensures the soundness and completeness. In Section 6.4, we have studied two approaches in terms of their search strategies as well as their differences using several examples.

Chapter 7

Conclusions and Future Work

7.1 Summary

In this thesis, we have investigated the logical mechanism and computational procedures in IE-based explanatory methods. Compared with other inductive learning paradigms, their characteristic role lies in the applicability to completion tasks for finding some missing facts or causal relations in the prior knowledge. Based on this merit, we have provided a practical example in systems biology. This motivating problem was necessary to find both missing facts and general rules, simultaneously. Whereas significant progress has been made in molecular biology, the prior biological inference is still incomplete. For this situation, we have showed an inherent possibility to find some missing rules and facts in the context of explanatory induction.

This example involves logical estimation of possible reaction states, that is, activated or inactivated, in metabolic pathways. Like dynamic change from fermentation to oxidation emerged in microorganisms, transitions of highly-activated (master) reactions affect the emergence of phenotype. For analyzing those reaction states, we introduced a logical model that describes some of biological relations, and then use the ILP setting to estimate reaction states with respect to the logical model. The initial experimental result showed that CF-induction could realize an advanced inference simultaneously integrating both abduction and induction. On the one hand, abductive inference was used to estimate possible reaction states. On the other hand, induction was used to complete missing general rules in the prior theory. In the initial experiment, we have shown that CF-induction could find complex hypotheses involving in both estimation and completion tasks.

Unlike the other IE-based ILP methods, CF-induction is sound and complete for finding hypotheses in full clausal theories. CF-induction consists of two non-deterministic procedures: The first one is construction of bridge theories and the second is generalization into hypotheses. These two procedures are commonly used in the modern IE-based methods, and each one constructs a bridge theory and generalizes into hypotheses in its own way. In terms of the generalization relation, methods like CF-induction use the inverse relation of entailment to ensure the completeness. On the other hand, methods like Progol are based on subsumption, instead of entailment, due to computational efficiency. However, it had not been clarified whether or not the logical reduction from entailment to subsumption could cause some incompleteness in generalization. For this open problem, the thesis has shown that inverse subsumption with minimal complements can ensure the completeness in generalization. We have also investigated the case of residue complements and pointed out their fundamental limitation that arises in the fact for a ground clausal theory S, it does not necessarily hold $R^2(S) = S$. In contrast, we have shown the property of minimal complements that $M^2(S) = \mu S$ holds.

To treat the inverse relation of entailment, we had to deal with a variety of generalizers each of which has many ways to be applied. This fact made generalization highly non-deterministic. In this sense, our result can be used to reduce the non-determinisms in IE-based methods, including CF-induction that originally uses the inverse entailment for the generalization relation. We then considered how the generalization procedure of CF-induction can be logically simplified based on two approach.

The first approach uses deductive operations that enable us to directly derive the negation of a particular hypothesis from a bridge theory. Bridge theories of CF-induction are also constructed in manner of consequence finding techniques. Hence, we may be able to regard the first approach as a specific case that deductive operations are used in deriving hypotheses as much as possible. As a result, we have proposed a new deductive operator, called

γ-operator, that is sufficient to derive any negations of ground hypotheses, and showed that a concatenation of γ-operator and anti-instantiation ensures the soundness and completeness of generalization. The second approach uses inverse subsumption with minimal complements. In this approach, we first clarified the logical relation between induction fields and production fields. Along with that, we have extended induction fields into a richer representation formalization that can describe the syntax of target hypotheses (i.e. maximum length and number of clauses in the target hypotheses). We also compared those two approaches for generalization of CF-induction in terms of their search strategies as well as their characteristics using several examples.

The original CF-induction contains another non-deterministic procedure: construction of bridge theories. Let n be the number of characteristic clauses. Then, the possible choice points to select a bridge theory should be $2^n - 1$. We have shown that without losing completeness for finding hypotheses, this non-deterministic procedure can be simplified into a deterministic way. This result is based on the property that for every two bridge theories CC_1 and CC_2 such that $CC_1 \subset CC_2$, $\tau M(CC_1) \succeq \tau M(CC_2)$ holds. This property implies that any hypothesis obtained from $\tau M(CC_1)$ can be derived from $\tau M(CC_2)$. In brief, if $Carc(B \wedge \overline{E_{sk}}, \mathcal{P})$ contains a finite number of ground clauses $\{C_1, \ldots, C_n\}$, it is enough to select all those ground clauses in the bridge theory. Due to interests of computational efficiency, the thesis has introduced an arbitrary order over characteristic clauses, and then proposed an incremental way to construct bridge theories.

This thesis gives logical foundations for the reduction from inverse entailment to inverse subsumption. Using them, we reconstruct the non-deterministic procedures of CF-induction into more simplified ones.

7.2 Future Work

The issue on hypothesis enumeration by CF-induction needs to be addressed in future work. The theoretical advantage of CF-induction is to preserve the completeness for finding hypotheses in full clausal theories. In other words, CF-induction is the unique procedure that can enumerate all the possible hypotheses. If it should become possible to enumerate the target hypotheses, we would be able to obtain some new hypotheses in many cases. Currently, we reach the fact that for every hypothesis H, it holds that $CC \vdash_{\gamma}^{*} M(CC)$ and $H \succeq \tau M(CC)$ for some bridge theory CC. In this connection, we remark the issue on how to realize the generalization algorithm based on inverse subsumption as well as γ-operator. Though we have shown that the previous generalization can be reduced into inverse subsumption or a specific deductive operator, we do not describe those practical algorithms in the thesis. We then intend to develop heuristics for searching relevant hypotheses on the subsumption lattice. It would be fruitful to investigate ways of automatically finding which literals must be added to selected clauses by the γ-operator. We believe that studying various restrictions of the γ-operator may allow us to systematically compare the generalisation power of previously proposed operators. Besides, we have introduced the notion of induction fields to efficiently focus on the target hypotheses that users wish to obtain. As we discussed in Section 5.5, in the specific case that target hypotheses are ground and minimal, arbitrary hypothesis can be obtained by applying only dropping operators. Accordingly, it would be possible to enumerate the hypotheses using dropping operators at least with brute force.

Other important future work is efficient algorithms for computing minimal complements. Computation of the minimal complement is equivalent to enumeration of the minimal hitting sets, as we have shown in Chapter 2. Though this task is solvable in quasi-polynominal time (In brief, $n^{O(log^2 n)})$), in case that a given induction field contains a lot of complementary literals, the computational cost should not be ignorable. Because the number of tautological clauses added to a bridge theory becomes large. This problem is related to how relevant induction fields can be considered in advance. Progol systems use the notion of mode declarations to describe the head and body literals. Thus, it would be fruitful how mode declarations can be translated into the notion of induction fields. In computational efficiency point of view, it would be convenient to assume such an induction field where the number of complementary literals is as small as possible. It is interesting to consider why we have to add tautological clauses that never affect in the semantics.

When we look back on the examples in the thesis, its role tends to *invent* some unknown predicates that never appear in the prior knowledge. For instance, in Example 3.3, neither $arc(b, c)$ nor $\neg arc(b, c)$ appear in B and E. Despite that, we allow ourself to includes two complementary literals $arc(b, c)$ and $\neg arc(b, c)$ in the induction field.

It would be worth considering the issue on hypotheses evaluation that was not sufficiently described in the thesis. In general, possible hypotheses are not unique. In other word, there are a lot of logical theories that satisfy the two conditions of explanatory induction However, most of the previously proposed systems output the unique hypothesis according to their own specifications. Some adherent of pragmatism may counter as follows: It will be sufficient that the system can find a certain valid hypothesis at some point. It is thus practically difficult especially for the situations that need highly expensive experiments to sequentially test the current hypothesis one by one. In terms of abduction, SOLAR can automatically compute the subsumption-minimal abductive hypotheses. There is a recent work [28] to statistically evaluate those abductive hypotheses by SOLAR. Then, it would be worth considering how CF-induction can *enumerate* the target inductive hypotheses and evaluate them like SOLAR.

Bibliography

[1] E. Almaas, B. Kovács, B. Vicsek, Z. N. Oltvai and A. Barabási. Global organization of metabolic fluxes in the bacterium escherichia coli. *Nature*, 427(6977):839–843, 2004.

[2] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1-3):21–46, 1996.

[3] L. Badea. A refinement operator for theories. In *Proc. of the 11th Int. Conf. on Inductive Logic Programming*, volume 2157 of *LNCS*, pages 1–14. Springer, 2001.

[4] R. Banerji. A language for the description of concepts. *General Systems*, 9:135–141, 1964.

[5] M. Bramer. *Principles of Data Mining*. Springer, 2007.

[6] J. Carbonell, editor. *Machine learning paradigms and methods*. 1990.

[7] C. Chang and R. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, INC., 1987.

[8] K. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, pages 119–140, 1978.

[9] P. Cox and T. Pietrzykowski. A complete, nonredundant algorithm for reversed skolemization. *Theoretical Computer Science*, (3):239–261, 1984.

[10] M. Davis and H. Putnam. A computing procedure for quantification theory. *the ACM*, 7:201–215, 1960.

[11] Y. Deville, D. Gilbert, J. van Helden and S. Wodak. An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics*, (4):246–259, 2003.

[12] A. Doncescu, K. Inoue and T. Sato. Hypothesis-finding in systems biology. *The ALP Newsletter*, 21(1-3), 2008.

[13] A. Doncescu, K. Inoue and Y. Yamamoto. Knowledge-based discovery in systems biology using CF-induction. In *New Trends in Applied Artificial Intelligence: Proc. of the 20th Int. Conf. on Industrial, Engineering and Other Applications of Applied Intelligence Systems*, volume 4570 of *LNAI*, pages 395–404. Springer, 2007.

[14] A. Doncescu, Y. Yamamoto and K. Inoue. Biological systems analysis using inductive logic programming. In *Proc. of the 21st Int. Conf. on Advanced Information Networking and Applications*, pages 690–695. IEEE Computer Society, 2007.

[15] T. Eiter and G. Gottlob. Hypergraph transversal computation and related problems in logic and AI. In *Proc. of 8th European Conf. on Logics in Artificial Intelligence*, volume 2424 of *LNCS*, pages 549–564. Springer, 2002.

[16] T. Eiter and K. Makino. Abduction and the dualization problem. In *Proc. of 6th Int. Conf. on Discovery Science*, volume 2843 of *LNCS*, pages 1–20. Springer, 2003.

[17] T. Either, K. Makino and G. Gottlob. Computational aspects of monotone dualization: a brief survey. *Discrete Applied Mathematics*, 156:2035–2049, 2008.

[18] S. Feferman *et al.*, editor. *Gödel's collected works.* Oxford University Press, 1980.

[19] M. Fredman and L. Khanchiyan. On the complexity of dualization of monotone disjunctive normal forms. *Algorithms*, 21:618–628, 1996.

[20] E. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[21] É. Grégoire and L. Saïs. Inductive reasoning is sometimes deductive. In *ECAI-96 Workshop on Abductive and Inductive Reasoning*.

[22] N. Helft. Induction as nonmonotonic inference. In *Proc. of the 1st Int. Conf. on Principles of Knowledge Representation and Reasoning*, volume 7, pages 149–156, 1989.

[23] D. Hume. *A treatise of human nature*. Dover Philosophical Classics. Dover Publications, Inc., 1739.

[24] K. Inoue. Principles of abduction. *Japanese Society for Artificial Intelligence*, 7(1):48–59, 1991.

[25] K. Inoue. Linear resolution for consequence finding. *Artificial Intelligence*, 56(2-3):301–353, 1992.

[26] K. Inoue. Induction as consequence finding. *Machine Learning*, 55(2):109–135, 2004.

[27] K. Inoue and H. Saito. Circumsciption policies for induction. In *Proc. the 14th Int. Conf. on Inductive Logic Programming*, volume 3194 of *LNAI*, pages 164–179. Springer, 2004.

[28] K. Inoue, T. Sato, M. Ishihata, Y. Kameya and H. Nabeshima. Evaluating abductive hypotheses using an EM algorithm on BDDs. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence*, pages 810–815, 2009.

[29] K. Iwanuma, K. Inoue, and K. Satoh. Completeness of pruning methods for consequence finding procedure SOL. In B. P. and Z. H., editors, *Proc. of the Third Int. Workshop on First-Order Theorem Proving*, pages 89–100, 2000.

[30] M. Kanehisa and S. Goto. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28:27–30, 2000.

[31] T. Kimber, K. Broda and A. Russo. Induction on failure: learning connected horn theories. In *Proc. of Logic Programming and nonmonotonic reasoning*, Lecture Notes in Computer Science, pages 168–181, 2009.

[32] R. King, K. Whelan, F. Jones, P. Reiser, C. Bryant, S. Muggleton, D. Kell and S. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427:247–252, 2004.

[33] N. Lachiche. Abduction and induction from a non-monotonic reasoning perspective. In P. A. Flach and A. C. Kakas, editors, *Abduction and Induction: Essays on their relation and integration*, 2000.

[34] P. Laird. *Learning from good and bad data.* Kluwer Acamedic Publishers, 1988.

[35] J. Larson and R. Michalski. Inductive inference of VL decision rules. In *Proc. of the Workshop in Pattern-Directed Inference Systems*, pages 38–44, 1977.

[36] N. Lavrač and S. Džeroski. *Inductive logic programming: techniques and applications.* 1994.

[37] C. Lee. *A completeness theorem and computer program for finding theorems derivable from given axioms.* PhD thesis, Dept. of Electronic Eng. and Computer Sci., Univ. of California, Berkeley, CA, 1967.

[38] J. Lloyd. *Foundations of logic programming.* Springer, second edition, 1987.

[39] J. McCarthy. Circumscrition-a form of nonmonotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980.

[40] P. Mendes and D. Kell. Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, 14:869–883, 1998.

[41] H. Midelfart. A bounded search space of clausal theories. In *Proc. of the 9th Int. Conf. on Inductive Logic Programming*, volume 1634 of *LNAI*, pages 210–221. Springer, 1999.

[42] T. Mitchell. *Version spaces: an approach to concept learning.* PhD thesis, Stanford University, 1978.

[43] S. Muggleton. Inductive logic programming. *New Generation Computing*, 4(8):295–318, 1991.

[44] S. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13(3-4):245–286, 1995.

[45] S. Muggleton. Scientific knowledge discovery using inductive logic programming. In *Communications of the ACM*, volume 42, pages 42–46, 1999.

[46] S. Muggleton and C. Bryant. Theory completion using inverse entailment. In *Proc. of the 10th Int. Conf. on Inductive Logic Programming*, volume 1866 of *LNAI*, pages 130–146, 2000.

[47] S. Muggleton and W. Buntine. Machine invention of first order predicates by inverting resolution. In *Proc. of the 5th Int. Conf. on Machine Learning*, pages 339–352, 1988.

[48] S. Muggleton and L. de Raedt. Inductive logic programming: theory and methods. *Logic Programming*, 19/20:629–679, 1994.

[49] S. Muggleton and C. Feng. *Efficient induction of logic programs.* Inductive Logic Programming. Academic Press, 1988.

[50] S. Muggleton, R. King and M. Sternberg. Protein secondary structure prediction using logic-bsed machine. *Protein Engineering*, 5(7):647–657, 1992.

[51] H. Nabeshima, K. Iwanuma and K. Inoue SOLAR: A consequence finding system for advanced reasoning. In *Proc. of TABLEAUX 2003*, volume 2796 of *LNAI*, pages 257–263. Springer, 2003.

[52] S. Nienhuys-Cheng and R. de Wolf. *Foundations of inductive logic programming*, volume 1228 of *LNCS*. Springer, 1997.

[53] C. Peirce. *Elements of logic*, volume 2 of *Collected Papers of Charles Sanders Peirce*. Harvard University Press, 1932.

[54] G. Plotkin. *Automatic methods of inductive inference.* PhD thesis, Edinburgh University, 1971.

[55] G. Plotkin. A further note on inductive generalization. *Machine Intelligence*, 6:101–124, 1971.

[56] G. Plotkin. A powerdomain construction. *SIAM Journal of Computing*, 5:452–497, 1976.

[57] J. Quinlan. Discovering rules from large collections of examples: a case study. *Expert Systems in the Micro electronic Age*, pages 168–201, 1979.

[58] J. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[59] O. Ray. *Hybrid Abductive Inductive Learning.* PhD thesis, The University of London and the Diploma of Imperial College, 2005.

[60] O. Ray. Nonmonotonic abductive inductive learning. *Applied Logic*, 7:329–340, 2009.

[61] O. Ray, K. Broda and A. Russo. Hybrid abductive inductive learning: a Generalisation of Progol. In *Proc. of the 13th Int. Conf. on Inductive Logic Programming*, volume 2835 of *LNAI*, pages 311–328. Springer, 2003.

[62] O. Ray and K. Inoue. A consequence finding approach for full clausal abduction. In *Proc. of the 10th Int. Conf. on Discovery Science*, volume 4755 of *LNAI*, pages 173–184. Springer, 2007.

[63] O. Ray and K. Inoue. Mode directed inverse entailment for full clausal theories. In *Revised and selected papers from the 17th Int. Conf. on*

Inductive Logic Programming, volume 4894 of *LNAI*, pages 225–238, 2007.

[64] J. Robinson. A machine-oriented logic based on the resolution principle. *JACM*, 12(1):23–41, 1965.

[65] S. Russell and P. Norvig. *Artificial intelligence a modern approach.* Pearson Education, Inc., 2003.

[66] C. Sakama and K. Inoue. Brave induction. In *Proc. the 18th Int. Conf. on Inductive Logic Programming*, volume 5194 of *LNAI*, pages 261–278. Springer, 2008.

[67] C. Sammut and R. Banerji. Learning concepts by asking questions. *Machine learning: an artificial intelligence approach*, 2:167–192, 1986.

[68] K. Satoh and T. Uno. Enumerating maximal frequent sets using irredundant dualization. In *Proc. of the 6th Int. Conf. on Discovery Science*, volume 2843 of *LNCS*, pages 256–268, 2003.

[69] C. Schilling, D. Letscher and B. Palsson. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Theoretical Biology*, 203:229–248, 2000.

[70] K. Schugerl and K. Bellgardt. *Bioreaction engineering: modeling and control.* Springer, 2000.

[71] S. Schuster, D. Fell and T. Dandekar. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology*, 18:326–332, 2000.

[72] E. Shapiro. *Algorithmic program debugging.* PhD thesis, Yale University, 1983.

[73] E. Shortliffe and B. Buchanan. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23:351–379, 1975.

[74] G. Stephanopoulos, A. Aristidou and J. Nielsen. *Metabolic engineering.* Academic Press, 1998.

[75] A. Tamaddoni-Nezhad and S. Muggleton. The lattice structure and refinement operators for the hypothesis space bounded by a bottom clause. *Machine Learning*, 76:37–72, 2009.

[76] A. Tamaddoni-Nezhad, R. Chaleil and S. Muggleton. Application of abductive ilp to learning metabolic network inhibition from temporal data. *Machine Learning*, 65:209–230, 2006.

[77] A. Turing and M. Woodger. *The automatic computing engine.* 1947.

[78] T. Uno. A practical fast algorithm for enumerating minimal set coverings. *IPSJ SIG Notes*, 29:9–16, 2002.

[79] L. Valiant A theory of the learnable. *Communications of the ACM*, 27(11):1132–1142, 1984.

[80] P. van der Laag. *An analysis of refinement operators in inductive logic programming.* PhD thesis, Erasmus Universiteit Rotterdam, 1995.

[81] A. Varma and B. Palsson. Metabolic flux balancing: basic concepts, scientific and practical use. *Nature Biotechnology*, 12:994–998, 1994.

[82] A. Yamamoto. Which hypotheses can be found with inverse entailment? In *Proc. of the 7th Int. Conf. on Inductive Logic Programming*, volume 1297 of *LNCS*, pages 296–308, 1997.

[83] A. Yamamoto. Hypothesis finding based on upward refinement of residue hypotheses. *Theoretical Computer Science*, 298:5–19, 2003.

[84] A. Yamamoto and B. Fronhöfer. Hypotheses finding via residue hypotheses with the resolution principle. In *Proc. of the 11th Int. Conf. on Algorithmic Learning Theory*, volume 1968 of *LNAI*, pages 156–165. Springer, 2000.

[85] Y. Yamamoto and K. Inoue. An efficient hypothesis-finding system implemented with deduction and dualization. In *Proc. of the 22nd Workshop on Logic Programming*, pages 92–103. University Halle-Wittenberg Institute of Computer Science Technical Report, 2008.

[86] Y. Yamamoto, K. Inoue and A. Doncescu. Estimation of possible reaction states in metabolic pathways using inductive logic programming. In *Proc. of the 22nd Int. Conf. on Advanced Information Networking and Applications*, pages 808–813. IEEE Computer Society, 2008.

[87] Y. Yamamoto, K. Inoue and A. Doncescu. Abductive reasoning in cancer therapy. In *Proc. of the 23rd Int. Conf. on Advanced Information Networking and Applications*, pages 948–953. IEEE Computer Society, 2009.

[88] Y. Yamamoto, K. Inoue and A. Doncescu. Integrating abduction and induction in biological inference using CF-induction. In *Elements of Computational Systems biology*, Wiley Book Series on Bioinformatics, pages 213–234. John Wiley and Sons, Inc., 2009.

[89] Y. Yamamoto, K. Inoue and K. Iwanuma. Hypothesis enumeration by CF-induction. In *Proc. of the 6th Workshop on Learning with Logics and Logics for Learning*, pages 80–87. The Japanese Society for Artificial Intelligence, 2009.

[90] Y. Yamamoto, K. Inoue and K. Iwanuma. From inverse entailment to inverse subsumption. In *Proc. of the 20th Int. Conf. on Inductive Logic Programming*, 2010.

[91] Y. Yamamoto, O. Ray and K. Inoue. Towards a logical reconstruction of CF-induction. In *Proc. of the 4th Int. Workshop on Learning with Logics and Logics for Learning*, pages 18–24. The Japanese Society for Artificial Intelligence, 2007.

[92] Y. Yamamoto, O. Ray and K. Inoue. Towards a logical reconstruction of CF-induction. In Satoh K., Inokuchi A., Nagao K., and Kawamura T.,

editor, *New Frontiers in Artificial Intelligence: JSAI 2007 Conference and Workshop Revised Selected Papers*, volume 4914 of *LNAI*, pages 330–343. Springer, 2008.

[93] B. Zupan, I. Bratko, J. Demšar, J. Beck, A. Kuspa and G. Shaulsky. Abductive inference of genetic networks. In *8th Conf. on AI in Medicine in Europe*, volume 2101, pages 304–313, 2001.

[94] B. Zupan, J. Demsar, I. Bratko, P. Juvan, J. Halter, A. Kuspa and G. Shaulsky. Genepath: a system for automated construction of genetic networks from mutant data. *Bioinformatics*, pages 383–389, 2003.