

# AN ABDUCTIVE PROPOSITIONAL LOGIC FOR DESIGN REASONING

Feng-Tyan Lin, Professor

Graduate Institute of Building and Planning, National Taiwan University  
Also with (1) Department of Civil Engineering, National Taiwan University,  
(2) Institute of Information Science, Academia Sinica, Taiwan  
1, Sec. 4, Roosevelt Rd., Taipei, Taiwan 106, R.O.C.

Hunghsiang Wang, Associate Professor

Department of Industrial Design, Shih Chien University  
70 Ta-Chih St., Taipei, Taiwan 104, R.O.C.

## ABSTRACT

Design can be seen as a reasoning process based on non-classic logic, in which non-monotonicity and abduction are two essential features. This paper introduces *Abductive Propositional Logic* (APL), which is a simple but powerful formal system to deal with these two features. To show its capabilities, APL is used to analyze the reasoning of a conceptual design of a scooter. The results reveal that APL is not only a profound basis for developing computer supported design systems, but also a useful tool for analyzing design processes.

**Keywords:** design reasoning, non-monotonicity, abduction

## I. INTRODUCTION

It is commonly recognized that logic is an effective tool to represent design knowledge, explain reasoning processes, and review results of design thinking in an explicit way. However, classic logic is based on deductive reasoning, in which

knowledge is monotonically increased. Obviously, classic logic cannot capture all the characteristics of a design process, where abduction, induction, analogy and other reasoning methods are practiced. Thus, the capability of design logic needs to be largely expanded.

On the other hand, conceptual design is one of the designer's primary activities (Ulrich and Eppinger, 1995; Baxter, 1995; Wang, 1995, 1996). It involves a lot of creation, evolution, modification, revision, rejection, and regeneration of ideas, which are expressed in terms of graphical (visual) and textual (verbal) statements. These statements can further be translated into logic expressions. Therefore, a formal analysis of the reasoning process in conceptual design can be quite suitable for exploring the usefulness of a non-classic logic. For example, a partial reasoning process about a hairdryer design is shown in Figure 1. Supposing that the goal *a* is to design a kind of hairdryer for travelers, the designer may use the rule *b* to deduce the conclusion *c*, the hairdryer is a compact one. Furthermore, the conclusion *c* and the rule *d* abductively suggest that the hairdryer might have a foldable handle (i.e., *e*).

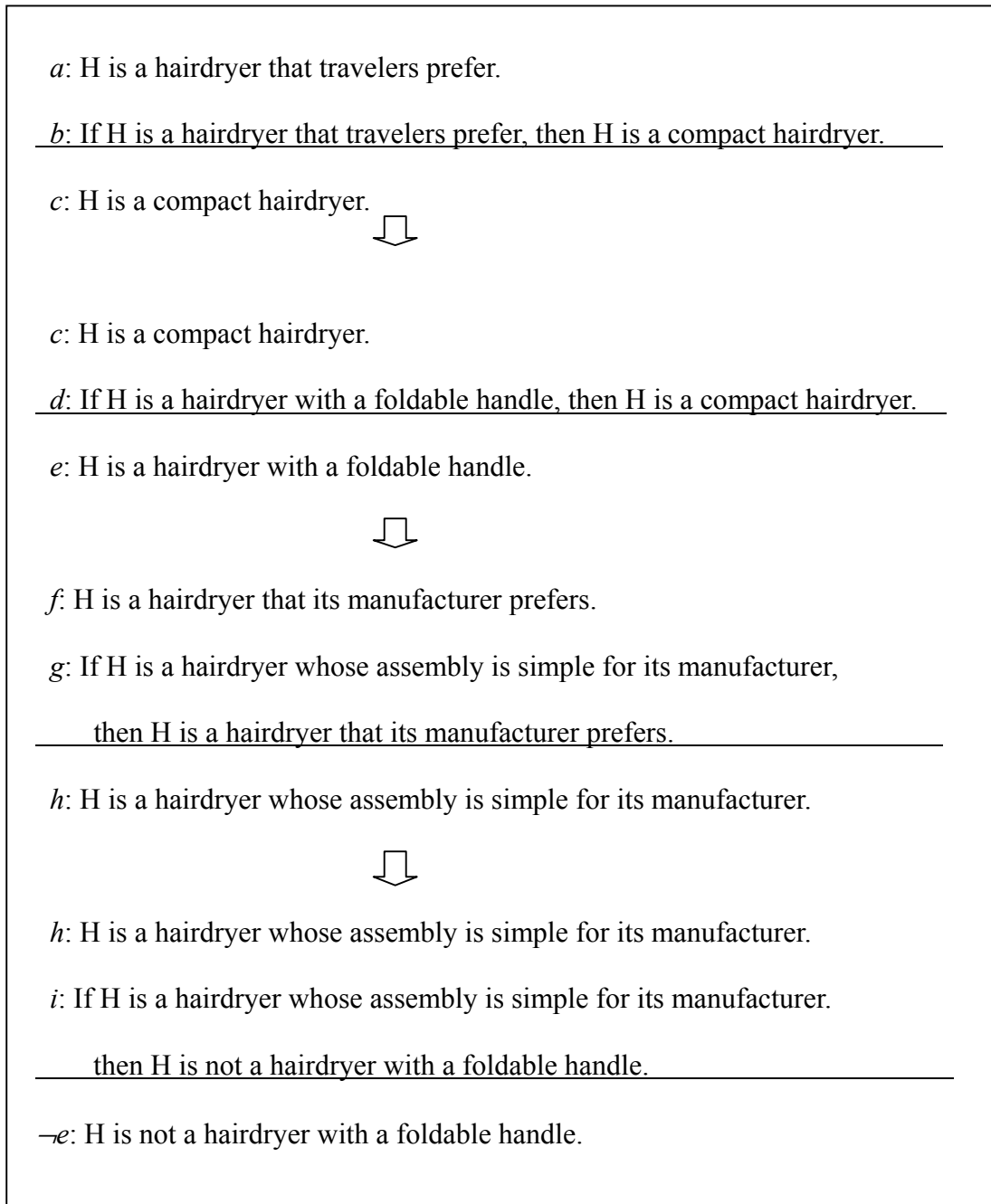


Figure 1. A partial reasoning process for a hairdryer design.

However, there are other contrary considerations. For instance, the designer may take the manufacturer's preference (i.e., *f*) into account and invoke the rule *g* to abduce that the hairdryer should be easy to mold and assemble (i.e., *h*). The conclusion *h* and the rule *i* deduce a hairdryer without any foldable handle (i.e.,  $\neg e$ ).

As a result, the new conclusion  $\neg e$  is inconsistent with the previous conclusion  $e$ . At this moment, the designer has two alternatives:  $e$  and  $\neg e$ , of which either may be rejected to maintain logic consistency. In case conclusion  $\neg e$  is rejected,  $h$  or  $i$  should be also rejected to invalidate the deductive derivation of conclusion  $\neg e$ . On the other hand, no other logical statements require rejection after the rejection of conclusion  $e$ , which can be seen as just an abductive conjecture.

This example reveals three characteristics of design reasoning. First, abductively derived conclusions may be retracted when a new consideration is concerned. This characteristic is called *non-monotonicity (NM)* (Genesereth and Nilsson, 1987; Coyne, 1988; Rich and Knight, 1991; Ginsberg, 1993; Luger, 1993; Giarratano and Riley, 1994). Second, when an inconsistency occurs, design reasoning may tolerate it to a certain extent, and tries to explore as many alternatives as possible to resolve it. This characteristic is similar to de Bono's (1970) *lateral thinking*. Third, each statement may have some logical relations with others. These relations may cause a chain-retraction, which is a series of retractions evoked by an initial one. To explore these three characteristics, this paper will start from the issues of abduction and leave other features, such as induction and analogy, for future study.

## II. ABDUCTION AND NON-MONOTONICITY

In classic logics, only deduction is employed; therefore, addition of new premises should never invalidate previous conclusions (Coyne, 1988; Genesereth and Nilsson, 1987). In terms of formal logic, a new clause cannot be accepted by classic

logics if it violates the consistency of the existing system. In other words, knowledge, theorems and beliefs are monotonically increased, and existing ones are never withdrawn. Obviously, monotonicity is inadequate for dealing with the design reasoning where the existing abductively derived statements can be falsified and then removed (e.g., Esgni and Kowalski, 1989; Cadoli and Schaerf, 1993; Pereira et al, 1993; Baral and Gelfond, 1994), and contradictions are handled differently.

Abduction is thought to be one of the major inferences in design reasoning (March, 1984; Coyne, 1988; Roozenburg, 1993; Rowe, 1994; Roozenburg and Eekels, 1995). A simple structure of abduction is shown in Figure 2. However, scientists and designers use abduction in a different way. While scientists use abduction to explain the ‘reasons’ of nature, designers explore possible designs. For example, a scientist may abductively conclude that ‘The occurrence of  $p$  is the reason for event  $q$ ’ if he or she believes that ‘ $q$  follows  $p$ ’ and observes that  $q$  occurs. However, designers may interpret  $p \rightarrow q$  as ‘If property  $p$  holds, goal  $q$  can be achieved.’ Therefore, to achieve goal  $q$ , property  $p$  should be designed.

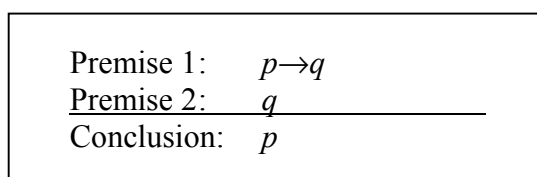


Figure 2. The structure of abduction.

NM has importance in the area of *artificial intelligence* (Rich and Knight, 1991; Ginsberg, 1993; Luger, 1993; Giarratano and Riley, 1994; Genesereth and Nilsson, 1997). In particular, many researchers have addressed the correlation between abductive reasoning and *non-monotonic reasoning (NMR)* from the viewpoint of

logical programming. Researchers have proposed some non-monotonic logic systems, e.g., McCarthy's (1980) *circumscription*, McDermott and Doyle's (1980) *non-monotonic logic*, Reiter's (1980) *default logic*, and Moore's (1985) *autoepistemic logic*. Many extensive versions can be seen as implementations of the ideas of non-monotonic logics, such as Doyle's (1979) *justification-based truth maintenance system* (JTMS), de Kleer's (1986) *assumption-based truth maintenance system* (ATMS), and Dov M. Gabbay's (1996) *labeled deductive systems*. For a comprehensive survey of the non-monotonic logics, see (Cadoli and Schaerf, 1993).

These efforts encouraged the study of NM in design reasoning; NM has become one of the most important topics in the area of design computing (Coyne, 1988; Mitchell, 1990a, 1990b; Galle, 1996a, 1996b). However, these non-monotonic logic systems are still arguable; more researches with various approaches are needed. In this sense, this paper tries to develop a simpler and more practical approach based on propositional logic.

### III. ABDUCTIVE PROPOSITIONAL LOGIC

To explore essential characteristics of NM, this paper starts out to extend propositional logic, which has a simple form, to a logic system with both deduction and abduction for NM. Since this logic system highlights abduction, it is called *Abductive Propositional Logic (APL)*.

In terms of formal logics, all statements in APL are *well-formed formulas*, with the same definitions and syntactic forms as those in conventional propositional logic. However, a statement derived by abduction may be falsified later during reasoning.

In other words, some statements are temporarily held. Thus, statements of APL are called *beliefs*, including *believed facts* and *believed rules*, to reflect that they are only true in some situations. A belief is *deductively true* if and only if (iff) it is derived by deduction; a belief is *abductively possible* iff it is derived by abduction. Still, a belief is called an *accepted belief* in a certain situation iff it is *not* retracted in this situation; a belief is called a *retracted belief* in a certain situation iff it is retracted in this situation.

Derivations of beliefs can be denoted in the form

$$\{\alpha_1, \alpha_2, \dots, \alpha_n\} \models^\gamma \beta$$

where  $\alpha_1, \alpha_2, \dots, \alpha_n$  are beliefs; the symbol  $\models$  represents that  $\alpha_1, \alpha_2, \dots, \alpha_n$  are the premises of  $\beta$ ; the symbol  $\gamma$  represents the inference rule used, where  $\gamma$  has three instances:  $\delta$  for deduction,  $\psi$  for abduction, and  $o$  for no premise and inference rule (i.e., ‘ $\beta$  is given as an assumption’). Accordingly, there are three types of derivations. First, *deductive derivation*  $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \models^\delta \beta$  represents that  $\beta$  is deductively true if  $\alpha_1, \alpha_2, \dots, \alpha_n$  are all accepted. Second, *abductive derivation*  $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \models^\psi \beta$  represents that  $\beta$  is abductively possible if  $\alpha_1, \alpha_2, \dots, \alpha_n$  are all accepted. Third, *assumptive derivation*  $\{ \} \models^o \beta$  represents that  $\beta$  is assumptively true, without any premise and inference rule. For short, it can be represented in a form as  $\beta$ . These derivations can be illustrated in the *derivation diagrams* shown in Figure 3.

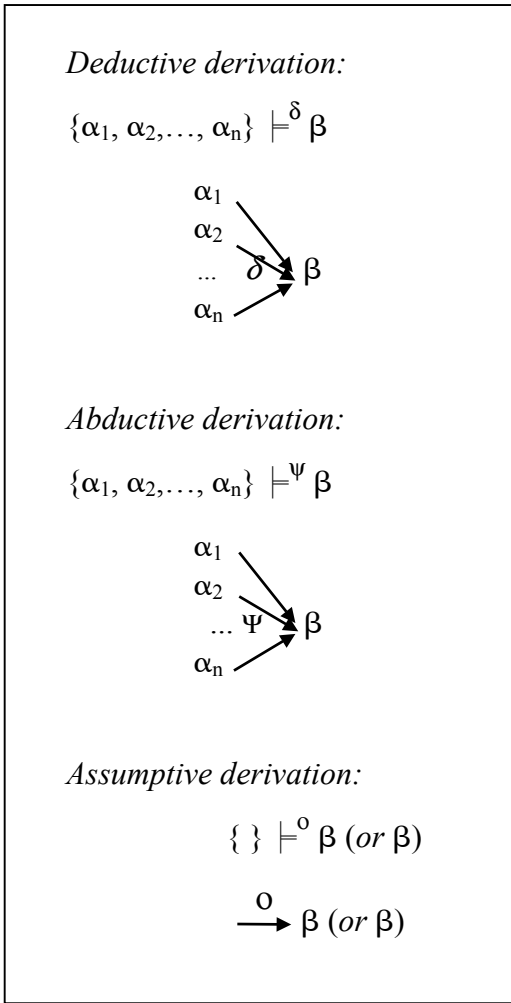


Figure 3. Three types of derivation diagrams.

Adding new beliefs into an existing reasoning system may deductively or abductively derive inconsistent beliefs. When two inconsistent beliefs, say  $\beta$  and  $\neg\beta$  in Figure 4, are detected, the symbol ‘ $\square$ ’ is used to denote the inconsistency.

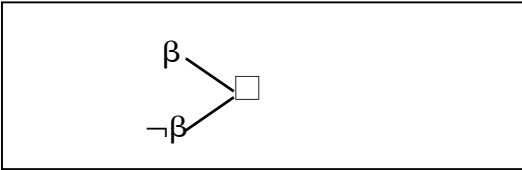


Figure 4. The diagram for two inconsistent beliefs.



The detection of inconsistent beliefs is a start point to identify *APL alternatives*. By definition, an APL alternative is a set consisting of *all* of the accepted beliefs in a certain situation, in which each APL alternative is unique, for it is not a subset of any other APL alternative. All elements of each APL alternative are consistent, because they are accepted beliefs. An APL alternative represents a possible solution, consisting of the maximum consistent accepted beliefs. It is noted that the structure of APL alternatives is based on the form of reasoning, instead of the cognitive process of human designers.

When APL detects an inconsistency as shown in Figure 4, it neither simply terminates the reasoning nor immediately decides to retract  $\beta$  or  $\neg\beta$ . Instead, it will retract either  $\beta$  or  $\neg\beta$ , but not both, in order to explore all the possibilities. The results will be two APL alternatives:  $\{\beta\}$  and  $\{\neg\beta\}$ . The action of retraction can be represented as the first retraction rule *exclusive retraction (R1)* as below. It is noted that the second antecedent (i.e., ' $\beta$  and  $\neg\beta$  have not yet been retracted') is for avoiding iterative retractions to  $\beta$  and  $\neg\beta$ .

**R1. (Exclusive retraction)**

If(1) there exist two derivations:  $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \models^{\gamma^1} \beta$  and  $\{\sigma_1, \sigma_2, \dots, \sigma_m\} \models^{\gamma^2} \neg\beta$ ,

and

(2)  $\beta$  and  $\neg\beta$  have not yet been retracted,

then retract either  $\beta$  or  $\neg\beta$ .

Once a belief is retracted, two kinds of chain-retractions may happen. First,

when a retracted belief is a premise of the other beliefs, all deductive and abductive conclusions of this retracted belief should be also retracted. This action of retraction can be represented as the second retraction rule, namely, *conclusive retraction (R2)* below. It is noted that the third antecedent (i.e., ‘ $\beta$  has not yet been retracted’) is for avoiding iterative retractions to  $\beta$ .

**Rule 2. (Conclusive retraction)**

If (1) there exists a derivation  $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \models^\delta \beta$  or  $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \models^\psi \beta$ , and  
 (2) at least an  $\alpha_i$  has been retracted ( $1 \leq i \leq n$ ), and  
 (3)  $\beta$  has not yet been retracted,  
 then retract  $\beta$ .

On the other hand, a retracted belief may be a conclusion derived from other beliefs. The action to retract its premises will depend on whether the inference rule is deduction or not. If the retracted belief is deductively derived, then at least one of its premises should be retracted so that the retracted belief cannot again be derived any more at this moment. For exploring all kinds of possibilities, each of the premises will be respectively retracted for generating APL alternatives. On the contrary, if the retracted belief is abductively derived, then nothing needs to be done. In other words, the abductively derived statement seems a bad conjecture, while all of the premises can remain. This action can be represented as the following rule *premise retraction (R)*. It is noted that the third antecedent (i.e., ‘no  $\alpha_i$  has been retracted’) is for avoiding iterative retractions to  $\alpha_i$ .

**Rule 3. (Premise retraction)**

If (1) there exists a derivation  $\{\alpha_1, \alpha_2, \dots, \alpha_n\} \models^\delta \beta$ ,

(2)  $\beta$  has been retracted, and

(3) no  $\alpha_i$  has been retracted ( $1 \leq i \leq n$ ),

then retract  $\alpha_i$ .

The power of the three retraction rules *R1*, *R2*, and *R3* can be demonstrated as follows. It is supposed that there is a reasoning process as shown in Figure 5. Beliefs  $p$  and  $q \rightarrow p$  abductively derive  $q$ ; accordingly, the beliefs  $q$  and  $q \rightarrow r$  deductively derive  $r$ . On the other hand, the beliefs  $p$  and  $p \rightarrow \neg q$  deductively derive  $\neg q$ , which contradicts the existing belief  $q$ .

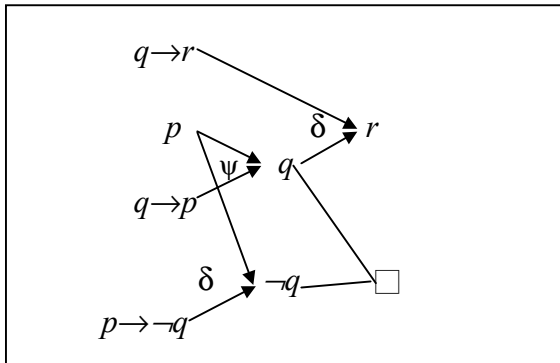


Figure 5. An example for retraction rules.

How APL uses the retraction rules to deal with this inconsistency is displayed in Figure 6. The retracted beliefs are marked with the double deletions, ‘~~—~~’, and are pointed by the dot-line arrows, ‘ $\dashrightarrow$ ’. The retraction rule used is marked at the root of each arrow. As depicted, three different cases result from the chain-retraction. The set consisting of all accepted beliefs in case 1,  $\{q \rightarrow r, p, q \rightarrow p, p \rightarrow \neg q, \neg q\}$ , and that in the case 3,  $\{q \rightarrow r, p, q \rightarrow p, q, r\}$ , are APL alternatives. By definition, the set consisting of all accepted beliefs in case 2,  $\{q \rightarrow r, q \rightarrow p, p \rightarrow \neg q\}$ , is not an APL

alternative, because it is a subset of the APL alternative in the case 1.

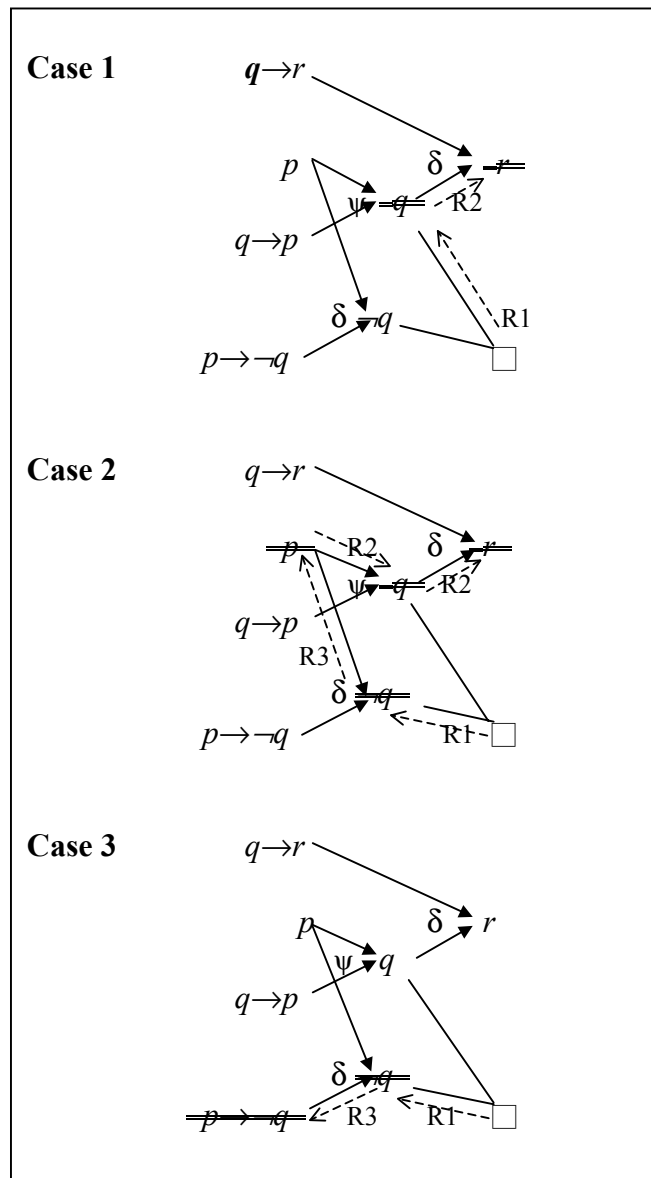


Figure 6. Three alternatives after a chain-retraction.

APL uses a *labeling table* to store the status of the beliefs of all APL alternatives, including the derivation types and the retractions, as shown in Table 1. In a labeling table, each deductively true belief, abductively possible belief, and assumptively true belief are respectively labeled  $\delta$ ,  $\psi$ , and  $o$ , as displayed in the column under ‘Derivations’. Still, each accepted belief is labeled ‘+,’ while each retracted belief is labeled ‘-,’ as displayed in the column under ‘Alternatives.’ Consequently, the

status of the beliefs in case 1 and 3 are respectively labeled in sub-column 1 and 2 under ‘Alternatives.’

A labeling table is not only useful for APL to identify APL alternatives, but also provides useful information for choosing alternatives. A reasoner can create an evaluation method based on the derivation types to choose a good solution among numerous APL alternatives. For example, he may think a suitable design alternative should contain some beliefs derived from the premises. Thus, he will think the APL alternatives without any deductively true belief (labeled  $\delta$ ) or abductively possible belief (labeled  $\psi$ ) are not suitable. Moreover, he can continue to determine the feasibility by the ratio of the deductively true beliefs and abductively possible beliefs of each APL alternative. Consider another example. A reasoner who has concerns that beliefs cannot exist for each APL alternative can examine the beliefs labeled ‘-’ in the labeling table. However, how to weight APL alternatives is not the objective of this study, though APL offers useful information. That is to say, the way in which APL identifies APL alternatives is based on the ‘form’ of reasoning, instead of the ‘meaning’ of beliefs.

Table 1. A labeling table.

<b>LABELING TABLE</b>			
Beliefs	Derivations	APL Alternatives	
		1	2
$q \rightarrow r$	$\delta$	+	+
$P$	$\delta$	+	+
$q \rightarrow p$	$\delta$	+	+
$q$	$\psi$	-	+
$r$	$\delta$	-	+
$p \rightarrow \neg q$	$\delta$	+	-
$\neg q$	$\delta$	+	-

#### IV. AN EXAMPLE: A SCOOTER DESIGN

This section will introduce the gold metal winner of ‘*The Fourth Student’s Scooter Design Contest for the Future 1999*,’ to demonstrate the power of APL. The annual contest, sponsored by *San Yang Motor Company (SYM)*, has become the most significant award for conceptual design in Taiwan since 1996. The objective of the contest is to encourage college students in Taiwan to stretch their imaginations about two-wheel vehicles in the future. More than ten graduate-design teams from various colleges were invited to join this contest in 1999. These works were reviewed by a jury consisting of five experts from Italy, Japan, and Taiwan, and were displayed in ‘*The 1999 Young Designer Show*,’ one of the most important intercollegiate design exhibitions in Taiwan.

The winning team consisted of Mr. Ching-Chan Yeh, Mr. Bou-Sheng Hwang, and Mr. Cheng-Hsing Lou, directed by Dr. Chen-Lian Cheng and Mr. Chiun-Ming Lian, of *Commodity Design Department of Ming Chuan University*. The major concept of their work was that the user could adjust some components to change the scooter’s ‘*facial expression*’ for various demands; thus, the nickname of this scooter was ‘*Transformer*.’ Figure 7 illustrates a rendering of this work. To analyze the designers’ reasoning, a forty-minute interview with Mr. Ching-Chan Yeh, the representative of the design team, was audio-taped. The results of the interview, focusing on how the objectives, ideas, and evaluations were proposed or rejected, were then represented in APL.

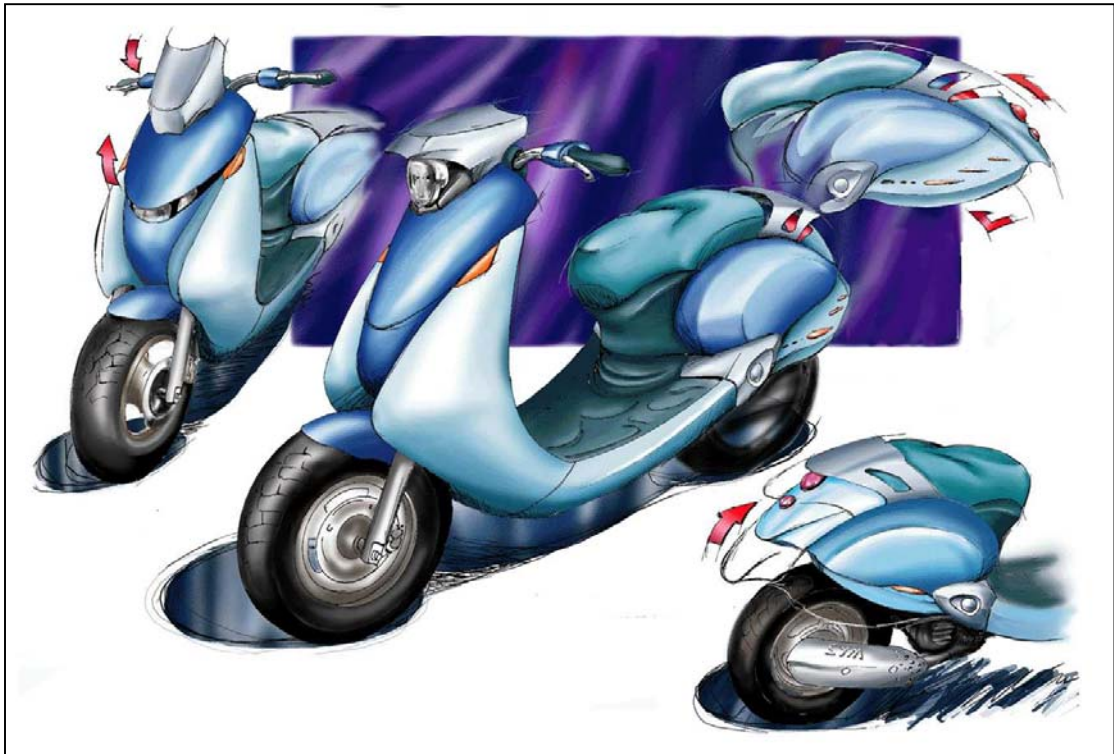


Figure 7. A rendering of the *Transformer* (courtesy of Ching-Chan Yeh, Bou-Sheng Hwang, and Cheng-Hsing Lou).

For convenience of description, the process of the designers' reasoning was divided into four stages with respect to the occurrence of inconsistency. Once an inconsistency was detected, all of the beliefs, excluding those of other stages, are identified as the beliefs of a new single stage. In the first stage, the designers borrowed some elegant curves from a collection of some existing scooter pictures, and used these curves as the geometric primitives to create new forms of the scooter. However, the directors thought that excessive use of these curves with modifications, adaptations, and combinations was extremely boring, and would not be useful in winning the contest. Thus, the directors asked the designers to restart their work, and follow two disciplines: (a) a unique '*theme*' or '*content*' of the design should be highlighted, (b) the doctrine "form follows function" should be obeyed. The

designers recalled that they ‘gave up their own design methods and ideas‘ at the end of this stage.

In the second stage, the designers found that each type of scooters had only one ‘*facial expression*,’ and it would be a good idea to design a multiple-expression scooter. Thus, the idea was first realized by mixing (1) the bold and uninhibited expression of off-road scooters with (2) the speedy and urban expression of on-road scooters. Later, the designers found they were unable to integrate the structure of an off-road scooter with that of an on-road scooter. It implied that they were incapable of building a scooter prototype for demonstrating the design theme; therefore, it was useless for winning the contest. Hence, the design idea was modified.

In the third stage, the designer tried to interchange the expressions of two types of scooters that had similar structures. The focus of the design was aimed to replace an on-road scooter’s expression with a leisure scooter’s carefree and suburban expression. To have an on-road scooter’s expression, four actions had to be made: (1) uncovering a higher headlight, (2) covering a lower headlight, (3) disassembling lower-tail shells, and (4) assembling high-tail shells. On the contrary, to have a leisure scooter’s expression, another four actions had to be made: (1) covering the higher headlight, (2) uncovering the lower headlight, (3) assembling the lower-tail shells, and (4) disassembling the high-tail shells. However, the tail shells of the under-designed scooter were so bulky that they had to be assembled and disassembled in a place such as a garage. This work limited the possibility that expressions could be dynamically interchanged at any time. Consequently, the designers decided to improve the shell design.



In the final stage, the designers proposed a built-in scooter shell, which was separated into three adjustable components: a rear fender and two rear frame-covers. The handle of the rear fender, in the form of a stabilizer, could be used as the passenger's backrest. On the other hand, the adaptation of the two rear frame-covers seemed to be like a beetle's opening and closing wings. These components were pulled up to represent the expression of an on-road scooter; they were pushed down to represent the expression of a leisure scooter. At this moment, the conceptual design process terminated because the contest time was running out.

## V. REPRESENTING REASONING USING APL

With respect to design reasoning, APL has two roles: a descriptor and a reasoner. As a descriptor, APL can be employed to represent the human designers' mental reasoning process of the illustrative scooter design in a symbolic way. A simplified derivation diagram of the mental reasoning is displayed in Figure 8. The beliefs and the interpretations of their *atoms* in Figure 8 are given in the Appendix. The designers' solutions at stages 1, 2, 3 and 4 can be respectively symbolized as the sets  $S_1$ ,  $S_2$ ,  $S_3$  and  $S_4$ , as follows.

$$S_1 = \{a, c \rightarrow \neg d \wedge \neg e, a \rightarrow d \wedge e, d \wedge e\}$$

$$S_2 = \{a, c \rightarrow \neg d \wedge \neg e, a \rightarrow d \wedge e, d \wedge e, f \rightarrow d, g \rightarrow f, i \rightarrow e, j \rightarrow i, h \rightarrow k, k \rightarrow l, l \rightarrow \neg a, f, i, g, j\}$$

$$S_3 = \{a, c \rightarrow \neg d \wedge \neg e, a \rightarrow d \wedge e, d \wedge e, f \rightarrow d, g \rightarrow f, i \rightarrow e, j \rightarrow i, h \rightarrow k, k \rightarrow l, l \rightarrow \neg a, f, i, g, j, \\ m \rightarrow g, n \wedge o \rightarrow m, p \rightarrow n, q \rightarrow r, r \rightarrow \neg e, m, n \wedge o, p\}$$

$$S_4 = \{a, c \rightarrow \neg d \wedge \neg e, a \rightarrow d \wedge e, d \wedge e, f \rightarrow d, g \rightarrow f, i \rightarrow e, j \rightarrow i, h \rightarrow k, k \rightarrow l, l \rightarrow \neg a, f, i, g, j, \\ m \rightarrow g, n \wedge o \rightarrow m, p \rightarrow n, q \rightarrow r, r \rightarrow \neg e, m, n \wedge o, p, s \rightarrow o, t \wedge u \rightarrow b \wedge s, a \rightarrow b, s, t \wedge u\}$$

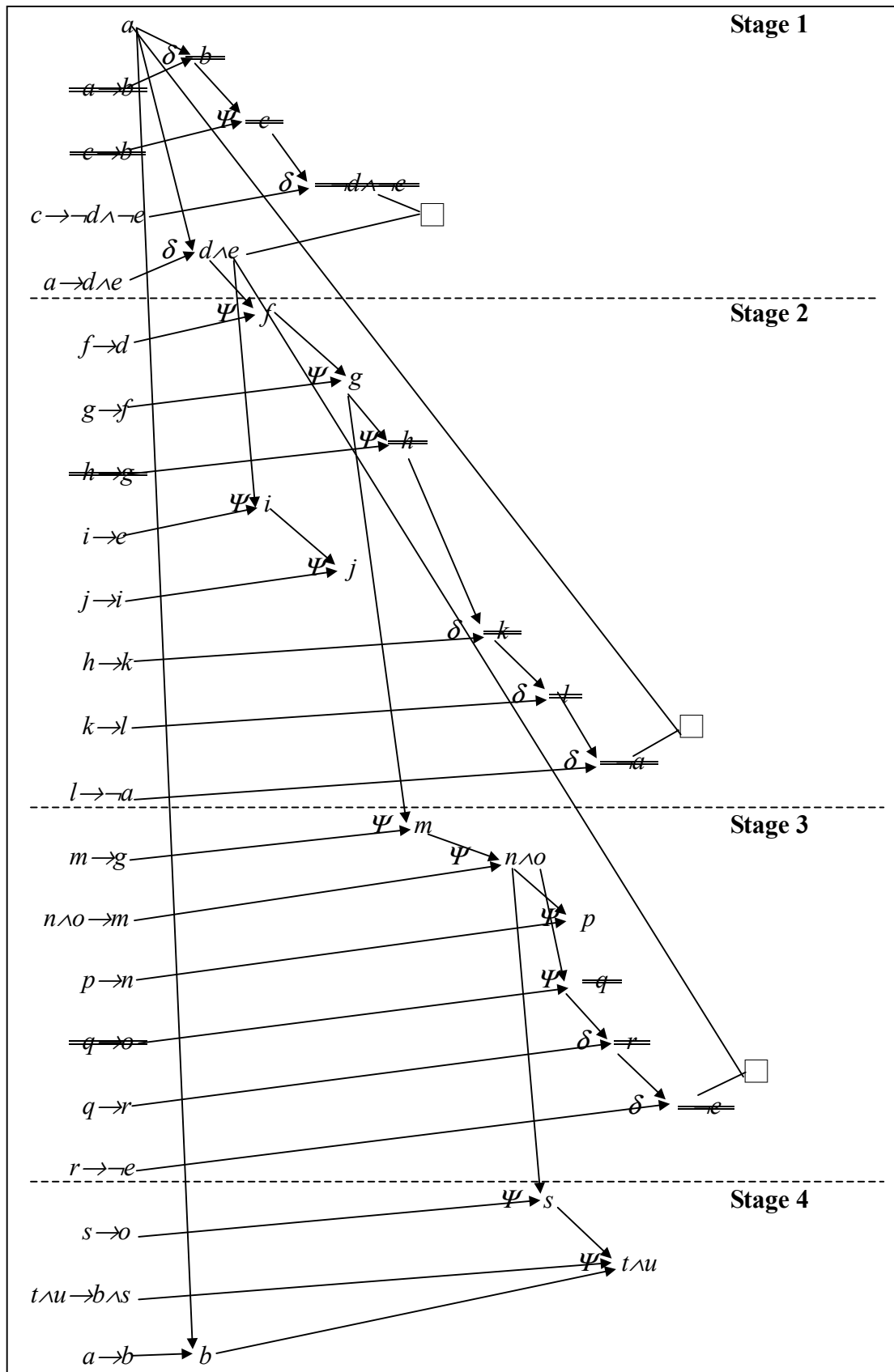


Figure 8. The derivation diagram of the human designers' mental reasoning (symbols

of the arrows and retraction rules are omitted).

On the other hand, to be a reasoner, APL may keep the beliefs used in the illustrative case but employ the two inference rules: deduction and abduction, as well as the three types of retraction: exclusive, conclusive and premise retractions. As a result, APL can generate 4 APL alternatives in stage 1, and 20 APL alternatives in stage 2. Furthermore, the number of APL alternatives is even increased to 41 in stages 3 and 4. The labeling table corresponding to the APL alternatives in stage 4 is shown in Table 2. The derivation diagram in Figure 9, for instance, illustrates the reasoning of the third APL alternative in Table 2. This APL alternative is symbolized as the set  $A_3$ , as follows. The set  $A_3$  includes the designers' solution at Stage 4, i.e.,  $S_4$ , and the set difference of  $A_3$  and  $S_4$  is the set  $\{c \rightarrow b, b, h \rightarrow g, q \rightarrow o\}$ .

$$A_3 = \{a, a \rightarrow b, c \rightarrow b, c \rightarrow \neg d \wedge \neg e, a \rightarrow d \wedge e, b, d \wedge e, f \rightarrow d, g \rightarrow f, h \rightarrow g, i \rightarrow e, j \rightarrow i, h \rightarrow k, k \rightarrow l, l \rightarrow \neg a, f, i, g, j, m \rightarrow g, n \wedge o \rightarrow m, p \rightarrow n, q \rightarrow o, q \rightarrow r, r \rightarrow \neg e, m, n \wedge o, p, s \rightarrow o, t \wedge u \rightarrow b \wedge s, a \rightarrow b, s, t \wedge u\}$$

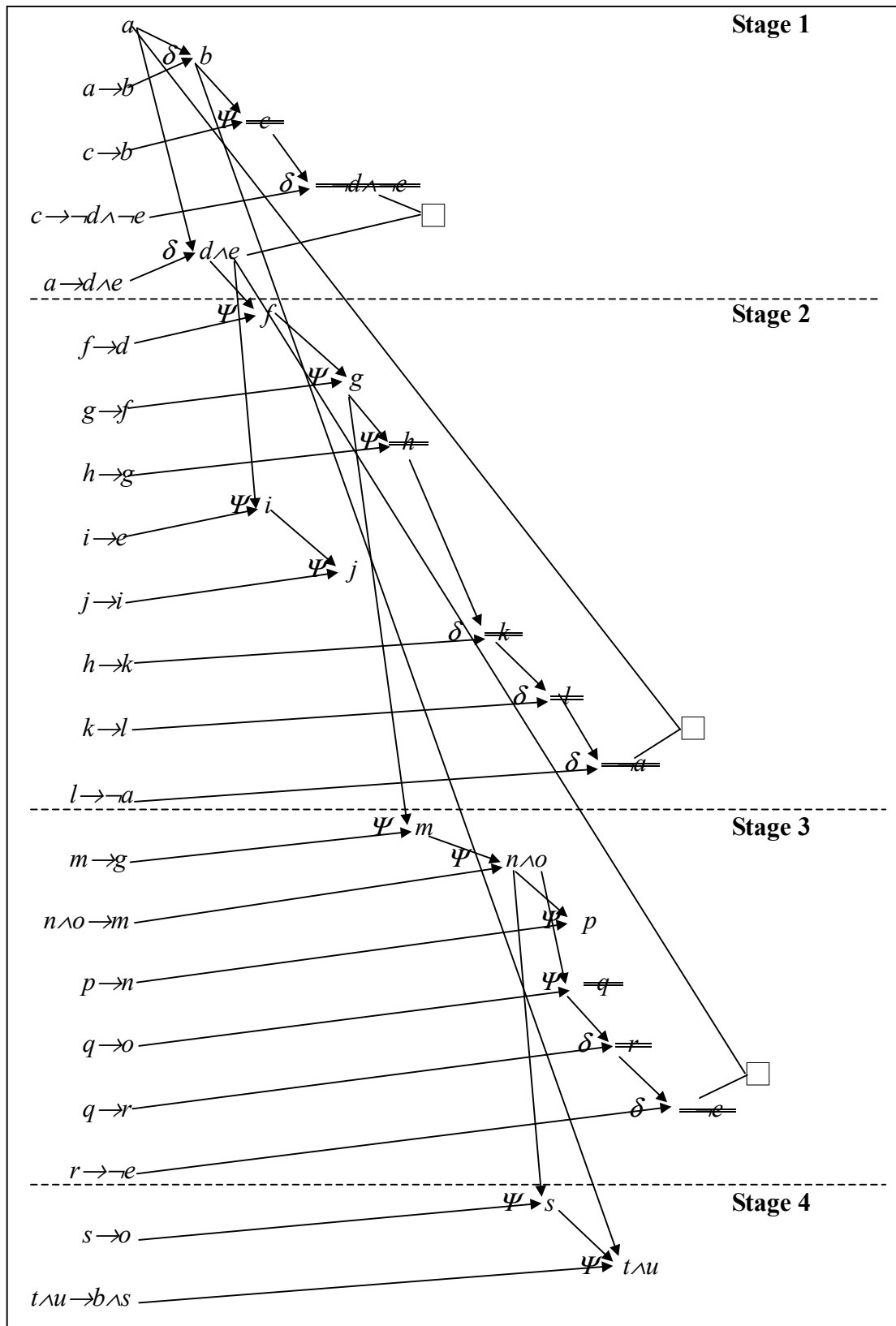


Figure 9. The derivation diagram of the third APL alternatives in Table 2 (symbols of

the arrows and retraction rules are omitted).

Table 2. Labeling table of APL alternatives in stage 4.

		LABELING TABLE																																																					
Beliefs	Derivations	APL Alternatives																																																					
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41													
a	$o$	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+											
a→b	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+								
c→b	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+							
c→¬d∧¬e	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+							
a→d∧e	$o$	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+							
b	$\delta$	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+							
d∧e	$\delta$	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+						
c	$\psi$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-							
¬d∧¬e	$\delta$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-					
f→d	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+					
g→f	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+				
h→g	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+			
i→e	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+			
j→i	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+			
h→k	$o$	+	+	+	+	+	-	-	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+			
k→l	$o$	+	+	+	+	+	+	+	+	+	+	-	-	-	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+			
l→¬a	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+			
f	$\psi$	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+				
i	$\psi$	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+				
g	$\psi$	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+				
j	$\psi$	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+				
h	$\psi$	-	-	-	-	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+					
k	$\delta$	-	-	-	-	-	-	-	-	-	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+					
l	$\delta$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
¬a	$\delta$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-				
m→g	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+			
n∧o→m	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+		
p→n	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+		
q→o	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+		
q→r	$o$	+	+	+	-	+	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+		
r→¬e	$\delta$	+	+	+	+	-	+	+	+	+	+	-	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+		
m	$\psi$	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+				
n∧o	$\psi$	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+				
p	$\psi$	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+				
q	$\psi$	-	-	-	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+				
r	$\delta$	-	-	-	-	+	-	-	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	
¬e	$\delta$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		
s→o	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
t∧u→b∧s	$o$	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
s	$\psi$	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+
t∧u	$\psi$	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+	+	-	-	+	+

Two interesting findings concerning design theory can be explored by comparing the mental reasoning (in Figure 8) and the logical reasoning (in Figure 9). First, the degree of using knowledge (or beliefs) is different between mental reasoning and logical reasoning. APL will try to maintain as many beliefs as possible in an APL alternative and store the information about the derivation types and retractions of each belief, but this is obviously not the human designers' method. In the illustrative case, the designers and directors did not take advantage of all the existing beliefs throughout the four stages. For example, the directors did not ask the designers to accept beliefs  $c \rightarrow \neg d \wedge \neg e$  and  $a \rightarrow d \wedge e$  until the designers obtained belief  $c$ . A possible explanation of this phenomenon is that human beings can not take into account too many things. Thus, even if both the designers and the directors had sufficient knowledge about the scooter design, they had to focus their attention on limited issues and temporarily ignore the others. Another explanation is that the directors intentionally played the role of evaluators, rather than co-designers. The directors would not give the designers any beliefs until they thought the designers had lost their way. Still, a third explanation could be that some design knowledge was so implicit that the designers could not express it. In this sense, the mental reasoning represented a version of the interviewer's personal interpretation of the designers' reasoning process.

Second, the human designers and APL used retraction in different ways. In the interview, for example, the designers confessed their initial proposals (beliefs)  $a \rightarrow b$ ,  $c \rightarrow b$ , and  $c$  in stage 1 had been rejected by their directors. In fact, the designers added the retracted belief  $a \rightarrow b$  into their reasoning in stage 4. Thus, they could

derive the retracted belief  $b$  again, and then, the new belief  $t \wedge u$ . In contrast, APL tends to retract as few beliefs as possible and to search for as many APL alternatives as possible, whereas the status of each belief is kept in the labeling table. In stage 1, APL did not simply retract the beliefs  $a \rightarrow b$ ,  $c \rightarrow b$  and  $c$ . Instead, APL generated four APL alternatives, symbolized as the sets  $P_1$ ,  $P_2$ ,  $P_3$  and  $P_4$ , as below. In the final stage, APL generated 41 APL alternatives, as shown in Table 2.

$$P_1 = \{a, a \rightarrow b, c \rightarrow b, c \rightarrow \neg d \wedge \neg e, a \rightarrow d \wedge e, b, d \wedge e\}$$

$$P_2 = \{a, a \rightarrow b, c \rightarrow b, a \rightarrow d \wedge e, b, d \wedge e, c\}$$

$$P_3 = \{a, a \rightarrow b, c \rightarrow b, c \rightarrow \neg d \wedge \neg e, b, c, \neg d \wedge \neg e\}$$

$$P_4 = \{a \rightarrow b, c \rightarrow b, a \rightarrow d \wedge e, c \rightarrow \neg d \wedge \neg e, b, c, \neg d \wedge \neg e\}$$

Obviously, the designers' solution  $S_1$  ( $=\{a, c \rightarrow \neg d \wedge \neg e, a \rightarrow d \wedge e, d \wedge e\}$ ) is included into the APL alternative  $P_1$ . Similarly, the designers' solution at each stage is included into some APL alternative(s) generated at the same stage. Whether any one of the APL alternatives is a suitable design solution or no, however, depends on how to define a complete design solution and how to evaluate or interpret it. Although it is possible to define the completeness of a design solution and to devise a judgement mechanism (i.e., how the human designers determine a complete solution), such work is beyond the scope of this paper. The major reason is that APL is simply a reasoning method suggesting possible solutions that the human designer could derive or should have derived, instead of an expert system to generate the very solution of the human designers.

Nevertheless, in this paper the completeness of a design solution can be discussed in terms of the form of reasoning. For example, the characteristics of the

design solutions  $S_1$ ,  $S_2$ ,  $S_3$  and  $S_4$ , as previously described at the beginning of this chapter, are twofold. First, all beliefs of each design solution are consistent. Second, there exists some derivation(s) in each design solution; i.e., there exist both believed facts and believed rules as the premise, and some beliefs derived from the premise. The second characteristic is not found in the definition of an APL alternative, as described in Chapter 3. In a word, any APL alternative is not necessarily a design solution.

In addition to the evaluation mechanisms for each APL alternative, as previously described, some interpretation mechanisms are needed to determine whether an APL alternative is a meaningful design solution or not. For example, the APL alternative  $P_2$  may be interpreted as meaning that the designers maintained their belief (about idea)  $c$  and avoided the belief (about performance)  $\neg d \wedge \neg e$  by rejecting the directors' belief (about evaluation)  $c \rightarrow \neg d \wedge \neg e$ . Still, the APL alternative  $P_3$  might correspond to another possibility that the designers avoided the belief (about goal)  $d \wedge e$  by rejecting the belief (about goal)  $a \rightarrow d \wedge e$  given by the directors. Thus, the designers maintained the belief (about performance)  $\neg d \wedge \neg e$ .

## VI. CONCLUSIONS

APL is a simple but powerful logic system for studying reasoning processes at the stage of conceptual design. Using a well-formed propositional language to represent designers' reasoning processes, APL owns the capabilities of deduction, abduction and inconsistency tolerance, so that it can emulate non-monotonicity occurring in a design process. In the illustrated example, APL is employed in a



formal way to reveal derivation relationships among beliefs, to discover potential alternatives, and to analyze interactive patterns between designers and directors. Consequently, it is found that APL can help to discover possible design solutions that are missed or ignored by human designers. Beyond the primitive findings in this paper, many future applications are expected: (1) developing methods to interpret and evaluate APL alternatives from a bunch of exhaustively generated ones, (2) developing methods for analyzing how the human designer reasons, (3) expanding APL to be a formal language with the capability of inductive reasoning, (4) expanding APL to be a system using first order logic. To sum up, this study suggests that the applications of APL have potential to improve the current theories of design thinking.

## REFERENCES

1. Baral, C. and Gelfond, M., 1994, "Logic Programming and Knowledge Representation," *The Journal of Logic Programming*, Vol. 19, pp. 73-148.
2. Baxter, M., 1995, *Product Design: A practical guide to systematic methods of new product development*, Chapman & Hall, London, UK.
3. Cadoli, M. and Schaerf, M., 1993, "A Survey of Complexity Results for Non-monotonic Logics," *The Journal of Logic Programming*, Vol. 17, pp. 127-160.
4. Coyne, R., 1988, *Logic Models in Design*, Pitman Publishing, London, UK.
5. de Bono, E., 1970, *Lateral Thinking: Creativity step by step*, Harper & Row, New York, NY.
6. de Kleer, J., 1986, An Assumption-based TMS. *Artificial Intelligence*, Vol. 28, pp. 127-162.

7. Doyle, J., 1979, "A Truth Maintenance System." *Artificial Intelligence*, Vol. 12, pp. 231-272.
8. Esghi, K. and Kowalski, R. A., 1989, "Abduction Compared with Negation by Failure," in *Proceedings of the sixth International Conference on Logic Programming 1989*, pp. 234-254.
9. Gabbay, D. M., 1996, *Labelled Deductive Systems*, Oxford University Press Inc., New York, NY.
10. Galle, P., 1996, "Design Rationalization and the Logic of Design: An Alternative Study," *Design Studies*, Vol. 17, pp. 235-275.
11. Galle, P., 1997, "Towards a Formal Logic of Design Rationalization," *Design Studies*, Vol. 18, pp. 195-219.
12. Genesereth, M. and Nilsson, N., 1987, *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, Los Altos, CA.
13. Giarratano, J. C. and Riley, G., 1994, *Expert Systems: Principles and Programming*, PWS Publishing, Boston, MA.
14. Ginsberg, M., 1993, *Essentials of Artificial Intelligence*, Morgan Kaufman Publishers, San Mateo, CA.
15. Luger, G., 1993, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, The Benjamin/Cummings Publishing Company, Inc., CA, 1993.
16. March, L., 1984, "The Logic of Design," in *Developments in Design Methodology*, N. Cross (ed), John Wiley & Sons, New York, pp. 265-276.
17. McCarthy, J., 1980, "Circumscription- a form of non-monotonic reasoning," *Artificial Intelligence*," Vol. 13, pp. 27-39.
18. McDermott, D. and Doyle, J., 1980, "Non-monotonic Logic I," *Artificial Intelligence*, Vol. 13, pp. 41-72.

19. Mitchell, W. J., 1990a, Introduction, *The Electronic Design Studio: Architectural education in the computer era*, McCullough, M., Mitchell, W. J. and Purcell, P. (eds), MIT Press, Cambridge, MA.
20. Mitchell, W. J., 1990b, *The Logic of Architecture*, MIT Press, Cambridge, MA.
21. Moore, R., 1985 “Semantics Considerations on Non-monotonics Logics,” *Artificial Intelligence*, Vol. 25, pp. 75-94.
22. Pereira, L. M. et al., 1993, “Non-monotonic Reasoning with Logic Programming,” *The Journal of Logic Programming*, Vol. 17, pp. 227-263.
23. Reiter, R., 1980, “A Logic for Default Reasoning,” *Artificial Intelligence*, Vol. 13, pp. 81-132.
24. Rich, E. and Knight, K., 1991, *Artificial Intelligence*, McGraw-Hill, New York, NY.
25. Roozenburg, N. F. M., 1993, “On the Pattern of Reasoning in Innovative Design,” *Design Studies*, Vol. 14, pp. 4-19.
26. Roozenburg, N. F. M. and Eekels, J., 1995, *Product Design: Fundamentals and Methods*, John Wiley & Sons, Inc., New York, NY.
27. Rowe, P., 1994, *Design Thinking*, MIT Press, Cambridge, MA.
28. Ulrich, K. T. and Eppinger, S. D., 1995, *Product Design and Development*, McGraw-Hill, Inc., New York, NY.
29. Wang, H., 1995, “An Approach to Computer-aided Styling,” *Design Studies*, Vol. 16, pp. 50-61.
30. Wang, H., 1996, “Testing with multimedia,” *Design Management Journal*, Fall, pp. 77-81.

## Appendix: Interpretations of the atoms in Figure 8.

- a*: *W* is the winner in the contest.
- b*: *W* is an elegant scooter.
- c*: *W* has elegant curves appearing on existing scooters.
- d*: The design of *W* has a unique theme.
- e*: The form of *W* follows its unique functions.
- f*: *W* mixes two expressions.
- g*: *W* mixes two forms.
- h*: *W* mixes forms of off-road and on-road scooters.
- i*: *W* can avoid the pollution of the nozzle's leaking gasoline and the entanglement of hose during refueling.
- j*: *W* has one refuel hole on the left-lower side and another on the right-lower side.
- k*: *W* cannot integrate the structures of two types of scooters.
- l*: There is no prototype of *W* for demonstrating the design theme.
- m*: *W* mixes a leisure scooter's form and an on-road scooter's form.
- n*: *W* mixes a leisure scooter's round headlight of and an on-road scooter's strip headlight.
- o*: *W* mixes a leisure scooter's low tail and an on-road scooter's high tail.
- p*: At the same time, *W*'s higher headlight can be uncovered and *W*'s lower headlight can be covered for the leisure scooter; or *W*'s higher headlight can be covered and *W*'s lower headlight can be uncovered for the on-road scooter.
- q*: To be an on-road scooter, *W*'s lower-tail shell can be disassembled, and *W*'s high-tail shell can be assembled; also, to be a leisure scooter, *W*'s low-tail shell can be assembled, and *W*'s higher-tail shell can be disassemble.
- r*: It is inconvenient to assemble and disassemble *W*'s shells.

*s*: *W* is designed to push down the rear fender and rear frame-covers for the leisure scooter; contrariwise, pull up the rear fender and rear frame-covers for the on-road scooter.

*t*: The handle of *W*'s rear fender looks like a stabilizer and performs like the passenger's backrest.

*u*: The adaptation of *W*'s two rear frame-covers seems to be like the beetle's opening and closing its wings.