Assumption–Based Reasoning and Probabilistic Argumentation Systems*

J. Kohlas and R. Haenni
Institute of Informatics
University of Fribourg
Regina Mundi
CH–1700 Fribourg
Switzerland

Phone: (+41 37) 29 83 20
Fax: (+41 37) 29 97 26
E–Mail: juerg.kohlas@unifr.ch

March 1996

Abstract

Propositional assumption–based systems provide for a convenient way to represent uncertainty in both a logical and probabilistic framework. Assumption–based reasoning permits to develop arguments supporting hypotheses. The methods are based on logical deduction and theorem proving. In addition, it is shown that also Shenoy's valuation networks are useful for computing supporting arguments. This forms a symbolic evidence theory. Once probabilities are assigned to assumptions, degrees of support defined as probabilities of provability can be defined. This leads then to probabilistic argumentation systems and thus to a numerical Dempster–Shafer theory of evidence. This chapter focuses on computational methods both for the symbolic or logical part of the assumption–based argumentation systems as well as for its probabilistic structure.

*Research supported by grants No. 21–30186.90 and 21–32660.91 of the Swiss National Foundation for Research, Esprit Basic Research Activity Project DRUMS II (Defeasible Reasoning and Uncertainty Management).

1 An Overview

Propositional logic language is an efficient and convenient way to encode knowledge and information. In particular, uncertainty can be incorporated into propositional knowledge and information by including assumptions which may or may not be true. Whether given hypotheses can be proved to be true in the light of the knowledge and the information depends on whether a sufficent number of assumptions can be assumed to be true. Assumption–based reasoning amounts then to derive arguments, under which given hypotheses can be deduced from the available knowledge and information. Probabilites may possibly be assigned to the assumptions occuring in the assumption–based knowledge. Then degrees of support (or credibility) may be associated to hypotheses according to the probability that they can be proved or deduced.

The determination of arguments supporting given hypotheses is essentially a task involving deduction and theorem proving. A variety of established methods and procedures exist for deduction in propositional logic. In particular, the methodology of ATMS (Assumption–Based Truth Maintenance Systems) provides the basic elements for assumption–based reasoning (de Kleer, 1986; Reiter & de Kleer, 1987a).

Once the arguments supporting a hypothesis are given and probabilites are assigned to the assumptions forming the arguments, the problem of computing the reliability of the arguments remains. This problem of reliability of reasoning with unreliably arguments is similar to the problem of the reliability of a technical system composed of unreliable components. Many techniques for computing these reliabilities are available and can be adapted to the problem of the determination of the degree of support of a hypothesis from its supporting arguments. It is well known that this probabilistic assumption–based reasoning is closely related to Dempster–Shafer Theory of Evidence (Laskey & Lehner, 1989; Provan, 1990).

This chapter develops the computational fundaments for this type of probabilistic argumentation systems. In Section 2 it starts with an informal introduction into assumption–based reasoning using a simple example. It shows how supporting arguments may be obtained and illustrates the probabilistic computations leading to the degrees of support. Then, in Section 3, it lays the fundaments of the methods by exposing the symbolic, logic version of evidence theory and the probabilistic structure built upon it.

In Section 4 computational methods for handling the symbolic or logical part of argumentation systems are presented. Various approaches based on resolution or related schemes are discussed. In addition methods based on decomposing the knowledge base and the use of Shenoy's valuation networks are introduced. This kind of approach is less known in logic. Finally, Section 5 covers computational methods for the probabilistic part of argumentation systems. It shows how degrees of support may be obtained from symbolic arguments once probabilties are assigned to the assumptions. Alternatively, it is pointed out that degrees of support may also be obtained by applying valuation network techniques to the usual numerical Dempster–Shafer theory of evidence (Lauritzen & Shenoy, 1996).

Probabilistic argumentation systems are related in this chapter exclusively to classical propositional logic. This provides a convenient computational context. However, it has been pointed out elsewhere that the concept can be seen in a much more general framework, related to arbitrary logics (Kohlas & Besnard, 1995) or to a general but basic algebraic structure called body of arguments (Kohlas, 1995).

2 An Informal Introduction

Assumption–based systems provide a powerful technique of dealing and reasoning with uncertain information or knowledge. This section presents an informal introduction to assumption–based reasoning. A small example based on a fictitous story illustrates the idea of the theory. A formal description of the theory is presented in Section 3.

The example considered here is a small story around the alarm system of Mr. Holmes' house (Pearl, 1988). A burglary in Mr. Holmes house generates an alarm if the alarm system is functioning. But the alarm may also be caused by an earthquake or by other (unspecified) reasons. The neigbours of Mr. Holmes, Mr. Watson and Mrs. Gibson, phone Mr. Holmes in the case of an alarm. Possibly, Mr. Watson may also phone Mr. Holmes as a joke. Mrs. Gibson is hard of hearing, and she may possibly not be able to hear the alarm. Furthermore, if Mr. Holmes' daughter is at home, then she surely will phone too in the case of an alarm. Finally, if there is an earthquake and if the earthquake is registered, then there is a confirmation of it on the radio.

To transform the information contained in this fictive story into an assumption–based knowledge, two disjoint sets of propositional symbols have to be introduced, $P = \{a, b, c, d, e, g, w\}$ and $A = \{a_1, a_2, a_3, a_4, a_5, a_5\}$. The elements of P are called propositions. They represent possible facts (e.g. “there is a phone call of Mr. Watso”) or statements about interesting questions (e.g. “is there a burglary in Mr. Holmes’ house ?”). In this example the symbols in P have to be interpreted as follows:

- a: the alarm system in Mr. Holmes’ house is ringing,
- b: there is a burglary,
- c: there is a confirmation of the earthquake on the radio,
- d: Mr. Holmes’ daughter phones,
- e: an earthquake has occured,
- g: Mrs. Gibson, a neighbour of Mr. Holmes, phones,
- w: Mr. Watson, another neighbour of Mr. Holmes, phones.

The elements of A are called assumptions. They are used to represent the uncertainty of the statements in which they appear. For example, a_1 represents the assumption that the alarm system is really functioning in the case of a burglary or an earthquake. Here, the symbols in A have the following meaning:

- a_1: the alarm system is functioning,
- a_2: other (unspecified) causes producing an alarm are present,
- a_3: Mr. Watson is in a joking mood,
- a_4: Mrs. Gibson is able to hear the alarm,
- a_5: Mr. Holmes’ daughter is at home,
- a_6: the earthquake is registered.

Now, the knowledge given in the short story can be transformed into a set of statements or clauses over the set of all available symbols $L = P \cup A$. Here, the logical connectives $\sim$, $\wedge$, $\vee$ and $\rightarrow$ are needed to build statements of the form $\ell_1 \wedge \ldots \wedge \ell_q \rightarrow \ell_{q+1} \vee \ldots \vee \ell_r$, where the ℓ_i’s are literals of N. Such an implication is equivalent to a clause ${\sim}\ell_1 \vee \ldots \vee {\sim}\ell_q \vee \ell_{q+1} \vee \ldots \vee \ell_r$. The following statements $\xi_1, \ldots, \xi_{17}$ represent the given information about the alarm system. The corresponding clauses are put in parentheses.

ξ_1: $b \wedge a_1 \rightarrow a$	$(\sim b \vee \sim a_1 \vee a)$	ξ_{10}: $g \rightarrow a$	$(\sim g \vee a)$
ξ_2: $e \wedge a_1 \rightarrow a$	$(\sim e \vee \sim a_1 \vee a)$	ξ_{11}: $g \rightarrow a_4$	$(\sim g \vee a_4)$
ξ_3: $a_2 \rightarrow a$	$(\sim a_2 \vee a)$	ξ_{12}: $a \wedge a_5 \rightarrow d$	$(\sim a \vee \sim a_5 \vee d)$
ξ_4: $a \rightarrow b \vee e \vee a_2$	$(\sim a \vee b \vee e \vee a_2)$	ξ_{13}: $d \rightarrow a$	$(\sim d \vee a)$
ξ_5: $a \rightarrow a_1 \vee a_2$	$(\sim a \vee a_1 \vee a_2)$	ξ_{14}: $d \rightarrow a_5$	$(\sim d \vee a_5)$
ξ_6: $a \rightarrow w$	$(\sim a \vee w)$	ξ_{15}: $e \wedge a_6 \rightarrow c$	$(\sim e \vee \sim a_6 \vee c)$
ξ_7: $a_3 \rightarrow w$	$(\sim a_3 \vee w)$	ξ_{16}: $c \rightarrow e$	$(\sim c \vee e)$
ξ_8: $w \rightarrow a \vee a_3$	$(\sim w \vee a \vee a_3)$	ξ_{17}: $c \rightarrow a_6$	$(\sim c \vee a_6)$
ξ_9: $a \wedge a_4 \rightarrow g$	$(\sim a \vee \sim a_4 \vee g)$		

For example, the statements $\xi_1, \ldots, \xi_3$ describe the fact that a burglary (ξ_1), an earthquake (ξ_2) or some other reasons (ξ_3) can cause an alarm. Inversely, an alarm implies that there is either a burglary, an earthquake or another reason (ξ_4), and that the alarm system is functioning (ξ_5). The statements ξ_6 to ξ_{17} are obtained from the given information in a similar way.

The clauses ξ_1 to ξ_{17} form a complete description of the story around the alarm system. This description has to be extended if additional facts arrive. For example, if there is a phone call of Mr. Watson and there are no phone calls of Mrs. Gibson and of Mr. Holmes' daughter, then ξ_{18} to ξ_{20} are the new statements to be added.

$$\xi_{18}: \quad w \qquad\qquad \xi_{19}: \quad \sim g \qquad\qquad \xi_{20}: \quad \sim d$$

Now, the description of story is complete. The set of statements or clauses is called the knowledge base Σ. Some of the clauses in Σ are subsuming others and can therefore be dropped. Here, ξ_6 and ξ_7 subsume ξ_{18}, ξ_{10} and ξ_{11} subsume ξ_{19}, and ξ_{13} and ξ_{14} subsume ξ_{20}. Finally, the knowledge base consists of 14 remaining clauses:

$$\Sigma = \Sigma_{\{a,b,c,d,e,g,w\}} = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_8, \xi_9, \xi_{12}, \xi_{15}, \xi_{16}, \xi_{17}, \xi_{18}, \xi_{19}, \xi_{20}\}. \tag{2.1}$$

After the phone call of Mr. Watson, suppose that Mr. Holmes is seriously concerned. He worries that there was a burglary. In order to be reassured Mr. Holmes tries to find arguments that there was no burglary and that everything is all right. This is the interesting question or hypothesis which is represented by $\sim b$. An argument in favour of $\sim b$ is a conjunction $\ell_1 \wedge \ldots \wedge \ell_q$ of literals of assumptions (e.g. $a_1 \wedge \sim a_2 \wedge a_3$), such that $\{\ell_1, \ldots, \ell_q\} \cup \Sigma$ entails $\sim b$. To find such arguments the propositions in $P - \{b\} = \{a, c, d, e, g, w\}$ are not really needed and they can be eliminated from Σ.

Eliminating a variable p from Σ means: (1) eliminating from Σ all clauses containing the symbol p, (2) finding all possible resolutions of the eliminated clauses with respect to p, (3) adding the obtained resolvents to Σ, (4) dropping subsuming clauses.

The symbols in $\{a, c, d, e, g, w\}$ can be eliminated from Σ in any order. To illustrate this idea, which will be justified later, suppose e to be the first variable to be eliminated from Σ. The symbol e appears in the clauses ξ_2, ξ_4, ξ_{15} and ξ_{16}. They can be eliminated from Σ. Two new clauses ξ_{21} and ξ_{22} are obtained by resolution:

$$\xi_{21} = \rho(\xi_2, \xi_{16}) = a \vee {\sim}c \vee {\sim}a_1, \tag{2.2}$$

$$\xi_{22} = \rho(\xi_4, \xi_{15}) = {\sim}a \vee b \vee c \vee a_2 \vee {\sim}a_6. \tag{2.3}$$

The new clauses are added to Σ. At the moment it is not possible to drop subsuming clauses. Thus, after eliminating the variable e the following knowledge base is obtained:

$$\Sigma_{\{a,b,c,d,g,w\}} = \{\xi_1, \xi_3, \xi_5, \xi_8, \xi_9, \xi_{12}, \xi_{17}, \xi_{18}, \xi_{19}, \xi_{20}, \xi_{21}, \xi_{22}\}. \tag{2.4}$$

The remaining variables to be eliminated are a, c, d, g and w. Applying the same procedure sequentially to all of them generates new clauses ξ_{23} to ξ_{32}. Finally, only the clauses ξ_{26} to ξ_{32} remain.

ξ_{26}:	${\sim}b \vee {\sim}a_1 \vee {\sim}a_4$	ξ_{30}:	$a_3 \vee {\sim}a_4$
ξ_{27}:	${\sim}b \vee {\sim}a_1 \vee {\sim}a_5$	ξ_{31}:	$a_3 \vee {\sim}a_5$
ξ_{28}:	${\sim}a_2 \vee {\sim}a_4$	ξ_{32}:	$a_1 \vee a_2 \vee a_3$
ξ_{29}:	${\sim}a_2 \vee {\sim}a_5$		

Note that the final knowledge base

$$\Sigma_{\{b\}} = \{\xi_{26}, \xi_{27}, \xi_{28}, \xi_{29}, \xi_{30}, \xi_{31}, \xi_{31}, \xi_{32}\}. \tag{2.5}$$

is much smaller than the original one.

Now, arguments for ${\sim}b$ may be found. Considering ξ_{26} it is clear that ${\sim}b$ is implied if $a_1 \wedge a_4$ is true. Thus, $a_1 \wedge a_4$ is an argument for ${\sim}b$. Similarly, another argument $a_1 \wedge a_5$ is found for ξ_{27}. The set of the arguments in favour of ${\sim}b$ is called the quasi–support of ${\sim}b$ relative to Σ. Furthermore, note that according to ξ_{28} the conjunction $a_2 \wedge a_4$ for example is not possible. $a_2 \wedge a_4$ is called a contradiction relative to the knowledge base Σ. Further contradictions $a_2 \wedge a_5$, ${\sim}a_3 \wedge a_4$, ${\sim}a_3 \wedge a_5$, and ${\sim}a_1 \wedge {\sim}a_2 \wedge {\sim}a_3$ are derived from ξ_{29} to ξ_{32}. The set of possible contradictions can be transformed into an equivalent logical formula:

$$\begin{aligned} qs(\perp, \Sigma) \quad = \quad & (a_2 \wedge a_4) \vee ({\sim}a_3 \wedge a_4) \vee \\ & (a_2 \wedge a_5) \vee ({\sim}a_3 \wedge a_5) \vee \\ & ({\sim}a_1 \wedge {\sim}a_2 \wedge {\sim}a_3). \end{aligned} \tag{2.6}$$

Using this result, the following logical formula is derived from the arguments $a_1 \wedge a_4$ and $a_1 \wedge a_5$ of $\sim b$. It represents the conditions under which $\sim b$ can be deduced from Σ avoiding contradictions (systematic methods to derive such a logical expression are presented later).

$$\begin{aligned} sp(\sim b, \Sigma) &= (a_1 \wedge \sim a_2 \wedge a_3 \wedge a_4) \vee \\ &\quad (a_1 \wedge \sim a_2 \wedge a_3 \wedge a_5). \end{aligned} \tag{2.7}$$

Thus, the hypothesis $\sim b$ is supported by two arguments $a_1 \wedge \sim a_2 \wedge a_3 \wedge a_4$ and $(a_1 \wedge \sim a_2 \wedge a_3 \wedge a_5$. The first argument for example can be interpreted as followed: there is certainly no burglary in Mr. Holmes' house, if the alarm system is functioning (a_1), if no other reason has caused an alarm ($\sim a_2$), if Mr. Watson's phone call is only a joke (a_3), and if in the case of an alarm Mrs. Gibson is able to hear it (a_4).

Possibly, Mr. Holmes is also interested to judge his question quantitatively. For that purpose he introduces subjective and independent probability values for the assumptions a_1 to a_6. These values are based on Mr. Holmes' previous experience and his knowledge about the alarm system, the neighbourhood, his daughter, and earthquakes. Suppose he chooses the values

$$\begin{array}{lll} p(a_1) = 0.8, & p(a_3) = 0.4, & p(a_5) = 0.3, \\ p(a_2) = 0.1, & p(a_4) = 0.5, & p(a_6) = 0.9. \end{array}$$

Now, the qualitative results in (2.7) can also be evaluated quantitatively. To obtain the probability of a logical expression, it has to be transformed into an equivalent disjunctive normal form with disjoint terms. This problem is a well known problem, especially in reliability theory (two methods are described in Section 5). For the support of $\sim b$ the following disjoint expression is generated:

$$\begin{aligned} sp(\sim b, \Sigma) &= (a_1 \wedge \sim a_2 \wedge a_3 \wedge a_4 \wedge \sim a_5) \vee \\ &\quad (a_1 \wedge \sim a_2 \wedge a_3 \wedge a_5). \end{aligned} \tag{2.8}$$

The probability of $sp(\sim b, \Sigma)$ is the sum the probailities of the disjoint terms in (2.8). The probability of a term is the product of the probabilities of its literals. With $p(\sim a_i) = 1 - p(a_i)$ the following result is obtained:

$$\begin{aligned} p(sp(\sim b, \Sigma)) &= 0.8 \cdot 0.9 \cdot 0.4 \cdot 0.5 \cdot 0.7 + 0.8 \cdot 0.9 \cdot 0.4 \cdot 0.3 \\ &= 0.1872. \end{aligned} \tag{2.9}$$

However, this probability has to be conditioned on the fact that no contradicting configuration of assumptions is possible. For that purpose, the probability $p(qs(\bot,\Sigma)) = 0.4538$ is determined in a similar way. To obtain the so–called degree of support of $\sim b$, the conditional probability given $qs(\bot,\Sigma)$ can then be computed:

$$\begin{aligned} sup(\sim b,\Sigma) &= p(sp(\sim b,\Sigma) \mid \sim qs(\bot,\Sigma)) = \frac{p(sp(\sim b,\Sigma))}{1-p(qs(\bot,\Sigma))} \\ &= \frac{0.1872}{1-0.4538} = 0.3427. \end{aligned} \tag{2.10}$$

Based on the subjective probability values, the hypothesis that there is no burglary in Mr. Holmes' house is only supported by a degree of $\approx\frac{1}{3}$. Thus, Mr. Holmes has to be seriously concerned.

3 Fundamental Notions

In this section assumption–based systems as informally introduced in the previous section will be formally defined and analyzed.

3.1 Symbolic Evidence Theory

Let A be a set of propositional symbols $\{a_1, a_2, \ldots, a_m\}$, called **assumptions**, P another set of propositional symbols $\{p_1, p_2, \ldots, p_n\}$, $L = A \cup P$ and $\Sigma = \{\xi_1, \xi_2, \ldots, \xi_r\}$ a set of well–formed propositional formulae (wff) over L. In many cases (but not in general) ξ_i are supposed to be clauses and Σ is always to be interpreted as a conjunction of its members, $\xi = \xi_1 \wedge \xi_2 \wedge \ldots \wedge \xi_r$. The triple (Σ, A, P) is called an **assumption–based knowledge**. It constitutes the basic model to be considered in the following.

We are interested in a certain **hypothesis**, expressed as a well–formed formula h over L. $\mathcal{L}_L$ denotes the set of all possible hypotheses. It is important to note that in the sequel **equivalent formulae** can be considered as describing the **same hypothesis**. What can we learn about h from Σ? If a sufficient number of assumptions can be assumed to be true, then h may be deducible from Σ. So it is interesting to ask under what assumptions h follows from Σ. The more reasonable, the more likely these assumptions are, the more credible is the hypothesis h in the light of the knowledge Σ. This may be quantified in a second stage by assigning probabilities to the assumptions and by computing the probability that h can in fact be deduced from Σ.

In a first stage, however, we study only symbolic (non–numeric) arguments for h. A **literal** is either a proposition p or its negation $\sim p$. Let $A^{\pm}$

be the set of all literals of A and $C_{A^\pm}$ be the family of all subsets of $A^\pm$ which do not contain pairs $\{a_i, \sim a_i\}$ of opposite literals. The elements a of $C_{A^\pm}$ will be interpreted as conjunctions of its members. They will be called **arguments**. The empty set $\emptyset$ is then interpreted as the tautology $\top$.

We are now interested in some subsets of $C_{A^\pm}$:

$$QS(h, \Sigma) = \{a \in C_{A^\pm} : a, \Sigma \models h\}, \tag{3.1}$$

$$SP(h, \Sigma) = \{a \in C_{A^\pm} : a, \Sigma \models h, \ a, \Sigma \not\models \bot\}. \tag{3.2}$$

$QS(h, \Sigma)$ is called **quasi–support** for h (given Σ), because its elements are sufficient, together with the knowledge Σ, to imply h. That is, the elements a of $QS(h, \Sigma)$ are arguments supporting h (given Σ). However, it is possible that a and Σ are not satisfiable, that is a and Σ are contradictory: $a, \Sigma \models \bot$. Such arguments a are clearly not proper for h, hence the term quasi–support for $QS(h, \Sigma)$.

$SP(h, \Sigma)$ contains all arguments for h which are not contradictory to Σ, i.e. all proper arguments for h. Therefore, $SP(h, \Sigma)$ is called the **support** for h (given Σ).

$QS(\bot, \Sigma)$, the quasi–support for the contradiction, contains exactly the arguments which are contradictory in Σ. Note that if we accept Σ as being true, then no $a \in QS(\bot, \Sigma)$ can be true (see also Subsection 3.3). Therefore, $QS(\bot, \Sigma)$ is called **contradiction** (relative to Σ). Clearly,

$$SP(h, \Sigma) = QS(h, \Sigma) - QS(\bot, \Sigma). \tag{3.3}$$

In some cases, it may also be interesting to consider two other subsets of $C_{A^\pm}$:

$$RF(h, \Sigma) = \{a \in C_{A^\pm} : a, \Sigma \models \sim h\} = \{a \in C_{A^\pm} : a, h, \Sigma \models \bot\} \tag{3.4}$$

$$PO(h, \Sigma) = \{a \in C_{A^\pm} : a, \Sigma \not\models \sim h\} = \{a \in C_{A^\pm} : a, h, \Sigma \not\models \bot\} \tag{3.5}$$

$RF(h, \Sigma)$ contains the arguments for $\sim h$, i.e. arguments against h. These are the arguments which refute h. Therefore, $RF(h, \Sigma)$ is called the **refutation** of h (given Σ). $PO(h, \Sigma)$ contains just all arguments which do not permit to deduce $\sim h$, hence the arguments for which h is possible given Σ, although not necessarily deducible from Σ. Therefore, $PO(h, \Sigma)$ is called the **possibility** of h (given Σ).

To illustrate the notions of quasi–support, contradiction and support, consider the assumption–based knowledge (Σ, A, P) with $A = \{a_1, a_2\}$, $P = \{p, q\}$ and $\Sigma = \{\xi_1, \xi_2, \xi_3\}$:

$$\xi_1\!: \ a_1 \rightarrow p \qquad (\sim\!a_1 \vee p) \qquad \xi_3\!: \ a_2 \rightarrow q \qquad (\sim\!a_2 \vee q)$$
$$\xi_2\!: \ a_2 \rightarrow \sim\!p \qquad (\sim\!a_2 \vee \sim\!p)$$

Clearly, a_1 is sufficient to deduce p from Σ. Therefore, any element of $C_{A^\pm}$ containing a_1 belongs to $QS(p, \Sigma)$, that is $QS(p, \Sigma) = \{a_1, a_1 \wedge a_2, a_1 \wedge \sim\!a_2\}$. Similarly, $QS(\sim\!p, \Sigma) = \{a_2, a_1 \wedge a_2, \sim\!a_1 \wedge a_2\}$ and $QS(\perp, \Sigma) = \{a_1 \wedge a_2\}$. These sets are shown in Figure 3.1 as subsets of $C_{A^\pm}$.

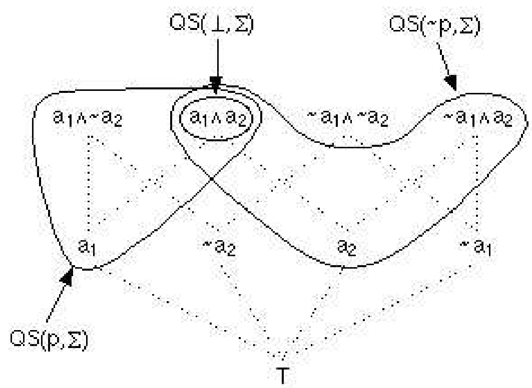

Figure 3.1: The quasi–supports for p, $\sim\!p$ and $\perp$.

Support follows from Equation 3.3, i.e. $SP(p, \Sigma) = \{a_1, a_1 \wedge \sim\!a_2\}$ and $SP(\sim\!p, \Sigma) = \{a_2, \sim\!a_1 \wedge a_2\}$. These sets are shown in Figure 3.2.

The following two theorems summarize some basic properties for quasi–support and support.

Theorem 3.1 *(Kohlas, 1995) Let* (Σ, A, P) *be an assumption–based knowledge. Then, if* h_1 *and* h_2 *are well–formed formulae over* $L = A \cup P$*:*

(Q1) $QS(\top, \Sigma) = C_{A^\pm}$,

(Q2) $QS(h_1 \wedge h_2, \Sigma) = QS(h_1, \Sigma) \cap QS(h_2, \Sigma)$,

(Q3) $h_1 \models h_2$ *implies* $QS(h_1, \Sigma) \subseteq QS(h_2, \Sigma)$,

(Q4) $QS(h_1 \vee h_2, \Sigma) \supseteq QS(h_1, \Sigma) \cup QS(h_2, \Sigma)$.

Theorem 3.2 *(Kohlas, 1995) Let* (Σ, A, P) *be an assumption–based knowledge.Then, if* h_1 *and* h_2 *are well–formed formulae over* $L = A \cup P$*:*

(S1) $SP(\perp, \Sigma) = \emptyset$,

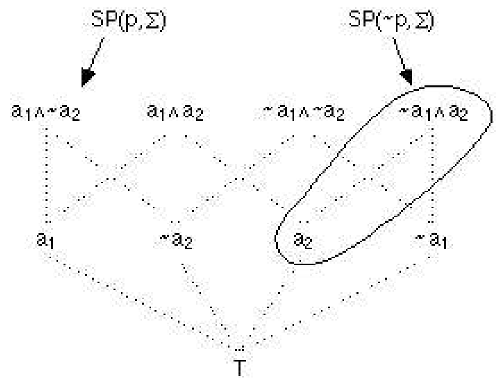

Figure 3.2: The supports for p and $\sim p$.

(S2) $SP(h_1 \wedge h_2, \Sigma) = SP(h_1, \Sigma) \cap SP(h_2, \Sigma)$,

(S3) $h_1 \models h_2$ *implies* $SP(h_1, \Sigma) \subseteq SP(h_2, \Sigma)$,

(S4) $SP(h_1 \vee h_2, \Sigma) \supseteq SP(h_1, \Sigma) \cup SP(h_2, \Sigma)$.

The notion of support is a symbolic analogue of Shafer's (numerical) belief functions (Shafer, 1976; Shafer, 1979). In this sense the theory developed here is a symbolic version of Shafer's evidence theory. This will become more clear in Subsection 3.3.

Quasi–support and support are fundamental concepts of symbolic evidence theory. However, these sets have the disadvantage of containing a lot of redundancy. According to (Q3) and (S3), the sets $QS(h, \Sigma)$ and $SP(h, \Sigma)$ contain also all arguments in favour of every $h' \models h$. Therefore, for many practical, especially computational purposes, it is advantageous to consider arguments which support h, but not h' such that $h' \models h$ and $h' \neq h$.

However, this idea needs to be refined. Let Q be any set of propositional symbols and $\mathcal{L}_Q$ the propositional language over Q (the set of wff over Q). Q may or may not be a subset of $L = A \cup P$. Now, we consider arguments for some $h \in \mathcal{L}_Q$, which are not arguments for any $h' \in \mathcal{L}_Q$, $h' \models h$, $h' \neq h$:

$$M_Q(h, \Sigma) = \{a \in C_{A^{\pm}} : a, \Sigma \models h,\ a, \Sigma \not\models h', \text{ for } h' \in \mathcal{L}_Q, h' \models h,\ h' \neq h\}. \tag{3.6}$$

$M_Q(h, \Sigma)$ is called the **basic argument** for h relative to Q (given Σ). It is possible that $M_Q(h, \Sigma)$ is empty for some $h \in \mathcal{L}_Q$. In fact, this is often the case and it is one of the advantages of working with basic arguments rather than with quasi–supports or supports.

In the above example there are in fact only four non–empty basic arguments relative to P: $M_P(\bot, \Sigma) = \{a_1 \wedge a_2\}$, $M_P(p, \Sigma) = \{a_1, a_1 \wedge \sim a_2\}$, $M_P(\sim p \wedge q, \Sigma) = \{a_2, \sim a_1 \wedge a_2\}$ and $M_P(\top, \Sigma) = \{\top, \sim a_1, \sim a_2, \sim a_1 \wedge \sim a_2\}$ (see Figure 3.3).

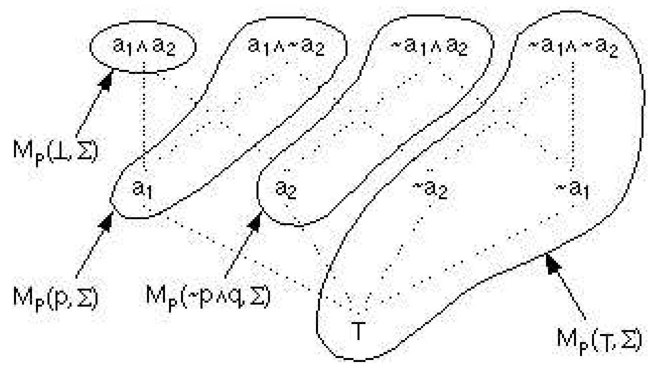

Figure 3.3: The basic arguments for the hypotheses $\bot$, p, $\sim p$ and $\top$.

The following theorem summarizes two basic properties of basic arguments (compare with Figure 3.3).

Theorem 3.3 *(Kohlas, 1993) Let (Σ, A, P) be an assumption–based knowledge and Q an arbitrary set of propositional symbols. The basic arguments satisfy the following properties:*

(M1) If $h_1, h_2 \in \mathcal{L}_Q$, $h_1 \neq h_2$, then

$$M_Q(h_1, \Sigma) \cap M_Q(h_2, \Sigma) = \emptyset. \tag{3.7}$$

(M2)

$$\bigcup \{M_Q(h, \Sigma) : \ h \in \mathcal{L}_Q\} = C_{A^\pm}. \tag{3.8}$$

Theorem 3.3 shows that basic arguments, in addition to be often empty, are not redundant, contrary to quasi–supports or supports. The next theorem shows that quasi–support and support can easily be obtained from basic arguments.

Theorem 3.4 *(Kohlas, 1993) Let (Σ, A, P) be an assumption–based knowledge and Q an arbitrary set of propositional symbols. For every $h \in \mathcal{L}_Q$,*

$$QS(h,\Sigma) = \bigcup \{M_Q(h',\Sigma) : h' \in \mathcal{L}_Q, h' \models h\}, \tag{3.9}$$

$$SP(h,\Sigma) = \bigcup \{M_Q(h',\Sigma) : h' \in \mathcal{L}_Q, h' \models h, h' \neq \bot\}. \tag{3.10}$$

Theorem 3.4 shows that basic arguments are central in the sense that they permit to obtain other elements like quasi–support and support very easily. They correspond essentially to the basic probability assignments of numerical evidence theory (Shafer, 1976).

Basic arguments can also be derived from quasi–supports:

$$M_Q(h,\Sigma) = QS(h,\Sigma) - \bigcup \{QS(h',\Sigma) : h' \in \mathcal{L}_Q, h' \models h, h' \neq h\}. \tag{3.11}$$

An assumption–based knowledge (Σ, A, P), together with the concepts of quasi–support, support, basic arguments, etc. forms a **(symbolic) argumentation system**.

3.2 Representations

The sets $QS(h,\Sigma)$, $SP(h,\Sigma)$ and $M_Q(h,\Sigma)$ may be very large and their representation by the explicit, exhaustive list of their elements may not be feasible. For practical reasons we need more economic representations and for theoretical needs, also other, more convenient representations may be helpful.

Clearly, the quasi–support $QS(h,\Sigma)$ is always an upward–set and can thus be represented by the minimal elements $\mu QS(h,\Sigma)$. These border sets are often much smaller than the sets themselves. In the example of Subsection 3.1 we have $\mu QS(\bot,\Sigma) = \{a_1 \wedge a_2\}$, $\mu QS(p,\Sigma) = \{a_1\}$, and $\mu QS(\sim p,\Sigma) = \{a_2\}$.

The support $SP(h,\Sigma)$ is always a convex set with $\mu SP(h,\Sigma)$ as minimal elements (lower border). Note that $\mu SP(h,\Sigma) \subseteq \mu QS(h,\Sigma)$. The minimal contradictions $\mu QS(\bot,\Sigma)$ are upper bounds for elements of $SP(h,\Sigma)$. This means that $\mu SP(h,\Sigma)$ or $\mu QS(h,\Sigma)$ and $\mu QS(\bot,\Sigma)$ together provide for a representation of $SP(h,\Sigma)$. In fact, $a \in SP(h,\Sigma)$ if and only if

(1) there is a $s \in \mu SP(h,\Sigma)$ (or $s \in \mu QS(h,\Sigma)$) such that $s \subseteq a$,

(2) there is no $s' \in \mu QS(\bot, \Sigma)$ such that $s' \subseteq a$.

Similarly, the basic arguments $M_Q(h, \Sigma)$ are also convex sets. Here again, $\mu QS(h, \Sigma)$ contains the minimal elements of $M_Q(h, \Sigma)$. All elements of $QS(h', \Sigma)$ with $h' \in \mathcal{L}_Q$, $h' \models h$, $h' \neq h$, and especially the elements of $\mu QS(h', \Sigma)$ are upper bounds for the elements of $M_Q(h, \Sigma)$. Therefore, all the $\mu QS(h', \Sigma)$ for all $h' \in \mathcal{L}_Q$ form a possible representation of $M_Q(h, \Sigma)$.

This shows finally that the **minimal quasi–support** $\mu QS(h, \Sigma)$ forms a possible and convenient representation for quasi–support, basic arguments, and other related concepts. It is easy to express the basic set operations using the minimal elements of upward–sets. If Q', Q'' are upward–sets, then

$$\mu(Q' \cap Q'') = \mu\{a' \cup a'' : a' \in \mu Q', \ a'' \in \mu Q''\}, \tag{3.12}$$

$$\mu(Q' \cup Q'') = \mu(\mu Q' \cup \mu Q''). \tag{3.13}$$

Another representation of some theoretical importance (especially in Subsection 3.3) can be obtained by considering interpretations (Chang & Lee, 1973). Interpretations for a propositional language $\mathcal{L}_R$ are Boolean vectors in $N_R = \{0,1\}^{|R|}$. If $h \in \mathcal{L}_R$ and $x \in \{0,1\}^{|R|}$, then each occurrence of a propositional symbol p_i in h is replaced by x_i. Then a Boolean formula is obtained which evaluates by the usual Boolean operations either to 0 or 1. This evaluation is denoted by $h(x)$ and $N_R(h) = \{x \in \{0,1\}^{|R|} : \ h(x) = 1\}$ is defined as the set of interpretations for which h evaluates to 1 (true).

Now, let

$$N(QS(h, \Sigma)) = \bigcup \{N_A(a) : \ a \in QS(h, \Sigma)\} \ \subseteq \ N_A \tag{3.14}$$

be the set of all interpretations $x \in N_A$ of assumptions which together with Σ induce h. This is the representation of $QS(h, \Sigma)$ in the space of interpretations of assumptions. Similarly, let

$$N(SP(h, \Sigma)) = N(QS(h, \Sigma)) - N(QS(\bot, \Sigma)). \tag{3.15}$$

Furthermore, according to (3.11) we define

$$N(M_Q(h, \Sigma)) = N(QS(h, \Sigma)) - \bigcup \{N(QS(h', \Sigma)) : h' \in \mathcal{L}_Q, h' \models h, h' \neq h\}. \tag{3.16}$$

These are representations of $QS(h, \Sigma)$, $SP(h, \Sigma)$ and $M_Q(h, \Sigma)$ in the space of interpretations of assumptions. Note that the relations of theorem 3.1 to

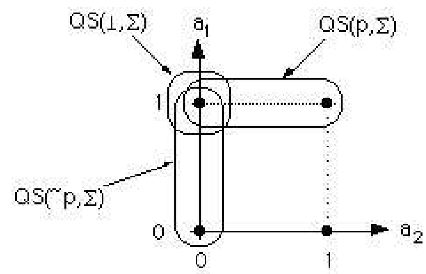

Figure 3.4: Arguments in the space of the interpreations.

theorem 3.4 can be translated into this new representation. In Figure 3.4 some sets are displayed for the example of subsection 3.1.

It is possible to reconstruct $QS(h,\Sigma)$, $SP(h,\Sigma)$ and $M_Q(h,\Sigma)$ from these sets. For any subset $S \subseteq \{0,1\}^{|R|}$ there is a formula $f \in \mathcal{L}_R$ such that $N_R(f) = S$. Any such formula is a logical representation of the set S. In fact there are many such formulae. We denote by $qs(h,\Sigma)$ any formula $f \in \mathcal{L}_A$ such that $N_A(f) = N(QS(h,\Sigma))$. $qs(h,\Sigma)$ is called the **logical representation** of $QS(h,\Sigma)$. One particular logical representation of $QS(h,\Sigma)$ is given by the disjunction of the conjunctions in $\mu QS(h,\Sigma)$:

$$qs(h,\Sigma) = \vee \{a : \ a \in \mu QS(h,\Sigma)\}. \tag{3.17}$$

Note that the symbol = used in Equation 3.17 (as well as in the following equations) has to be interpreted as logical equivalence. According to (3.3) and (3.11) we obtain further logical representations for $SP(h,\Sigma)$ and $M_Q(h,\Sigma)$:

$$sp(h,\Sigma) = qs(h,\Sigma) \wedge {\sim} qs(\perp,\Sigma), \tag{3.18}$$

$$m_Q(h,\Sigma) = qs(h,\Sigma) \wedge {\sim} \vee \{qs(h',\Sigma) : \ h' \in \mathcal{L}_Q, \ h' \models h, \ h' \neq h\}. \tag{3.19}$$

In practice only some particular logical forms are of interest for computational purposes. We are going to isolate such a particular form. A conjunction $g \in \mathcal{L}_Q$ is called an **implicant** of f, if $g \models f$, and g is called a **prime implicant** of f, if g is an implicant of f and there is no $g' \subset g$, which is also an implicant of f. The set of all prime implicants of f will be denoted by $\Psi(f)$. A disjunction of conjunctions of literals is called a disjunctive normal form (DNF). The disjunction $\vee \{g : \ g \in \Psi(f)\}$ of all prime implicants g of f is denoted by $\psi(f)$. It is a particular DNF and it can be shown to be equivalent to f (Birkhoff & Bartee, 1970).

We propose such DNFs to be used as logical represention of quasi–supports, supports, etc. It is not the minimal nor the shortest logical form, but it is convenient to compute it.

3.3 Probability Structure

So far, we have discussed the definition and representation of arguments for hypotheses. How likely is it that such arguments hold? In order to answer this question, probabilities on the assumptions must be defined. This induces a probabilistic structure upon the symbolic argumentation system discussed so far. We speak then of a **probabilistic argumentation system**.

Probabilities on assumptions are introduced by defining a probability measure on the space $N_A = \{0,1\}^{|A|}$ of interpretations of assumptions. Often assumptions are a priori assumed to be mutually independent. Then, for all $x \in N_A$ we have

$$p(x) = \prod \{p_i : \ x_i = 1\} \cdot \prod\{(1-p_i) : \ x_i = 0\}. \tag{3.20}$$

Here p_i is the probability that the assumption a_i is true, and $1-p_i$ the probability that a_i is not true ($\sim a_i$ is true). The probability of a formula $f \in \mathcal{L}_A$ is defined by $p(f) = p(N(f))$. In particular, this determines probabilities

$$p(qs(h,\Sigma)) = p(N(QS(h,\Sigma))) \tag{3.21}$$

for the quasi–supports of h relative to Σ. However, the probability p defined on $\{0,1\}^{|A|}$ is to be considered as an **a priori** probability which may be changed when additional information or knowledge becomes available. The knowledge Σ is in fact an additional information, which excludes the contradictory interpretations $N(QS(\bot,\Sigma))$ relative to Σ as impossible. This information shows that the only possible interpretations given Σ are those of $N^c(QS(\bot,\Sigma))$. Therefore, the **conditional probability** given $N^c(QS(\bot,\Sigma))$ has to be taken into account when the probabilities of the supports are to be considered.

The conditional probability of the support of a hypothesis h is called the **degree of support** of h. In view of (3.15) this conditional probability can be computed by

$$\begin{aligned} sup(h,\Sigma) &= p(N(SP(h,\Sigma)) \mid N^c(QS(\bot,\Sigma))) \\ &= \frac{p(QS(h,\Sigma)) - p(QS(\bot,\Sigma))}{1 - p(QS(\bot,\Sigma))} \\ &= \frac{p(SP(h,\Sigma))}{1 - p(QS(\bot,\Sigma))}. \end{aligned} \tag{3.22}$$

The conditional probability of the possibility of h is called the **degree of plausibility** of h. It is obtained by

$$\begin{aligned} pla(h,\Sigma) &= p(N^c(QS(\sim h,\Sigma)) \mid N^c(QS(\bot,\Sigma))) \\ &= \frac{1 - p(QS(\sim h,\Sigma)}{1 - p(QS(\bot,\Sigma)} = 1 - sup(\sim h,\Sigma). \end{aligned} \tag{3.23}$$

The conditional probabilities of the basic arguments can be computed by

$$bpa_Q(h,\Sigma) = \frac{p(M_Q(h,\Sigma))}{1 - p(M_Q(\bot,\Sigma))} \tag{3.24}$$

for any $h \in \mathcal{L}_Q$, $h \neq \bot$. Note that $M_Q(\bot,\Sigma) = QS(\bot,\Sigma)$. $bpa_Q(h,\Sigma)$ is called the **basic probability assignment** (relative to the language $\mathcal{L}_Q$). For any $h \in \mathcal{L}_Q$ we have

$$sup(h,\Sigma) = \sum \{bpa_Q(h',\Sigma) : h' \in \mathcal{L}_Q,\ h' \models h,\ h' \neq \bot\}, \tag{3.25}$$

$$pla(h,\Sigma) = \sum \{bpa_Q(h',\Sigma) : h' \in \mathcal{L}_Q,\ h' \wedge h \neq \bot\}. \tag{3.26}$$

This corresponds essentially to the notion of basic probability assignment in (Shafer, 1979).

Furthermore, $sup(h,\Sigma)$ and $pla(h,\Sigma)$ have the properties of belief and plausibility functions of Shafer's evidence theory (Shafer, 1976), that is for $n \geq 1$ and $i = 1,\ldots,n$

$$sup(h) \geq \sum \left\{(-1)^{|I|+1} \cdot sup(\wedge\{h_i :\ i \in I\}) :\ \emptyset \neq I \subseteq \{1,\ldots,n\}\right\}, \quad \text{for } h_i \models h, \tag{3.27}$$

$$pla(h) \leq \sum \left\{(-1)^{|I|+1} \cdot pla(\vee\{h_i :\ i \in I\}) :\ \emptyset \neq I \subseteq \{1,\ldots,n\}\right\}, \quad \text{for } h \models h_i. \tag{3.28}$$

That is, *sup* is monotone and *pla* alternating of infinite order. All this shows that probabilistic argumentation systems constructed on propositional logic are a special case of Shafer's original evidence theory (Shafer, 1976).

4 Computational Methods: Symbolic Argumentation Systems

The computational problem in a symbolic argumentation system based upon an assumption–based knowledge (Σ, A, P) consists in determining arguments like quasi–supports, supports, etc. for a given hypothesis h. This section presents an overview of the existing computational approaches to find symbolic arguments.

4.1 Clause Management Systems

The methods discussed in this first subsection are based on **resolution**. They are used to derive quasi–supports from a general assumption–based knowledge (Σ, A, P) in a convenient representation. Other symbolic arguments like support, plausibility and refutation are obtained indirectly from quasi–support.

From Equation 3.1 we know that the quasi–support is the set of the conjunctions $a \in C_{A^{\pm}}$ which, together with the knowledge Σ, logically entail the hypothesis h, that is $a, \Sigma \models h$. How are such conjunctions obtained from the given knowledge Σ and the hypothesis h? Note that if f and g are arbitrary logical formulae such that f entails g, $f \models g$, then $\sim f$ is entailed by $\sim g$, that is $\sim g \models \sim f$. Therefore, it follows that

$$a, \Sigma \models h \iff \sim h, \Sigma \models \sim a, \tag{4.1}$$

where $\sim a$ is a clause. A clause g is called an **implicate** of a formula f, if $f \models g$, and g is called a **prime implicate** of f, if g is an implicate of f and there is no $g' \subset g$, which is also an implicate of f. The set of all prime implicates of f will be denoted by $\Phi(f)$. A conjunction of clauses is called a conjunctive normal form (CNF). The conjunction $\wedge \{g : \; g \in \Phi(f)\}$ of all prime implicates g of f is denoted by $\phi(f)$. An algorithm to generate the prime implicates of a set Σ of clauses is described in Subsection 4.1.3.

Thus, determining arguments in favour of h essentially means determining those implicates of $\sim h, \Sigma$ which consist of assumptions only. If h is a clause then $\sim h$ is a conjunction, and instead of $\sim h, \Sigma$ we write Σ^*. In the following, we restrict h to be a clause, and we present three different methods to generate such implicates from Σ^*.

4.1.1 Variable Elimination Method

The first approach is called **variable elimination** method. The idea is to eliminate all propositional symbols $p_i \in P$ from Σ^*. First, consider the eliminiation of a single proposition p from an arbitrary set of clauses Σ. For that purpose, let us decompose Σ into three disjoint subsets:

$$\Sigma_p = \{\xi_i \in \Sigma : \ p \in \xi_i\}, \tag{4.2}$$

$$\Sigma_{\sim p} = \{\xi_i \in \Sigma : \ \sim p \in \xi_i\}, \tag{4.3}$$

$$\Sigma_{-p} = \{\xi_i \in \Sigma : \ p, \sim p \notin \xi_i\} = \Sigma - (\Sigma_p \cup \Sigma_{\sim p}). \tag{4.4}$$

Then, the new set Σ' obtained after eliminating the variable p is

$$\Sigma' = \mu(\Sigma_{-p} \cup \{\rho(\xi_i, \xi_j) : \ \xi_i \in \Sigma_p, \ \xi_j \in \Sigma_{\sim p}\}). \tag{4.5}$$

The same procedure can be applied repeatedly in order to eliminate all n propositions $p_i \in P$, $i = 1, \ldots, n$, from $\Sigma^* = \Sigma^*_0$. This is illustrated in the introductory example of Section 2. The new knowledge base Σ^*_n obtained after eliminating all propositions satisfies two important properties:

Theorem 4.1 *(Kohlas & Moral, 1995) Let Σ^* be the set of clauses obtained from the assumption–based knowledge (Σ, A, P) and the hypothesis $h \in \mathcal{L}_L$, and let Σ^*_n be the resulting set obtained after eliminating all propositions $p_i \in P$ from Σ^* in an arbitrary order. Σ^*_n satisfies*

- *(1) $\Sigma^* \models \Sigma^*_n$,*
- *(2) $\Sigma^* \models \sim a$, $a \in C_{A^\pm}$, implies $\Sigma^*_n \models \sim a$.*

If K is an arbitrary set of clauses, then the set of conjunctions obtained by negating the clauses in K is denoted by $\sim K$. Using this notation, the main theorem of this method can be formulated:

Theorem 4.2 *If Σ^*_n and h are defined as in Theorem 4.1, then*

$$\mu QS(h, \Sigma) = \sim \Phi(\Sigma^*_n). \tag{4.6}$$

If Σ^*_n is a consequence of Σ^*, that is $\Sigma^* \models \Sigma^*_n$, then the clauses $\xi \in \sim C_{A^\pm}$ of Σ^*_n are all implicates of Σ^*. Thus, the negations $\sim \xi$ of these clauses are arguments of h relative to Σ, that is $\sim \xi \in QS(h, \Sigma)$. In order to obtain all arguments for h, the new set Σ^*_n must also satisfy a second condition: if $a \in C_{A^\pm}$ is an argument of h, that is $\Sigma^* \models \sim a$, then $\Sigma^*_n \models \sim a$. Thus, the arguments $a \in QS(h, \Sigma)$ of h are the implicates of Σ_n^*, and the set of minimal arguments $\mu QS(h, \Sigma)$ corresponds to $\sim\Phi(\Sigma_n^*)$. Note that the order in which the variables are eliminated does not influence the resulting set of prime implicates $\Phi(\Sigma_n^*)$.

The disadvantage of this method is that for each new hypotheses the whole computation has to be redone. To avoid redundant computations we show in Subsection 4.3 how the variable elimination method can be adapted for a given decomposition of the knowledge base.

4.1.2 SOL Resolution

Another alternative approach to compute arguments like quasi–supports is based on a resolution procedure called **skipped ordered linear (SOL) resolution**, initially proposed in (Siegel, 1987) and further developed in (Inoue, 1991; Inoue, 1992).

Some further terminology and concepts are necessary in order to describe this approach. A **production field** F is a non–empty set of clauses. The production field which will interest us most is the set of all clauses consisting of literals of assumptions $\sim C_{A^\pm}$. A production field F is called **stable** if $c \in F$ and $c' \subseteq c$ imply $c' \in F$ (that is F is a downward–set). Clearly $\sim C_{A^\pm}$ is a stable production field. The empty clause $\perp$ belongs to every stable production field.

If F is a stable production field, then $Carc(\Sigma, F)$ denotes the set of all prime implicates of Σ which belong also to F, $Carc(\Sigma, F) = \Phi(\Sigma) \cap F$. The elements of $Carc(\Sigma, F)$ are called **characteristic clauses** of Σ with respect to F. Theorem 4.3 shows how characteristic clauses relate to minimal quasi–supports:

Theorem 4.3 *(Kohlas & Monney, 1995) For any hypothesis $h \in \mathcal{L}_L$ we have*

$$\mu QS(h, \Sigma) = \sim Carc\left(\Sigma \cup \{\sim h\}, \sim C_{A^\pm}\right), \tag{4.7}$$

$$\mu QS(\perp, \Sigma) = \sim Carc\left(\Sigma, \sim C_{A^\pm}\right). \tag{4.8}$$

According to this theorem characteristic clauses have to be computed in order to obtain the minimal quasi–supports. The skipped ordered linear (SOL) resolution offers a possibility to solve this problem.

This procedure works with **structured clauses**, that is pairs (p, q^*), where p is an ordinary clause and q^* a sequence $s_1, s_2, \ldots, s_t$ of non–empty sequences s_i of literals. q^* represents a tree, the sequences $s_i = (\ell_1, \ell_2, \ldots, \ell_r)$

are called its branches and the first element ℓ_1 of a branch is called its leaf. Two operations are defined for structured clauses:

(1) **Skip**: If $(p, (\ell s)q^*)$ is a structured clause with a branch (ℓs), with leaf ℓ, and a remaining tree q^*, then

$$\sigma(p, (\ell s)q^*) = (p \vee \ell, q^*). \tag{4.9}$$

(2) **Resolve**: If $(p, (\ell s)q^*)$ is a structured clause with a branch (ℓs), with leaf ℓ and a remaining tree q^*, and if there is a clause ξ in Σ which satisfies the following conditions of non–repetition,

  (1) x contains the literal $\sim\ell$,
  (2) no literal of x is contained in the branch (ℓs),
  (3) no literal of x appears negated as a leaf in q^*,
  (4) no literal of x appears negated in p,

then

$$\rho((p, (\ell s)q^*), \xi) = (p, (\ell'_1 \ell s)(\ell'_2 \ell s) \ldots (\ell'_t \ell s)q^*), \tag{4.10}$$

where the literals ℓ'_i are those literals from x which are not removable, that is which satisfy the following conditions:

  (2′) $\sim\ell'_i$ does not appear in the branch (ℓs),
  (3′) ℓ'_i does not appear as a leaf in q^*,
  (4′) ℓ'_i does not appear in p.

Note that the literal $\sim\ell$ of x is removable, because it does not satisfy (2′). If all literals of x are removable, then the tree on the left in (4.10) reduces to q^*. The order in which the literals ℓ'_i are arranged in (4.10) is irrelevant for the correctness of the procedure, but may influence its performance.

Given a set of clauses Σ, a **production** of a clause f from Σ is a sequence of structured clauses (p_i, q_i^*), $i = 1, 2, \ldots, n$, which satisfies the following conditions:

(1) $(p_1, q_1^*) = (\bot, (\ell_1)\,(\ell_2) \ldots (\ell_m))$, where the ℓ_i are the literals of a clause ξ from Σ arranged in an arbitrary order.
(2) $(p_n, q_n^*) = (f, \emptyset)$.
(3) (p_{i+1}, q_{i+1}^*) equals either $\sigma(p_i, q_i^*)$ or $\rho((p_i, q_i^*), \xi)$ for some ξ of Σ.

These productions are designed to generate implicates of Σ. It can be shown that this goal is achieved:

Theorem 4.4 *(Siegel, 1987)*

(1) A finite set of clauses Σ has a finite set of productions (finiteness).

(2) If a clause f is produced from Σ, then f is an implicate of Σ (soundness).

(3) If g is an implicate of Σ then there is a production from Σ which produces a clause $f \subseteq g$ (completeness).

(4) If c is a clause, g an implicate of $\Sigma \cup \{c\}$, but not of Σ, then there is a production f from $\Sigma \cup \{c\}$ starting with c such that $f \subseteq g$.

Let's illustrate the procedure with a very simple example. Σ consists of the three clauses

$$(1) \quad {\sim}c \vee {\sim}a, \qquad (2) \quad {\sim}c \vee {\sim}b, \qquad (3) \quad a \vee b.$$

The following sequence is a production from Σ starting with $a \vee b$:

(1) $(\bot, (a)\,(b))$ initial clause,
(2) $(\bot, ({\sim}ca)\,(b))$ resolve with clause (1),
(3) $({\sim}c, (b))$ skip,
(4) $({\sim}c, \emptyset)$ resolve with clause (2).

Thus, ${\sim}c$ is an implicate of Σ.

These productions can now be controlled to produce only minimal clauses in a stable production field: at any step (p_i, q_i^*), p_i never decreases. Hence, once p_i is outside a stable production field, the production can be stopped, if only clauses in the production field should be produced. This is because all subsequent clauses, and in particular the final clause f, remain outside the production field. Furthermore, once a p_i contains a clause already produced earlier, the production can also be stopped, if only minimal clauses are to be produced. Theorem 4.3 remains valid for this adapted procedure, now for implicates in the production field considered.

Given a set of clauses Σ, a clause c and a stable production field F, let $Prod(\Sigma, c, F)$ denote the set of all clauses produced from Σ, starting with c, being in F and not containing another clause satisfying the first two conditions. These production can be used to compute characteristic clauses incrementally as the following theorem shows:

Theorem 4.5 *(Inoue, 1992) If c is a clause, Σ a set of clauses and F a stable production field, then*

$$\begin{aligned} Carc(\emptyset, F) &= \{p \vee {\sim}p \in F : \; p \in L\}, & (4.11) \\ Carc(\Sigma \cup \{c\}, F) &= \mu(Carc(\Sigma, F) \cup Prod(\Sigma, c, F)). & (4.12) \end{aligned}$$

First, it is important to remark that in practice the complete knowledge Σ generally decomposes into a relatively stable knowledge base Σ_K and a set of varying facts Σ_F, $\Sigma = \Sigma_K \cup \Sigma_F$. This has to be taken into account if efficient computational procedures are to be designed. Then theorem 4.5 can be applied to argumentation systems in the following manner (Kohlas & Monney, 1994):

The first task is to compute the minimal contradictions, that is essentially $Carc(\Sigma, {\sim}C_{A^\pm})$ (see theorem 4.3). In order to apply theorem 4.5, the clauses of Σ are ordered into a sequence $\xi_1, \xi_2, \ldots, \xi_r$ where it is convenient to take first the clauses of Σ_K. Then, using theorem 4.5, first the minimal contradictions relative to Σ_K, ${\sim}Carc(\Sigma_K, {\sim}C_{A^\pm})$, can be computed by applying theorem 4.5 sequentially to $\Sigma_i = \{\xi_1, \xi_2, \ldots, \xi_i\}$ and $c = \xi_{i+1}$. In many practical cases ${\sim}Carc(\Sigma_K, {\sim}C_{A^\pm})$ will contain only trivial contradictions $a \wedge {\sim}a$, because the knowledge base itself may or should be consistent. This computation is sort of a **compilation** of the knowledge base.

The facts themselves often also arise sequentially or can, in any case, be arranged sequentially. The set of minimal contradictions ${\sim}Carc(\Sigma, {\sim}C_{A^\pm})$ can be updated from the compiled knowledge base by again repeatedly using theorem 4.5.

At this stage queries about hypotheses h can be accepted. The minimal quasi–supports of h equal ${\sim}Carc(\Sigma \cup \{{\sim}h\}, {\sim}C_{A^\pm})$ by theorem 4.3. If ${\sim}h$ is a clause (that is, h a conjunction) then this set can be computed from the contradictions ${\sim}Carc(\Sigma_K, {\sim}C_{A^\pm})$ using theorem 4.4. This means essentially computing $Prod(\Sigma, {\sim}h, {\sim}C_{A^\pm})$.

If h is not a conjunction, then ${\sim}h$ must first be transformed into an equivalent CNF. If ${\sim}h = h_1 \wedge \ldots \wedge h_t$, then

$$Carc(\Sigma \cup \{{\sim}h\}, {\sim}C_{A^\pm}) = Carc(\Sigma \cup \{h_1, \ldots, h_t\}, {\sim}C_{A^\pm}), \qquad (4.13)$$

and theorem 4.4 can once more be applied sequentially to compute the latter characteristic clauses. Note that SOL resolution and theorem 4.4 can also be used to compute prime implicates of Σ. The production field must only be set equal to ${\sim}C_L$.

Also, other production fields than $\sim C_{A^\pm}$ may be of interest. Suppose for example that arguments (conjunctions of literals of assumptions) are classified somehow into **relevant** and **irrelevant** ones, such that, if a is relevant, $a' \subseteq a$, then a' is also relevant. This clearly defines a stable production field F of relevant clauses $\sim a$ and the search can be limited to relevant arguments. Theorem 4.4 and 4.5 apply also to this more general situation. For example only arguments of a length of k or shorter may be considered as relevant (arguments with many assumptions are not interesting) or only arguments with a probability larger than some small ε may be considered as relevant (very unlikely arguments may be neglected).

Another idea to classify arguments into relevant and irrelevant ones is to define a **cost function** $c : C_{A^\pm} \to \mathbb{R}$, which represents somehow the desirability of the arguments $a \in C_{A^\pm}$ (Kohlas, 1996). It is assumed that $a \subseteq a'$ implies $c(a) \leq c(a')$. If β is a cost bound, then an argument a is said to be relevant, if $c(a)$ is smaller than β. Let $F_\beta = \{a \in C_{A^\pm} : c(a) \leq \beta\}$ be the stable production field of such arguments. Then the problem is to find $QS_\beta(h, \Sigma) = QS(h, \Sigma) \cap F_\beta$. Depending on the choice of β, $QS_\beta(h, \Sigma)$ may possibly become much smaller than $QS(h, \Sigma)$.

4.1.3 Generating Prime Implicates

First, the following theorem shows how quasi–supports relative to a knowledge Σ relate to the prime implicates of Σ.

Theorem 4.6 *(Reiter & de Kleer, 1987b) If $h \in {}_L$ is a clause, then*

$$\mu QS(h, \Sigma) = \sim\mu\{f - h \in \sim C_{A^\pm} : f \in \Phi(\Sigma)\}, \tag{4.14}$$

$$\mu QS(\bot, \Sigma) = \sim\{f \in \sim C_{A^\pm} : f \in \Phi(\Sigma)\}. \tag{4.15}$$

According to this theorem, another way to determine quasi–supports and contradictions for clauses consists in determining the prime implicates of Σ, followed by filtering the minimal quasi–supports out of them with the aid of Equation 4.14.

For small knowledge bases Σ with a small number of prime implicates, this may be an appropriate method to determine quasi–supports. For example the decomposition method described in Subsection 4.3 may yield such small factor bases where this method can be applied. This presupposes the availability of methods to determine prime implicates (see below). For larger knowledge bases Σ however the set of prime implicates may soon become too large to be enumerated completely. Then more controlled methods must be used (see Subsection 4.1.2).

One method to obtain prime implicates of a set of clauses uses **resolution**. If f and g are two clauses with exactly one pair of literals with opposite sign, that is, if $f = f' \vee p$ and $g = g' \vee \sim p$, then the **resolvent** $\rho(f,g)$ is the clause $f' \vee g'$, where it is understood that all multiple literals are regrouped such that a literal appears only once. The next theorem shows how all prime implicates of a set of clauses can be obtained.

Theorem 4.7 *(Birkhoff & Bartee, 1970) If Σ is a satisfiable set of clauses, then the complete set of prime implicates of Σ can be found by repeatedly performing the following operation on Σ as long as it is possible:*

(1) If a clause ξ of Σ is contained in another clause ξ' of Σ, $\xi \subseteq \xi'$, then drop ξ'.

(2) If for two clauses ξ' and ξ'' of Σ a resolution is possible and if the resolvent $\rho(\xi', \xi'')$ does not contain a clause already present in Σ, then add $\rho(\xi', \xi'')$ to Σ.

The clauses in Σ upon termination of this procedure are the prime implicates of Σ.

This theorem describes of course a non–deterministic procedure and not yet an algorithm. Many optimizations are possible in selecting the resolvents, if several are possible. See also Subsection 4.1.2 for a method to determine prime implicates.

Let us illustrate the method by a simple example. A sensor produces a signal s, if a cause c is present and if the system is operational (assumption a_1). There is the possibility that some other cause (assumption a_2) produces by error also the signal s. Otherwise there is no signal s. This system can be coded by the following three clauses:

$$\sim c \vee \sim a_1 \vee s, \quad \sim a_2 \vee s, \quad c \vee a_2 \vee \sim s.$$

Suppose that the signal s is observed. Then the clause s is added to the knowledge. What is the quasi–support for the hypothesis that the cause c is producing the signal? Note that the first two clauses contain both s and can thus be dropped according to theorem 4.7. The remaining third clause can be resolved with s which gives $c \vee a_2$. Now the third clause can also be dropped and the two clauses

$$s \quad \text{and} \quad c \vee a_2$$

remain. Here the procedure stops and these two clauses are the two only prime implicates of the knowledge. By theorem 4.6 there is only one minimal quasi–support for c, namely $\sim a_2$ and there are no contradictions.

Once the minimal quasi–support of a hypothesis h and the contradictions are determined, then the support of h is also determined (see Subsection 3.2). As there are no contradictions in the example above supports and quasi–support coincide. Therefore, the conjunctions $\sim a_2$, $a_1 \wedge \sim a_2$ and $\sim a_1 \wedge \sim a_2$ are supports for the hypothesis c. Similarly, the refutation and the possibility of h can be determined as soon as the quasi–support for $\sim h$ is known.

4.2 Horn Clauses

The methods of the previous subsections are general in the sense that they are able to treat general clauses. Thus, the full expressiveness of propositional logic can be used to formulate the model. Computationally, this may be very expensive, and hence, the methods described so far are not powerful enough for large models. In this subsection we present a method to generate arguments relative to a knowledge base consisting of **Horn clauses** only. A Horn clause is a clause with at most one positive literal. This is a special case of the general theory. Such systems are known as **basic assumption–based truth maintenance systems (basic ATMS)** (de Kleer, 1986).

A basic ATMS is an assumption–based knowledge (Σ, A, P) with two important restrictions:

(1) The clauses $\xi_i \in \Sigma$ are Horn clauses, that is clauses consisting of at most one positive literal.

(2) The assumptions $a_i \in A$ are only used as negative literals, except if $a_i \in \Sigma$ is a clause.

The second restriction implies that arguments $a \in C_{A^\pm}$ of hypotheses are always conjunctions of positive assumptions. Let $C_A \subseteq C_{A^\pm}$ be the set of conjunctions of positive assumptions. The elements of C_A are called **environments**. Thus, the aim of the basic ATMS algorithm is to determine environments $a \in C_A$ such that $a, \Sigma \models h$. A set of such environments corresponds to the concept of quasi–support. Environments relative to a given basic ATMS can be computed more efficiently than quasi–supports relative to a general assumption–based knowledge.

The Horn clauses ξ_i of a basic ATMS are called **justifications**. The positive literal of a justification represents the **consequent** of the justification. If a justification ξ_i has no positive literal, then the inconsistency $\bot$ is said to be the consequent of ξ_i. The consequent of a justification with no negative literal is called **premise**.

The propositional symbols in $N = P \cup A \cup \{\bot\}$ are called the **nodes** of the ATMS. The basic ATMS algorithm generates simulataneously the quasi–supports (environments) for all the nodes of the ATMS. Thus, only hypotheses $h \in N$ are allowed. The quasi–support of composed hypotheses is obtained using (Q2) and (Q4) of Subsection 3.1. Note that in the case of a basic ATMS the inequality of (Q4) becomes an equality for positive literals:

Theorem 4.8 *(Anrig & Haenni, 1996) Let (Σ, A, P) be a basic ATMS. If $h_1, h_2 \in A \cup P$ are two positive literals, then*

$$QS(h_1 \vee h_2, \Sigma) = QS(h_1, \Sigma) \cup QS(h_2, \Sigma). \tag{4.16}$$

The basic ATMS algorithm generates its results incrementally. The **state** of the basic ATMS algorithm is determined by the current enviroments $V(n)$ for all nodes $n \in N$. The initial state of the algorithm is a state such that

(1) $V(a_i) = \{a \in C_A : \ a \text{ contains } a_i\}$, for all assumptions $a_i \in A$,

(2) $V(p_i) = C_A$, if $p \in P$ is a premise in Σ,

(3) $V(n_i) = \emptyset$, for every other node $n_i \in N$.

A justification $\xi \in \Sigma$ of the form

$$\xi = {\sim}n_1 \vee \ldots \vee {\sim}n_k \vee p = n_1 \wedge \ldots \wedge n_k \rightarrow p, \tag{4.17}$$

$n_i \in N$, $p \in P \cup \{\bot\}$, is **applicable** (on p) to the current state of the algorithm if

$$V(n_1) \cap \ldots \cap V(n_k) \not\subseteq V(p). \tag{4.18}$$

The **application** of ξ to the current state of the ATMS algorithm results in the modification of $V(p)$ as follows:

$$V(p) := V(p) \cup (V(n_1) \cap \ldots \cap V(n_k)). \tag{4.19}$$

Let $(\xi_1, \ldots, \xi_s)$ be a sequence of justifications. The sequcence is said to be **applicable** to the current state of the ATMS algorithm if ξ_1 is applicable and each ξ_i, $2 \leq i \leq s$, is applicable after the sequential application of $\xi_1, \ldots, \xi_{i-1}$. The **application** of the sequence is defined as to be the sequential application of $\xi_1, \ldots, \xi_s$. An applicable sequence of justifiactions is said to be **complete** with respect to Σ if, after the application of the sequence, no justification in Σ is applicable (Ngair, 1992).

Given a basic ATMS (Σ, A, P), the **basic ATMS algorithm** is formally defined to be the selection and application of a complete applicable sequence $(\xi_1, \ldots, \xi_s)$ of justifications $\xi_i \in \Sigma$ to the corresponding initial state. By repeating the search for an applicable justification in Σ, we can always derive a finite complete applicable sequence of justifications. Thus, starting with the initial state, the algorithm repeatedly performs applications of justifications as long as applicable justifications can be found in Σ. Possibly, some justifications are applied more that once, whereas other justifications are never applied. The final state obtained at the end of the algorithm, that is when no more applicable justification can be found, contains the quasi–supports for the nodes of the ATMS:

Theorem 4.9 *(Ngair, 1992) If a basic ATMS is given, then after the basic ATMS algorithm we have for every $n \in N$:*

$$QS(n, \Sigma) = V(n) \cup V(\perp). \tag{4.20}$$

According to (4.19), the application of a justification strictly increases the size of a single $V(n)$ of the current state. Obviously, since C_A and N are both finite sets, there is no infinite applicable sequence of justifications, and therefore, the basic ATMS algorithm terminates.

To implement the algorithm, only procedures to compare sets and to compute unions and intersections of sets are needed. Note that the sets $V(n)$ are always downward–sets. Thus, it is sufficient to store only the minimal elements $\mu V(n)$, called the **label** of $V(n)$, and the set operations can be performed on the labels.

4.3 Decomposition and Valuation Networks

The methods described in the previous subsections are only useful for small knowledge bases. If a large set of clauses Σ is given, then a decomposition approach based on the theory of valuation (Lauritzen & Shenoy, 1996) may be a valuable alternative. It permits to focus on smaller factor bases and on combination operations between these factor bases.

Consider the introductory example of Section 2. The idea of a repeated variable elimination from Σ corresponds to Shenoy's fusion algorithm described in (Lauritzen & Shenoy, 1996). If the individual clauses ξ_i of Σ are regarded as valuations for P_i, where $P_i \subseteq P$ is the set of propositions contained in ξ_i, then the operation described in Equation 4.5 corresponds to Shenoy's fusion operation. The method of eliminating variables can thus be used to construct a rooted join tree from a given knowledge base Σ. Figure 4.1 shows the join tree obtained for the introductory example of Section 2. The exact algorithm is described in detail in (Lauritzen & Shenoy, 1996).

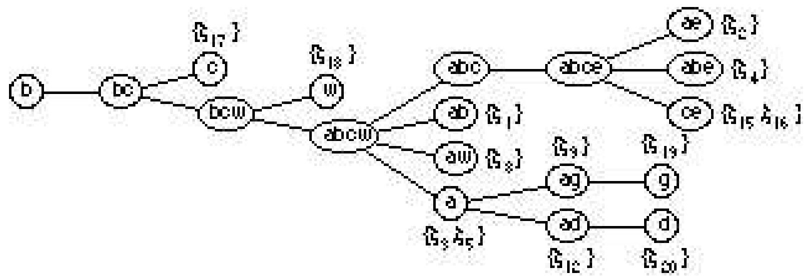

Figure 4.1: The join tree obtained for the introductory example.

The order in which the variables have to be eliminated to obtain the join tree of Figure 4.1 is e, g, d, a, w, c. The root of the join tree is b. The clauses of the initial knowledge base

$$\Sigma = \{\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_8, \xi_9, \xi_{12}, \xi_{15}, \xi_{16}, \xi_{17}, \xi_{18}, \xi_{19}, \xi_{20}\} \tag{4.21}$$

are distributed among the nodes of the join tree. As we will see, the sets of clauses at each node form valuations in the sense of Shenoy's valuation networks. These valuations can be propagated through the network in order to compute the marginals of the factorization for each node of the tree.

Constructing a join tree and distributing the clauses of Σ among the nodes of the tree can also be seen as a decomposition of Σ into disjoint sets of clauses Σ_{P_1} to Σ_{P_k} with $\Sigma_{P_1} \cup \ldots \cup \Sigma_{P_k} = \Sigma$. $P_i \subseteq P$ is the set of propositions contained in Σ_{P_i}, $i = 1, \ldots, k$. Note that $D_P = \{P_1, \ldots, P_k\}$ forms a hypertree (Lauritzen & Shenoy, 1996). Each set Σ_{P_i} defines a corresponding assumption–based knowledge (Σ_{P_i}, A, P_i).

Let $D_\Sigma = \{\Sigma_{P_1}, \ldots, \Sigma_{P_k}\}$ denote the set of knowledge bases obtained after decompositions. The sets Σ_{P_i} are valuations for P_i. The operations of combination and marginalization are the following:

(1) **Combination:** If Σ_{P_1} and Σ_{P_2} are valuations for P_1 and P_2 respectively, then

$$\Sigma_{P_1} \otimes \Sigma_{P_2} = \mu(\Sigma_{P_1} \cup \Sigma_{P_2}) \tag{4.22}$$

is called **combination** of Σ_{P_1} and Σ_{P_2}. $\Sigma_{P_1} \otimes \Sigma_{P_2}$ is a valuation for $P_1 \cup P_2$.

(2) **Marginalization:** Let Σ_{P_1} be a valuation for P_1. If P_2 is another set of propositions such that $P_2 \subseteq P_1$, then $\Sigma_{P_1}^{\downarrow P_2}$ denotes the set of clauses obtained after all variables in $P_1 - P_2$ are eliminated from Σ_{P_1} using the method described in Subsection 4.1.1. $\Sigma_{P_1}^{\downarrow P_2}$ is called the **marginal** of Σ_{P_1} for P_2.

Depending on the choice of the elimination sequences, different results may result from marginalization. However, all possible results are logically equivalent. From this point of view, marginalization is unique. In the following, the symbols $=$ and $\neq$ refer to logical equivalence.

The following theorem summarizes a number of fundamental properties for the operations of combination and marginalization:

Theorem 4.10 *(Kohlas & Moral, 1995)*

(1) The combination operation $\otimes$ is commutative and associative:

$$\begin{aligned} \Sigma_{P_1} \otimes \Sigma_{P_2} &= \Sigma_{P_2} \otimes \Sigma_{P_1}, & (4.23)\\ \Sigma_{P_1} \otimes (\Sigma_{P_2} \otimes \Sigma_{P_3}) &= (\Sigma_{P_1} \otimes \Sigma_{P_2}) \otimes \Sigma_{P_3}. & (4.24) \end{aligned}$$

(2) Consonance of marginalization: if $P_1 \subseteq P_2 \subseteq P_3$, then

$$[(\Sigma_{P_1})^{\downarrow P_2}]^{\downarrow P_3} = (\Sigma_{P_1})^{\downarrow P_3}. \tag{4.25}$$

(3) Distributivity of marginalization over combination:

$$(\Sigma_{P_1} \otimes \Sigma_{P_2})^{\downarrow P_1} = \Sigma_{P_1} \otimes (\Sigma_{P_2})^{\downarrow P_1 \cap P_2}. \tag{4.26}$$

This theorem is crucial for local computation in decomposed knowledge bases. It states that the basic axioms of valuation networks (Lauritzen &

Shenoy, 1996) are also satisfied in the case of assumption–based knowledges. Using Shenoy's generic propagation algorithm, it is therefore possible to compute marginals of the form

$$\Sigma^{\downarrow Q} = (\Sigma_{P_1} \otimes \ldots \otimes \Sigma_{P_k})^{\downarrow Q} \tag{4.27}$$

locally for all subsets $Q \subseteq P_i$, $i = 1, \ldots, k$. If h is a hypothesis on Q and if Σ^* is used instead of Σ (see Subsection 4.1.1), then Theorem 4.1 tells us how to derive quasi–support from $(\Sigma^*)^{\downarrow Q}$.

Instead of propagating sets of clauses Σ_{P_i} through the network, it is also possible to use their corresponding families of basic arguments

$$M_{P_i} = \{M_{P_i}(h, \Sigma_{P_i}) : h \in \mathcal{L}_{P_i}\}. \tag{4.28}$$

There are different methods to derive the familiy of basic arguments M_{P_i} from a knowledge base Σ_{P_i} (Haenni, 1996). Again, such families are valuations in the sense of Shenoy's axiomatic framework of valuation networks. The operations of combination and marginalization are defined as follows:

(1) **Combination:** If M_{P_1} and M_{P_2} are valuations for P_1 and P_2 respectively, then the **combination** of M_{P_1} and M_{P_2} is given by

$$(M_{P_1} \otimes M_{P_2})(h) = \bigcup\{M_{P_1}(h_1) \cap M_{P_2}(h_2) : h_i \in \mathcal{L}_{P_i}, h_1 \wedge h_2 = h\}. \tag{4.29}$$

Obviously, $M_{P_1} \otimes M_{P_2}$ is a valuation for $P_1 \cup P_2$.

(2) **Marginalization:** Let M_{P_1} be a valuation for P_1 and P_2 another set of propositions such that $P_2 \subseteq P_1$. The **marginal** $M_{P_1}^{\downarrow P_2}$ of M_{P_1} for P_2 is obtained by

$$M_{P_1}^{\downarrow P_2}(h_2) = \bigcup\{M_{P_1}(h_1) : h_1 \in \mathcal{L}_{P_1}, h_1^{\downarrow P_2} = h_2\}. \tag{4.30}$$

The marginalization $h^{\downarrow Q}$ of a logical formula h to Q is obtained by dropping all propositional symbols $p \in Q$ from a possible DNF representation of h.

Again, the axioms for local computation in valuation networks are satisfied:

Theorem 4.11 *(Kohlas, 1993)*

(1) The combination operation $\otimes$ is commutative and associative:

$$M_{P_1} \otimes M_{P_2} = M_{P_2} \otimes M_{P_1}, \tag{4.31}$$

$$M_{P_1} \otimes (M_{P_2} \otimes M_{P_3}) = (M_{P_1} \otimes M_{P_2}) \otimes M_{P_3}. \tag{4.32}$$

(2) Consonance of marginalization: if $P_1 \subseteq P_2 \subseteq P_3$, then

$$[(M_{P_1})^{\downarrow P_2}]^{\downarrow P_3} = (M_{P_1})^{\downarrow P_3}. \tag{4.33}$$

(3) Distributivity of marginalization over combination:

$$(M_{P_1} \otimes M_{P_2})^{\downarrow P_1} = M_{P_1} \otimes (M_{P_2})^{\downarrow P_1 \cap P_2}. \tag{4.34}$$

Thus, the use of families of basic arguments is a second possibility to put symbolic argumentation systems into the framework of valuation networks. Thus, there are two different ways to compute symbolic arguments like quasi–support from a decomposed knowledge base D_Σ:

(1) After propagating the sets $\Sigma_{P_i} \in D_\Sigma$ through the valuation network, quasi–support can be derived from the marginals obtained at the nodes of the network.

(2) First, the sets $\Sigma_{P_i} \in D_\Sigma$ are transformed into corresponding families of basic arguments M_{P_i}. After propagating these families through the network, quasi–support can be derived from the resulting marginals.

Figure 4.2 shows a commutative diagram that illustrates how quasi–support can be computed in two different ways. For both cases we have an additional property called **idempotency** (Kohlas & Moral, 1995). Let P' be a subset of P, then

$$\Sigma_P \otimes \Sigma_P^{\downarrow P'} = \Sigma_P, \tag{4.35}$$

$$M_P \otimes M_P^{\downarrow P'} = M_P. \tag{4.36}$$

Idempotency reduces the number of necessary combinations during the outward phase of the propagation process (Lauritzen & Shenoy, 1996).

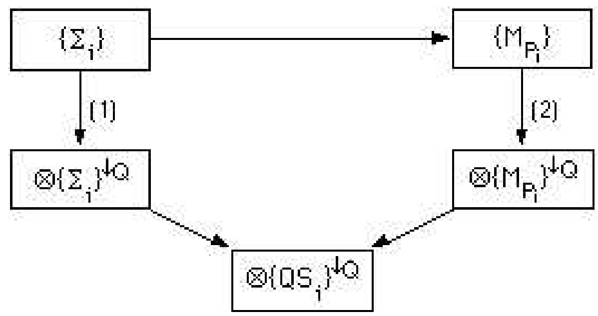

Figure 4.2: The commutative diagram for assumption–based knowledges, quasi–support, and basic arguments.

5 Computational Methods: Probabilistic Argumentation Systems

This section describes computational methods for the probabilistic part of argumentation systems. First, it shows how degrees of support may be obtained from symbolic arguments once probabilties are assigned to the assumptions. Finally, it points out that degrees of support may also be obtained by applying valuation network techniques to the usual numerical Dempster–Shafer theory of evidence.

5.1 From Symbolic to Numerical Evidence

From a given symbolic argumentation system, we get, by the methods of Section 4, the quasi–supports for hypotheses (for example in logical representation). How do we use this symbolic information to compute the numerical degrees of support and plausibility of the hypotheses?

The numerical values $sup(h, \Sigma)$ and $pla(h, \Sigma)$ can be derived from given (symbolic) quasi–supports, as soon as the numerical values of $p(QS(h, \Sigma))$, $p(QS(\sim h, \Sigma))$ and $p(QS(\perp, \Sigma))$ are determined (see Equations 3.22 and 3.23 of Subsection 3.3). Throughout this subsection we assume that assumptions are stochastically independent, although other probability models are also possible. Independence between assumptions means that the probabilities on $\{0, 1\}^{|A|}$ are given by (3.20).

Suppose that $QS(h, \Sigma)$ is given by its minimal elements $\mu QS(h, \Sigma)$.

Then, $qs(h,\Sigma) = \vee\{a \in \mu QS(h,\Sigma)\}$ is a possible logical representation of $QS(h,\Sigma)$, and we have

$$p(QS(h,\Sigma)) = p(N_A(qs(h,\Sigma))) = p(\cup\{N(a) :\ a \in \mu QS(h,\Sigma)\}). \tag{5.1}$$

Thus, the probability of a union of sets has to be computed. This is a well known problem, especially in reliability theory. There are essentially two methods. The first one is the so–called inclusion–exclusion method which is presented in the following theorem:

Theorem 5.1 *(Feller, 1968)*

$$p\Big(\bigcup_{i=1}^{t} N(a_i)\Big) = \sum_{k=1}^{t} (-1)^{k+1} \cdot S_k \tag{5.2}$$

where

$$S_k = \sum \Big\{ p\Big(\bigcap_{i \in I} N(a_i)\Big) :\ I \subseteq \{1,\ldots,t\},\ |I| = k \Big\}. \tag{5.3}$$

The probabilities of the intersections $\cap\{N(a_i) :\ i \in I\}$ are easy to compute because $p(\cap\{N(a_i) :\ i \in I\}) = p(N(\wedge\{a_i :\ i \in I\}))$ and $\wedge\{a_i :\ i \in I\}$ is a conjunction of literals from A. But in the model with independent assumptions, the probability of such a conjunction $a = \wedge \ell_j$ is simply

$$\begin{aligned} p(a) &= p(N(\wedge \ell_j)) = p(\cap N(\ell_j)) \\ &= \prod\{p_i :\ \ell_j = a_i\} \cdot \prod\{(1-p_i) :\ \ell_j = \sim a_i\}. \end{aligned} \tag{5.4}$$

Unfortunately, the S_k may contain many terms which makes the complete computation of (5.2) tedious. However, the partial sums of the complete sum are alternating lower and upper bounds of $p(\cup\{N(a_i) :\ i = 1,\ldots,t\})$, as the next theorem states.

Theorem 5.2 *(Feller, 1968; Kohlas & Monney, 1994) If ℓ is a positive integer such that $2\ell + 1 < t$, then*

$$\sum_{k=1}^{2\ell} (-1)^{k+1} \cdot S_k \ \leq p\Big(\bigcup_{i=1}^{t} N(a_i)\Big) \ \leq \sum_{k=1}^{2\ell+1} (-1)^{k+1} \cdot S_k. \tag{5.5}$$

Sometimes, if the bounds guarantee a sufficient approximation, these bounds permit to stop the computation of the complete sum at a relatively small value of $2\ell + 1$. However, note that these bounds are not monotone in ℓ.

The second general approach to compute probabilities of unions transforms the union first into a union of disjoint sets. This is best done in the logical setting where a disjunctive form $\vee\{a_i : i = 1, \ldots, t\}$ is developed into an equivalent disjunction $\vee\{d_j : j = 1, \ldots, s\}$ of disjoint formulae, $d_j \wedge d_k = \perp$, if $j \neq k$. Then, we have

$$p\Big(\bigvee_{i=1}^{t} a_i\Big) = p\Big(\bigvee_{j=1}^{s} d_j\Big) = p\Big(\bigcup_{j=1}^{s} N(d_j)\Big) = \sum_{j=1}^{s} p(N(d_j)), \tag{5.6}$$

and the probability of the union becomes again a simple sum. This method presupposes that $p(N(d_j))$ is easy to compute.

Abraham (1979) proposed an algorithm to develop $\vee\{a_i : i = 1, \ldots, t\}$ into an equivalent disjoint disjunction $\vee\{d_j : j = 1, \ldots, s\}$ where d_j are still conjunctions of literals of assumptions. This means that $p(N(d_j))$ can easily be computed according to (5.4). Heidtmann (1989) introduced an alternative method where the d_j are conjunctions of conjunctions and negations of conjunctions of literals of assumptions. In general, this method yields less terms than Abraham's method and the probabilities $p(N(d_j))$ are still simple to compute. Both methods were developed for the special case of so–called monotone Boolean systems. However, they can be adapted to the more general case presented here (Kohlas & Monney, 1994).

Let's note that reliability theory proposes also a number of bounds for the probability of a union (Barlow & Proschan, 1975; Kohlas, 1987). However, these bounds are again developed for monotone Boolean systems. The question is still open whether and how these bounds can be adapted to general non–monotone systems.

5.2 Numerical Valuation Network Computations

Instead of first computing symbolic arguments, one may also compute more or less directly numerical degrees of support and plausibility relative to an assumption–based knowledge. In particular, this can been done in the framework of valuation networks.

Suppose the knowledge (Σ, A, P) is decomposed into (Σ_i, A_i, P_i), $i = 1, \ldots, k$, such that $\Sigma = \cup\Sigma_i$, $A = \cup A_i$, $P = \cup P_i$, and, in addition, the sets A_i are disjoint (see Section 4.3). Then, we can compute bpa_{P_i} for each factor knowledge (Σ_i, A_i, P_i), $i = 1, \ldots, k$, by the methods described so far.

It can be shown that these bpa's combine by Dempster's rule, and marginalize as usual with bpa's. Furthermore, it can be shown that these bpa's are valuations in the sense of (Lauritzen & Shenoy, 1996) satisfying the axioms needed for local combination. This gives a way to combine numericaly bpa's and then to compute degrees of support and plausibility, which corresponds to the model of belief networks.

As the degrees of support are essentially reliabilities of proofs for some hypotheses, there are other methods of reliability theory, which can be considered to compute them (Kohlas, 1984). One such method is the so–called factorization method based on the formula of total probability (Cardona, 1993).

References

Abraham, J.A. 1979. An Improved Algorithm for Network Reliability. *IEEE Transactions on Reliability*, **28**, 58–61.

Anrig, B., & Haenni, R. 1996. *The General ATMS Algorithm.* Tech. Rep. Institute of Informatics, University of Fribourg.

Barlow, R.E., & Proschan, R. 1975. *Statistical Theory of Reliability and Life Testing.* New York.

Birkhoff, G., & Bartee, T.C. 1970. *Modern Applied Algebra.* Mc-Graw Hill, New York.

Cardona, L. 1993. *Die Behandlung von Inferenznetzen mit unzuverlässigen Regeln anhand der Theorie der Hinweise.* Ph.D. thesis, Institut für Informatik, Universität Freiburg.

Chang, C.L., & Lee, R.C.T. 1973. *Symbolic and Mechanical Theorem Proving.* Boston.

de Kleer, J. 1986. An Assumption-based TMS. *Artificial Intelligence, Elsevier Science Publisher B.V. (Amsterdam)*, **28**, 127–162.

Feller, W. 1968. *An Introduction to Probability Theory and its Applications.* Third edn. Vol. 1. Wiley.

Haenni, R. 1996. *Propositional Argumentation Systems and Symbolic Evidence Theory.* Ph.D. thesis, Institute of Informatics, University of Fribourg, Switzerland.

Heidtmann, K.D. 1989. Smaller Sums of Disjoint Products by Subproduct Inversion. *IEEE Transactions on Reliability*, **38**(3), 305–311.

Inoue, K. 1991. An Abductive Procedure for the CMS/ATMS. *Pages 34–53 of:* Martins, J.P., & Reinfrank, M. (eds), *Truth Maintenance Systems, Lecture Notes in A.I.* Springer.

Inoue, K. 1992. Linear resolution for consequence finding. *Artificial Intelligence, Elsevier Science Publisher B.V. (Amsterdam)*, **56**, 301–353.

Kohlas, J. 1984. The Reliability of Reasoning with Unreliable Arguments. *Annals of Operations Research*, **32**, 67–113.

Kohlas, J. 1987. *Zuverlässigkeit und Verfügbarkeit.* Teubner.

Kohlas, J. 1993. *Symbolic Evidence, Arguments, Supports and Valuation Networks.* Tech. Rep. 93-03. Institute of Informatics, University of Fribourg.

Kohlas, J. 1995. Mathematical Foundations of Evidence Theory. *Pages 31–64 of:* G. Coletti, D. Dubois, & Scozzafava, R. (eds), *Mathematical Models for Handling Partial Knowledge in Artificial Intelligence.* Plenum Press.

Kohlas, J. 1996. *Focussing: Cost-Bounded Assumption-Based Reasoning and Most Likely Arguments.* Tech. Rep. Institute of Informatics, University of Fribourg.

Kohlas, J., & Besnard, P. 1995. *An Algebraic Study of Argumentation Systems and Evidence Theory.* Tech. Rep. 95-13. Institute of Informatics, University of Fribourg.

Kohlas, J., & Monney, P.A. 1994. *Probabilistic Assumption-Based Reasoning.* Tech. Rep. 94-22. Institute of Informatics, University of Fribourg.

Kohlas, J., & Monney, P.A. 1995. *A Mathematical Theory of Hints. An Approach to the Dempster-Shafer Theory of Evidence.* Lecture Notes in Economics and Mathematical Systems, vol. 425. Springer.

Kohlas, J., & Moral, S. 1995. *Propositional Information System.* Tech. Rep. 96–01. Institute of Informatics, University of Fribourg.

Laskey, K.B., & Lehner, P.E. 1989. Assumptions, Beliefs and Probabilities. *Artificial Intelligence, Elsevier Science Publisher B.V. (Amsterdam)*, **41**, 65–77.

Lauritzen, S.L., & Shenoy, P.P. 1996. Computing Marginals Using Local Computation. *In:* Kohlas, J., & Moral, S. (eds), *Defeasible Reasoning and Uncertainty Management Systems: Algorithms.* Oxford University Press.

Ngair, T.H. 1992. *Convex Spaces as an Order-theoretic Basis for Problem Solving.* Ph.D. thesis, University of Pennsylvania.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Publ. Inc.

Provan, G.M. 1990. A Logic-Based Analysis of Dempster-Shafer Theory. *International Journal of Approximate Reasoning*, **4**, 451–495.

Reiter, R., & de Kleer, J. 1987a. Foundations of Assumption-Based Truth Maintenance Systems. *Proceedings of the American Association in AI*, 183–188.

Reiter, R., & de Kleer, J. 1987b. Foundations of Assumption-Based Truth Maintenance Systems. *Automated Reasoning*, 183–188.

Shafer, G. 1976. *The Mathematical Theory of Evidence.* Princeton University Press.

Shafer, G. 1979. Allocations of Probablity. *Ann. of Prob.*, **7**, 827–839.

Siegel, P. 1987. *Représentation et Utilisation de la Connaissance en Calcul Propositionel.* Ph.D. thesis, Université d'Aix-Marseille II. Luminy, France.