Fuzzy decision diagrams for the representation, analysis, and optimization of rule bases

Author(s):
Strehl, Karsten; Moraga, Claudio; Temme, Karl-Heinz; Stancović, Radomir

Publication Date:
1999

Permanent Link:
https://doi.org/10.3929/ethz-a-004287248

Rights / License:
In Copyright - Non-Commercial Use Permitted
Fuzzy Decision Diagrams for the Representation, Analysis, and Optimization of Rule Bases

Karsten Strehl
Computer Engineering and Networks Lab (TIK)
Swiss Federal Institute of Technology (ETH)
Gloriastrasse 25, 8092 Zurich, Switzerland
eMail: strehl@tik.ee.ethz.ch
WWW: http://www.tik.ee.ethz.ch/~strehl

Claudio Moraga, Karl-Heinz Temme
Chair of Computer Science I
University of Dortmund
Otto-Hahn-Straße 16, 44221 Dortmund, Germany
eMail: {moraga | temme}@LS1.cs.uni-dortmund.de
WWW: http://ls1-www.informatik.uni-dortmund.de

Radomir Stancović
Faculty of Electronic Engineering
University of Niš
Beogradska 14, 18000 Niš, Yugoslavia
eMail: stankovic@503c1.elfak.ni.ac.yu
WWW: http://503c1.elfak.ni.ac.yu/lab/radomir.htm

TIK Report
No. 77, September 1999

also: Research Report No. 725 (1999),
Department of Computer Science, University of Dortmund
Abstract

When no expert knowledge is available, fuzzy if-then rules may be extracted from examples of performance of a system. For this, an a priori decision on the number of linguistic terms of the linguistic variables may be required. This may induce a "rigid granularity", usually finer than that actually required by the system. Fuzzy Decision Diagrams are introduced as an efficient data structure to represent fuzzy rule bases and to systematically check their completeness and consistency. Moreover if the hypothesis of rigid granularity holds, reordering of the variables of a Fuzzy Decision Diagram may lead to a compacter and more precise rule base. The concept of reconvergent subgraphs is introduced to support the search for effective reorderings.

1. Introduction

BDDs are efficient data structures for the representation of switching functions [Lee 59], [Bry 86], [SaF 96]. The ordering of the variables affect however the size of a BDD, but the problem of obtaining the optimal ordering is NP-complete [THY 93]. DDs with interval labelled edges have been introduced for the efficient representation of multi-valued functions in the area of symbolic formal verification in [StT 98], [StT 99]. A spectral interpretation of DDs has been developed [SSM 96], [Sta 98], and BDDs have made possible the computation of the Walsh transform of functions with many variables [CMZ 93]. The name "Fuzzy Decision Diagrams" seems to have been used for the first time in [HTHY 95]; however these diagrams are really especial ternary DDs, where two of the three values of their variables are crisp intervals. The concept of Fuzzy Decision Diagrams (FuDDs) used in this work is quite different, as will be seen below. The involved variables are linguistic ones and their values are linguistic terms specified by fuzzy sets. Moreover these variables are not required to have the same number of values. A FuDD is introduced as a graph(ical) representation of a fuzzy if-then rule base.

The paper is organized as follows. In section 2, fuzzy decision diagrams are defined and explained. An analysis on the applications of FuDDs based on examples is presented in section 3, where the hypothesis of rigid granularity is introduced. A short section with conclusions completes the paper.

2. Fuzzy Decision Diagrams

Definition 1: A fuzzy set is a collection of different elements (from a given universe), each of which has a possibly different degree of membership to the set. If the same symbol is used for a fuzzy set and its characteristic function, then given a universe U, a fuzzy set S is characterized by the mapping:

\[ S: U \rightarrow [0,1] \]  (1)
For any \( u \in U \), \( S(u) \) gives its degree of membership to the fuzzy set \( S \).

A \textit{singleton} is a fuzzy set with only one element having a degree of membership 1, while all others have a degree of membership 0. If \( S \) is a singleton and \( s \) is its defining element, then the following holds:

\[
S(s) = 1 \quad \text{and} \quad \forall u \in U \text{ such that } u \neq s, S(u) = 0
\]  

(2)

Linguistic variables were introduced in [Zad 75] as a formal way of working with concepts associated to real-world variables.

Given a physical variable with a numerical domain, it is possible to define another view of this variable as a linguistic one, with a linguistic domain consisting of a set of fuzzy sets called linguistic terms and having own names. The fuzzy sets formally give the interpretation of the concepts expressed by the names of the linguistic terms upon the numerical domain.

Linguistic variables considered in this paper satisfy the conditions stated below.

If \( v \) is a variable with a numerical domain \( D_n \) and a linguistic domain \( D_l = \{ T_1, \ldots, T_m \} \), where \( T_j, 1 \leq j \leq m \), are the fuzzy sets of the corresponding linguistic terms, then:

i) \( \forall x \in D_n \sum_j T_j(x) = 1 \)

ii) \( \forall 1 \leq j \leq m \exists x \in D_n : T_j(x) = 1 \)

A Fuzzy Decision Diagram FuDD is a decision diagram for the efficient \textit{representation} and \textit{manipulation} of fuzzy rule bases, where the variables are linguistic (or fuzzy set valued) variables and the edges are labelled by the corresponding linguistic terms. In this sense, a FuDD is a generalization of a multiterminal decision diagram MTDD [SSMS 94], [StS 98] as well as of an interval decision diagram (IDD) [StT 98], [StT 99] except that the labels (fuzzy sets) of neighbour edges are not totally disjoint as for IDDs. The structure of a FuDD is the basis for its interpretation (see below).

Let \( v_i, 1 \leq i \leq N \), be a linguistic variable with \( n_i \) linguistic terms \( T_{ij} \). Hence, \( v_i \) has the linguistic domain \( D_i = \{ T_{i1}, T_{i2}, T_{i3}, \ldots, T_{in_i} \} \). FuDDs are represented by \textit{function graphs}, similar to those of [Bry 86].

\textbf{Definition 2:} A function graph \( G \) is a rooted, directed acyclic graph with an edge set \( E \) and a node set \( V \) which contains two types of nodes. A \textit{non-terminal} node \( v \in V \) has as attributes an argument index \( i = \text{index}(v), i < N \), which corresponds to the index of the linguistic variable \( v_i \), and \( n_i \) children \( \text{child}_j(v) \in V, 1 \leq j \leq n_i \). The linguistic terms \( T_{ij} \in D_i \) of the linguistic variable \( v_i \) are assigned to the
corresponding graph edges \((v, \text{child}_j(v)) \in E\). \(V\) contains \(n_N\) terminal nodes with index \(N\) and labelled with the linguistic terms \(T_{Nj}\) of \(v_N\). Hence, a terminal node \(v\) has as attribute a fuzzy set value \(\nu(v) \in D_N\).

A fuzzy if-then rule has basically the structure "if \(<condition>\) then \(<conclusion>\)", where \(<condition>\) and \(<conclusion>\) are fuzzy sets, not necessarily in the same universe. In order to use such a rule, the following generalized modus ponens [MaS 75] is considered:

\[
\begin{array}{c}
A' \\
B' \\
\end{array} \\
\frac{A \rightarrow B}{A'}
\]

where \(A'\) is a fuzzy set not precisely equal but similar to \(A\). The expected conclusion \(B'\) will be a fuzzy set not necessarily equal but similar to \(B\). This process has been formalized by means of the Compositional Rule of Inference [Zad 73]. A generalized (pointwise) expression for the Compositional Rule of Inference is the following:

\[
\forall w \text{ in the universe of } B \text{ and } u \text{ in the universe of } A \\
B'(w) = \sigma_u \{ \tau(A'(u), I(A(u), B(w))) \}
\]

where \(\tau\) denotes a t-norm [Men 42], [ScS 83], [Web 83], \(\sigma\) a t-conorm, \(\sigma_u\) a \(\sigma\)-based supremum in the universe of \(A\) and \(I(A(u), B(w))\) denotes an implication operation [TrV 85].

If \(A'\) is a singleton such that \(A'(u_0)=1\) and \(A'(u)=0\) for all \(u \neq u_0\) then \(\forall w\) in the universe of \(B\) the following holds:

\[
B'(w) = \sigma_u \{ \tau(0, I(A(u), B(w))) \} \text{ or } \tau(1, I(A(u_0), B(w)))
\]

If the "\(A \rightarrow B\)" rule of eq. (3) were to be applied to linguistic variables it would turn into "\(T_{1g} \rightarrow T_{Nk}\)\", where from the indices it is clear, that from the \(g\)-th linguistic term of \(v_1\) as condition follows the \(k\)-th linguistic term of \(v_N\) as conclusion. Since however in practice the linguistic terms will not be identified with abstract symbols with indices, but with conceptual labels, then the rules will be written with explicit mention of the corresponding variables. This leads to the following structure for the above rule:

\[
R: \quad \text{if } v_1 \text{ is } T_{1g} \text{ then } v_N \text{ is } T_{Nk}
\]
Example 1:

Consider the linguistic variable *external temperature* with the linguistic terms \{freezing, cold, normal\} and the linguistic variable *heating* with the linguistic terms \{low, middle, high\}. A possible set of rules to control the room temperature could be:

R1: if the external temperature is freezing then the heating is –(should be set to)– high
R2: if the external temperature is cold then the heating is –(should be set to)– middle
R3: if the external temperature is normal then the heating is –(should be set to)– low

How reasonable these rules are, will of course depend on the definitions of freezing, cold, normal as well as of low, middle and high. Furthermore, it becomes apparent that the rule base may be seen as a (fuzzy) case statement.

An especial feature of a fuzzy rule base (compared to other types of rule bases) is that for a given crisp input (i.e. for a given numerical value of the linguistic variable), more than one rule may be activated and these rules must be aggregated to obtain the final decision. (See figure 1 where this is illustrated for the case where the product is selected as implication, minimum as intersection, and the aggregation is realized by computing (pointwise) the maximum, a selection of operations that is frequently chosen for applications in fuzzy control.)

A fuzzy decision diagram representing this extremely simple example is shown in Figure 2.

The <condition> of a rule may be an expression of a conjunction of (simpler) conditions (textually expressed by the word *and*). Such rules exhibit the following structure:

\[ R: \text{ if } v_1 \text{ is } T_{1g} \text{ and } ... \text{ and } v_{N-1} \text{ is } T_{(N-1)q} \text{ then } v_N \text{ is } T_{Nk} \]

The conjunction of conditions is expressed by the hierarchy of nodes of a FuDD. A rule base is evaluated by traversing the FuDD from the leaves to the root. Along a path from a leaf to the root, the conclusion of a rule will be calculated. For this, the degree of satisfaction (i.e. the conjunction of the membership degrees of the actual numerical values of the variables to the corresponding linguistic terms) will be computed and considered together with the fuzzy set that labels the leaf to evaluate the output implication. Finally an aggregation should combine all activated rules.
3. Analysis of Applications

Example 2:

Let a first linguistic variable have the domain \{T_{11}, T_{12}, T_{13}\}, a second, the domain \{T_{21}, T_{22}\}
and a third, the domain \{T_{31}, T_{32}, T_{33}, T_{34}\}. It becomes apparent that \(T_{ij}\) denotes the \(j\)-th linguistic term
of the \(i\)-th linguistic variable. Consider the following rule base on these variables, with the inputs \(x_1\) and \(x_2\),
and the conclusion \(x_3\).

\[
\begin{align*}
R1: & \text{ if } x_1 \text{ is } T_{11} \text{ and } x_2 \text{ is } T_{21} \text{ then } x_3 \text{ is } T_{31} \\
R2: & \text{ if } x_1 \text{ is } T_{11} \text{ and } x_2 \text{ is } T_{22} \text{ then } x_3 \text{ is } T_{34} \\
R3: & \text{ if } x_1 \text{ is } T_{12} \text{ and } x_2 \text{ is } T_{21} \text{ then } x_3 \text{ is } T_{31} \\
R4: & \text{ if } x_1 \text{ is } T_{12} \text{ and } x_2 \text{ is } T_{22} \text{ then } x_3 \text{ is } T_{32} \\
R5: & \text{ if } x_1 \text{ is } T_{13} \text{ and } x_2 \text{ is } T_{21} \text{ then } x_3 \text{ is } T_{33} \\
R6: & \text{ if } x_1 \text{ is } T_{13} \text{ and } x_2 \text{ is } T_{22} \text{ then } x_3 \text{ is } T_{34}
\end{align*}
\]

A FuDD representing this rule base is shown in Figure 3. This FuDD belongs to the class of
ordered DDs [Bry 86], characterized by the fact that all nodes at a given layer are associated to the
same variable and no repetition of variables appears along any path from a leaf to the root. The order
of the linguistic variables in the FuDD corresponds to the order of appearance of the elementary
conditions in the rules. If t-norms are used to realize the conjunctions, then a reordering of the
variables giving an equivalent FuDD may lead to a more compact rule base. This is possible due to
the fact that t-norms are commutative and associative. Finding the optimal ordering of the variables
for a DD is however NP-hard [BoW 96], and several heuristics have been suggested to find possibly
sub-optimal orderings, but within a reasonable amount of computing time (see e.g. [FrS 87], [ISY 91],
[FOH 93], [Rud 93], [THY 93], [DrB 98], [RSM 99]). If non-commutative conjunctions [Pra 99] are
used, no optimization of the FuDD (in the former sense) is possible.

It becomes apparent that a FuDD allows checking completeness and consistency of a rule
base. All nodes at the same level must have as many leaving edges as the corresponding linguistic
variable has linguistic terms (completeness), and no two of these edges may have the same label
(consistency). It is simple to see that this kind of check does not actually require the FuDD to be
ordered. In a rule base with a few rules (as in example 2) this kind of control may be done by simply
inspecting the list of rules; however in the case of a rule base with a large number of rules or with
rules with a large conjunction of conditions this is cumbersome if at all possible; meanwhile the
FuDD representation supports a systematic test for a large dimension of the rule base.
If no expert knowledge is available to state the rules for a given problem, it is possible to learn the rules with the help of examples (see e.g. [Jan 92], [TSK 92], [WaM92], [THM 99]). Depending on the system being used, it may be the case that rules with too small granularity may be obtained. This is the typical case of a rule generation process, where an a priori decision on a fixed number of linguistic terms for each linguistic variable is taken and the conjunction of premisses is based on t-norms. This is illustrated in figure 4. There, a problem space which exhibits five blocks and accordingly could be partitioned by using five rules, leads to nine rules if a rule generation process as mentioned above is used. It is easy to see that the optimal partition could be obtained if at every surrounding block, some neighbour linguistic terms were replaced by their convex hull and given a proper interpretation. Since linguistic variables are structured as an ordered set of linguistic terms, then the predicates “larger than or equal” as well as “smaller than or equal” would provide an adequate interpretation for the convex hulls covering the corresponding linguistic terms. Please notice that the new rule base is not semantically equivalent to the former one, but covers it and under the above “hypothesis of rigid granularity”, it has an improved accuracy –(lower mean square error)– since it supresses really non-existing “valleys” artificially produced between neighbour fuzzy blocks with the same label. It should be pointed out that this criterion has been applied (by other authors) in the past, to situations where in the matrix representation of a rule base, neighbour cells have the same label, implicitly assuming that this situation is due to the above discussed hypothesis of rigid granularity (see e.g. [Gut 93], [GuS 97]).

Example 3 below shows how a FuDD points out this problem and suggests ways of optimizing the rules, i.e. a rule base with less, more accurate and possibly more compact rules may be obtained. To simplify the FuDD representations, it will be agreed that edges leaving a node will be labelled (from left to right) in the same order as the linguistic terms of the corresponding linguistic variable.

Example 3:

Consider a rule base with two conjuncted conditions, having three and four linguistic terms respectively, and a conclusion with five linguistic terms. Assume that the prevailing rule base is the following:

\[
\begin{align*}
R1 : & \text{ if } x_1 \text{ is } T_{11} \text{ and } x_2 \text{ is } T_{21} \text{ then } x_3 \text{ is } T_{31} \\
R2 : & \text{ if } x_1 \text{ is } T_{11} \text{ and } x_2 \text{ is } T_{22} \text{ then } x_3 \text{ is } T_{33}
\end{align*}
\]
R3 : if \( x_1 \) is \( T_{11} \) and \( x_2 \) is \( T_{23} \) then \( x_3 \) is \( T_{33} \)
R4 : if \( x_1 \) is \( T_{11} \) and \( x_2 \) is \( T_{24} \) then \( x_3 \) is \( T_{33} \)
R5 : if \( x_1 \) is \( T_{12} \) and \( x_2 \) is \( T_{21} \) then \( x_3 \) is \( T_{31} \)
R6 : if \( x_1 \) is \( T_{12} \) and \( x_2 \) is \( T_{22} \) then \( x_3 \) is \( T_{32} \)
R7 : if \( x_1 \) is \( T_{12} \) and \( x_2 \) is \( T_{23} \) then \( x_3 \) is \( T_{34} \)
R8 : if \( x_1 \) is \( T_{12} \) and \( x_2 \) is \( T_{24} \) then \( x_3 \) is \( T_{34} \)
R9 : if \( x_1 \) is \( T_{13} \) and \( x_2 \) is \( T_{21} \) then \( x_3 \) is \( T_{35} \)
R10 : if \( x_1 \) is \( T_{13} \) and \( x_2 \) is \( T_{22} \) then \( x_3 \) is \( T_{35} \)
R11 : if \( x_1 \) is \( T_{13} \) and \( x_2 \) is \( T_{23} \) then \( x_3 \) is \( T_{34} \)
R12 : if \( x_1 \) is \( T_{13} \) and \( x_2 \) is \( T_{24} \) then \( x_3 \) is \( T_{34} \)

The FuDD is shown in figure 5. It becomes quite apparent, that a simplification (i.e. optimization) of the rule base is possible. The following new rules may be extracted from the FuDD:

R13: if \( x_1 \) is \( T_{11} \) and \( x_2 \) is larger than or equal to \( T_{22} \) then \( x_3 \) is \( T_{33} \)
R14: if \( x_1 \) is \( T_{12} \) and \( x_2 \) is larger than or equal to \( T_{23} \) then \( x_3 \) is \( T_{34} \)
R15: if \( x_1 \) is \( T_{13} \) and \( x_2 \) is larger than or equal to \( T_{23} \) then \( x_3 \) is \( T_{34} \)
R16: if \( x_1 \) is \( T_{13} \) and \( x_2 \) is smaller than or equal to \( T_{22} \) then \( x_3 \) is \( T_{35} \)

R13 summarizes R2, R3 and R4; R14 reduces R7 and R8; R15 comprises R11 and R12, and finally R16 represents R9 and R10.

Further optimization is still possible if the FuDD exhibits a reconvergent subgraph. A subgraph will be called reconvergent if it consists of different paths from a reference node to a given leaf or set of leaves. These paths satisfy two conditions:

i) they traverse only neighbour children-nodes of the reference node.

ii) starting at the level below the reference node, all graphs connecting the selected children of the reference node with the predefined set of leaves are isomorphic. Notice that if starting at the level below the reference node the reconvergent paths include all edges, then the FuDD may be reduced [Bry 86].

If a FuDD has a reconvergent subgraph, a proper reordering of the variables leads to an equivalent FuDD with an explicit suggestion for the optimization of rules. See figure 6 that illustrates the reconvergent subgraphs of the FuDD in figure 5, and figure 7, for an equivalent FuDD with a reordering of the variables. The new reconvergent subgraphs are illustrated in figure 8 and the reduced FuDD is presented in figure 9.
Now it is possible to obtain two new (optimized) rules:

R17: if $x_2$ is $T_{21}$ and $x_1$ is smaller than or equal to $T_{12}$ then $x_3$ is $T_{31}$

R18: if $x_2$ is larger than or equal to $T_{23}$ and $x_1$ is larger than or equal to $T_{12}$ then $x_3$ is $T_{34}$

R17 combines R1 and R5, meanwhile R18 is the result of R14 and R15. It should be noticed that R18 is obtained directly from the reduced FuDD.

Up to now, ordered FuDDs have been considered. If all paths from the root to the leaves of a FuDD are independently reordered, the result will be a free FuDD [BGMS 94]. The use of free FuDDs to optimize a rule base will be illustrated in the next example.

Example 4:

Assume that the following rule base is available. (To simplify the example, but without loss of generality, linguistic variables with only two linguistic terms are considered).

R1: if $x_1$ is $T_{11}$ and $x_2$ is $T_{21}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{41}$ then $x_5$ is $T_{51}$

R2: if $x_1$ is $T_{11}$ and $x_2$ is $T_{21}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{42}$ then $x_5$ is $T_{52}$

R3: if $x_1$ is $T_{11}$ and $x_2$ is $T_{21}$ and $x_3$ is $T_{32}$ and $x_4$ is $T_{41}$ then $x_5$ is $T_{52}$

R4: if $x_1$ is $T_{11}$ and $x_2$ is $T_{31}$ and $x_3$ is $T_{32}$ and $x_4$ is $T_{42}$ then $x_5$ is $T_{54}$

R5: if $x_1$ is $T_{11}$ and $x_2$ is $T_{23}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{41}$ then $x_5$ is $T_{51}$

R6: if $x_1$ is $T_{11}$ and $x_2$ is $T_{22}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{42}$ then $x_5$ is $T_{53}$

R7: if $x_1$ is $T_{11}$ and $x_2$ is $T_{31}$ and $x_3$ is $T_{32}$ and $x_4$ is $T_{42}$ then $x_5$ is $T_{53}$

R8: if $x_1$ is $T_{11}$ and $x_2$ is $T_{22}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{42}$ then $x_5$ is $T_{53}$

R9: if $x_1$ is $T_{11}$ and $x_2$ is $T_{22}$ and $x_3$ is $T_{32}$ and $x_4$ is $T_{42}$ then $x_5$ is $T_{55}$

R10: if $x_1$ is $T_{11}$ and $x_2$ is $T_{22}$ and $x_3$ is $T_{32}$ and $x_4$ is $T_{44}$ then $x_5$ is $T_{53}$

R11: if $x_1$ is $T_{12}$ and $x_2$ is $T_{23}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{41}$ then $x_5$ is $T_{52}$

R12: if $x_1$ is $T_{12}$ and $x_2$ is $T_{23}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{42}$ then $x_5$ is $T_{55}$

R13: if $x_1$ is $T_{12}$ and $x_2$ is $T_{23}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{44}$ then $x_5$ is $T_{53}$

R14: if $x_1$ is $T_{12}$ and $x_2$ is $T_{32}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{41}$ then $x_5$ is $T_{53}$

R15: if $x_1$ is $T_{12}$ and $x_2$ is $T_{32}$ and $x_3$ is $T_{31}$ and $x_4$ is $T_{42}$ then $x_5$ is $T_{54}$

R16: if $x_1$ is $T_{12}$ and $x_2$ is $T_{32}$ and $x_3$ is $T_{32}$ and $x_4$ is $T_{42}$ then $x_5$ is $T_{55}$
If the lexicographic order used to write the conditions of the rules (i.e. $x_1$, $x_2$, $x_3$, $x_4$) is used to generate the corresponding FuDD, the result would be as shown in figure 10. A reconvergent subgraph may be recognized, which suggests a reordering of the variables. Figure 11 shows the reordered FuDD, where the variables are ordered as $x_3$, $x_4$, $x_1$ and $x_2$. A new reconvergent subgraph rooted at $x_1$ in the middle of the FuDD may be recognized. Local reordering leads to the reduced free FuDD shown in figure 12. It becomes apparent that this FuDD has only 9 internal nodes meanwhile the former had 10 and the original FuDD 15.

The rules extracted from the reduced free FuDD are:

R1': if $x_3$ is $T_{31}$ and $x_4$ is $T_{41}$ then $x_5$ is $T_{51}$
R2': if $x_3$ is $T_{31}$ and $x_4$ is $T_{42}$ and $x_1$ is $T_{11}$ and $x_2$ is $T_{21}$ then $x_5$ is $T_{52}$
R3': if $x_3$ is $T_{31}$ and $x_4$ is $T_{42}$ and $x_1$ is $T_{11}$ and $x_2$ is $T_{22}$ then $x_5$ is $T_{53}$
R4': if $x_3$ is $T_{31}$ and $x_4$ is $T_{42}$ and $x_1$ is $T_{12}$ then $x_5$ is $T_{53}$
R5': if $x_3$ is $T_{32}$ and $x_4$ is $T_{41}$ and $x_2$ is $T_{21}$ then $x_5$ is $T_{52}$
R6': if $x_3$ is $T_{32}$ and $x_4$ is $T_{41}$ and $x_2$ is $T_{22}$ and $x_1$ is $T_{11}$ then $x_5$ is $T_{53}$
R7': if $x_3$ is $T_{32}$ and $x_4$ is $T_{41}$ and $x_2$ is $T_{22}$ and $x_1$ is $T_{12}$ then $x_5$ is $T_{54}$
R8': if $x_3$ is $T_{32}$ and $x_4$ is $T_{42}$ and $x_1$ is $T_{11}$ and $x_2$ is $T_{21}$ then $x_5$ is $T_{54}$
R9': if $x_3$ is $T_{32}$ and $x_4$ is $T_{42}$ and $x_1$ is $T_{11}$ and $x_2$ is $T_{22}$ then $x_5$ is $T_{55}$
R10': if $x_3$ is $T_{32}$ and $x_4$ is $T_{42}$ and $x_1$ is $T_{12}$ then $x_5$ is $T_{55}$

It becomes apparent that the new rule base is shorter and some of the rules are simpler than in the original rule base.

Definition 4:

A partially ordered FuDD is a free decision diagram that is strictly ordered from the root down to a certain intermediate level. (If the FuDD is reduced, all reduced redundant nodes are considered to be correctly ordered).

It is easy to see that the FuDD of figure 12 is partially ordered.

Up to now, neither the type of conjunction operation (except, perhaps, for the requirements on associativity and commutativity) nor the type of implication operation has been explicitly considered.
The FuDD representation is a general one. If in an ordered FuDD the rule base were of Mamdani type [MaA 75] (i.e. the implication of a rule and the conjunction used to combine the conditions are realized with the minimum operation, meanwhile the aggregation is done with the maximum operation), and the signals at the leaves were the result of other rule bases (also of Mamdani type), then the leaves would turn into roots of new FuDDs leading to a chaining of (partial) rules resulting in global rules with a (much) larger number of conjuncted conditions. The resulting FuDD would be ordered or partially ordered. The reverse situation is however more interesting. Given a Mamdani rule base leading to an ordered (or partially ordered) FuDD with a large depth, it is possible to extract a hierarchically ordered rule base by dividing the FuDD into two parts. A level will be selected at which the nodes will turn into leaves and a following part representing new rule bases feeding the new generated leaves. This idea is related to work on matrix-valued DDs [Sta 99]. For the time being, this is constrained to Mamdani systems since the distributivity between maximum and minimum is essential for this kind of decomposition: The leaves FuDDs would provide maximum-aggregated partial rules that must be conjuncted with the conditions along a path leading to the root. This will only then represent a new rule (set) if the result is equivalent to the maximum-aggregation of the conjunction of degrees of satisfaction of all conditions along the new compound paths. This is the reason why 1) the implication operation had to be chosen as a conjunction, so that the chaining of two rules can produce a new rule (of the same kind) and 2) the t-norm used for conjunction and the t-conorm used for aggregation had to be minimum and maximum, respectively, since this is the only pair of t-norm and t-conorm that is distributive [Web 83]. Finally, this is the characteristic of a Mamdani system.

Example 4: (Continuation)

The FuDD shown in figure 12 represents a reduced partially ordered one. If the rule base were of Mamdani type, the following decomposition would be possible:

i) Leaf rules

R1.1: if $x_1$ is $T_{11}$ and $x_2$ is $T_{21}$ then $y_{left}$ is $T_{52}$
R1.2: if $x_1$ is $T_{11}$ and $x_2$ is $T_{22}$ then $y_{left}$ is $T_{53}$
R1.3: if $x_1$ is $T_{12}$ then $y_{left}$ is $T_{53}$
R2.1: if $x_2$ is $T_{21}$ then $y_{middle}$ is $T_{52}$
R2.2: if $x_2$ is $T_{22}$ and $x_1$ is $T_{11}$ then $y_{middle}$ is $T_{53}$
R2.3: if $x_2$ is $T_{22}$ and $x_1$ is $T_{12}$ then $y_{middle}$ is $T_{54}$
R3.1: if \( x_1 \) is \( T_{11} \) and \( x_2 \) is \( T_{21} \) then \( \text{y}_{\text{right}} \) is \( T_{54} \)
R3.2: if \( x_1 \) is \( T_{11} \) and \( x_2 \) is \( T_{22} \) then \( \text{y}_{\text{right}} \) is \( T_{55} \)
R3.3: if \( x_1 \) is \( T_{12} \) then \( \text{y}_{\text{right}} \) is \( T_{55} \)

ii) Root rules

R0.1: if \( x_3 \) is \( T_{31} \) and \( x_4 \) is \( T_{41} \) then \( x_5 \) is \( T_{51} \)
R0.2: if \( x_3 \) is \( T_{31} \) and \( x_4 \) is \( T_{42} \) then \( x_5 \) is \( \text{y}_{\text{left}} \)
R0.3: if \( x_3 \) is \( T_{32} \) and \( x_4 \) is \( T_{41} \) then \( x_5 \) is \( \text{y}_{\text{middle}} \)
R0.4: if \( x_3 \) is \( T_{32} \) and \( x_4 \) is \( T_{42} \) then \( x_5 \) is \( \text{y}_{\text{right}} \)

The new rule base has 13 rules, but none of them has more than 2 conditions. Theses rules might be easier to interpret than the former ones.

4. Conclusions

Fuzzy Decision Diagrams provide an efficient data structure for adequate representation, analysis and optimization of if-then rule bases, independently of the operations used to evaluate the rules, under the hypothesis of rigid granularity. Moreover, FuDDs allow checking for completeness and consistency of the rule base. If the rule bases were obtained by predefining the number of linguistic terms of the linguistic variables and by using t-norms to realize the conjunction of premisses, then an optimization may be done by using techniques to minimize the corresponding FuDD. Both ordered FuDDs with reordering of variables and free FuDDs support the optimization. Efficient algorithms to do this are however still looked for. In the case of Mamdani rule bases, FuDDs allow additionally a simple decomposition of the rule base. This implies a chaining of rules [LTT 97], but the new rules use a reduced number of conditions, which may improve their interpretability.

Acknowledgements:

Part of the work leading to this paper was done during a short visit of C. Moraga to the Department of Computer Science of the University of Victoria, Canada.

The work of K.-H. Temme was partially supported by the German Research Society (DFG) as part of the Collaborative Research Center on Computational Intelligence (SFB531) at the University of Dortmund.
References


[THY 93] Tani S., Hamagushi K., Yajima S.: The complexity of the optimal variable ordering of a shared BDD. Technical Report 93-6, Department of Information Science, Faculty of Science, University of Tokyo, 1993


[Zad 75] Zadeh L.: The concept of a linguistic variable and its application to approximate reasoning. Information Sciences 8, 199–249, 301-357; 9, 43-80, 1975
Fig. 1: Activation of rules with product as implication and maximum as aggregation.

Fig. 2: FuDD for example 1

Fig. 3: FuDD representation for example 2
Fig. 4: A proper and a grid-like partition of a five-blocks space
Figure 5: FuDD for example 3

Fig. 6: Reconvergent subgraphs in the FuDD of figure 5
Fig. 7: Equivalent FuDD with reordered variables

Fig. 8: Reconvergent subgraphs of the FuDD of figure 7
Fig. 9: Reduced FuDD with reordered variables for example 3

Fig. 10: FuDD for example 4 showing a reconvergent subgraph
A new reconvergent subgraph within a single subgraph.

“Leaf-rules” for decomposition are indicated. (see text)