Digital solutions for inference rules in decision-supporting systems

Erika Matsak,
Tallinn University, Estonia
erika.matsak@tlu.ee

Peeter Lorents
Cooperative Cyber Defence Centre of Excellence
peeter.lorents@mil.ee

Abstract

In this paper, we discuss the use of digital circuits in decision support systems. For this, it is first necessary to translate the descriptions of situations and conclusions which are discovered step by step into a suitable Predicate Calculus. Then, a digital implementation for the constructs of Predicate Calculus must be found. So, the main issue is the realization of logically correct inference rules. In this paper , the following inference rules have been studied in a detailed fashion: Contrapository rule, the Modus Ponens and Modus Tollens.

Keywords. Decision-making support systems, formulas for Predicate Calculus, inference steps and inference rules, the realization of correct inference steps by digital circuits.

1. Introduction

One important advantage of modern intelligent control systems is the ability to help people in decision-making process, or ability to make the right decision automatically, without human intervention. For this purpose, a lot of tools have been designed, from simple tools like visualization utilities of decision trees [Trong Dung Nguyen, Tu Bao Ho, and Hiroshi Shimodaira 2000] to complex expert systems. In this context, let's mention the two major approaches: in the first case, seek and highlight (all) the possible ways for decision and the ensuing consequences, in the second case, employ a certain system of rules [Guilan Kong , Dong-Ling Xu , Xinbao Liu and Jian-Bo Yang . 2009]. Here the statistical methods are often used [M.F. Renzi, P. Vicard, R. Guglielmetti, F. Musella 2009]. Rules can be the based, for example, on the situations [J. McCarthy, P.J. Hayes 1969] or events [R. Kowalski, M. Sergot 1986].

The reliability of decision-support systems is deeply connected to the truth of decisions related to the correct logical formulas. One source of getting correct logical formulas is the implementation of inference rules. So far, the implementation of the inference rules (design of inference or proving) is has been realized mainly by means of software. [Rinard, 2009]. However, in addition to developing software solutions, recent papers have studied the opportunities of employing specialized hardware. [Vipan Kakkar 2009].

In this context, we reach a problem which we consider in this work: what kind of hardware would allow for the implementation of correct inference steps (including Modus Ponens (MP) and Modus Tollens (MT)), and would therefore reach reliable decisions faster and with lesser faults.

2. Logical instruments for decision-making process - example of a simple technical system

The next example which is considered in this section, is necessary for better understanding of the structure of following digital circuits and their linkages to technical systems. As an example, we have chosen a small heating system for residential usage, where important roles are played by the water boiler with the heater element, flaps, taps, switches and sensors.

The following things describe heating system events: the temperature t, pressure p, and the water level h. The minimum and maximum values are set for them (t_{min}, t_{max}; p_{min}, p_{max}; h_{min}, h_{max}). For changing the water temperature, the heater element should be used, which can be switched on or off. Boiler has the flap for the keeping of suitable pressure, which can be opened or closed. It is occasionally necessary to add water, or vice versa - reduce the amount of water in the boiler, and for this the relevant tap should be open or closed. The necessary decisions for the management of the system described aboveare derived from:

- Clauses which describe the situation. For example: the temperature is above of the maximum permitted value, or ($t > t_{max}$).
- Predefined implications (or clauses that have a "shapes": if A then B), where the clauses describing the events stay in antecedent and the clauses about taps, flaps and switches are

located in succedent. For example: *If the temperature is above the permitted maximum, then the heater must be turned off,* or $(t > t_{max}) \supset (Out(Heater))$.

- Application of inference rules. Each step has two premises. The first premise is a description of a situation. Another premise is an implication that has the same description of a situation as its antecedent. The result of the inference step is the succedent of this implication (i.e. a control decision, e.g. a statement about the position of a valve, a switch etc.).

The artificial intelligence systems usually use the ideas from the nature and we also decided to "copy" human argumentations to our digital developments. Namely – in [Matsak 2009a] it has been shown that children use the Modus Ponens and the Modus Tollens for presenting their arguments in quite early ages [Matsak 2009b]. We consider the implementation of these rules as an important step for the formation of decision making system.

Example 1. From the fact that

- the temperature threshold has been exceeded and
- the requirement that, if the temperature threshold has been exceeded, then the heater must be turned off

follows that the heater must be turned off.

$$\frac{(t > t_{max}) \qquad (t > t_{max}) \supset (Out(heater))}{Out(Heater)}$$

Example 2. From the fact that the heater is switched on (ie not off) and an agreement that, if the temperature threshold has been exceeded, then the heater must be switched off, we get the fact that the temperature threshold is not exceeded.

$$\frac{\neg Out(heater) \qquad (t > t_{max}) \supset (Out(heater))}{\neg(t > t_{max})}$$

Of course, not only MP and MT rules can be used in decision-making.

3. Components for the digital decision making

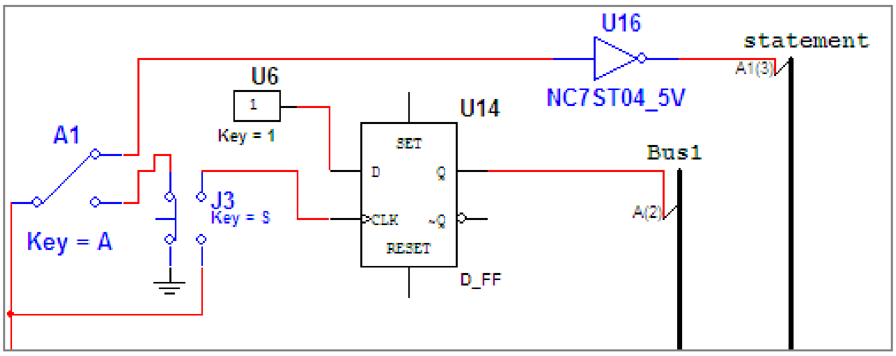
Next we will describe the technological aspects which are needed for the implementation of a decision making system. In this paper we do not explain how more complex systems could be created, what kind of knowledge base hierarchy is required, we just discuss how system elements could be presented and how the system uses the axioms and inference rules.

First, the experts must prepare a "base" with the knowledge and axioms for the decision making system. Here it is necessary to operate with atomic formulas in first order predicate calculation (where modus ponens and modus tollens belong). Formulas represent the knowledge [Lorents, 2009] - we have digital objects (triggers), which keep the 1(true) or 0(false) signal and every such trigger is connected to the memory, where the denotation is described. It should be noted that we do not use the traditional realization for the neurons which uses the RAM and programming part for operations [Redgers A., Aleksander I., 1995] - instead we have shifted our focus to the digital solution.

Every statement with a truth value 0 or 1 may be active or inactive statement. If a statement is active, then a user or automatic system must give the real truth value, based on some real situation. For example, lets assume that our example system is operating with a statement " system is overheated". In real situation automatic system discovers the events and yields the information about the statement "system is overheated", (i.e., is the statement true or false). But the value 0 and unknown value are not the same. The inactive statement is turned off and active is turned on by switch [drawing 1]. We use two buses: "Bus1" and "Statement". First of them keeps the information about the truth value (0 or 1) and second the information about active/inactive statement.

Let's limit our system to the very primitive: $\langle\{A, B\}\cup\{A\supset B\}, \{MP, MT\}\rangle$. In other words, lets assume our example system operates with two statements A and B, one axiom $A\supset B$, and two inference rules MP, MT. We will show how it is possible to deduct one statement from the other digitally.

For example, the formula A may be expressed as



Drawing 1. Digital part fo simple knowledge

(t > t_{max}) and formula B as Out(Heater) [see section above]

The premise of Modus Ponens consists of an atomic formula (for example A) and axiom (for example A⊃B) , where the atomic formula stands on the left part of an implication. The statement about B is not required.

The premise of Modus Tollens consists of atomic formulae and also an axiom, but in this inference rule we need the statement B, the statement A can be inactive.

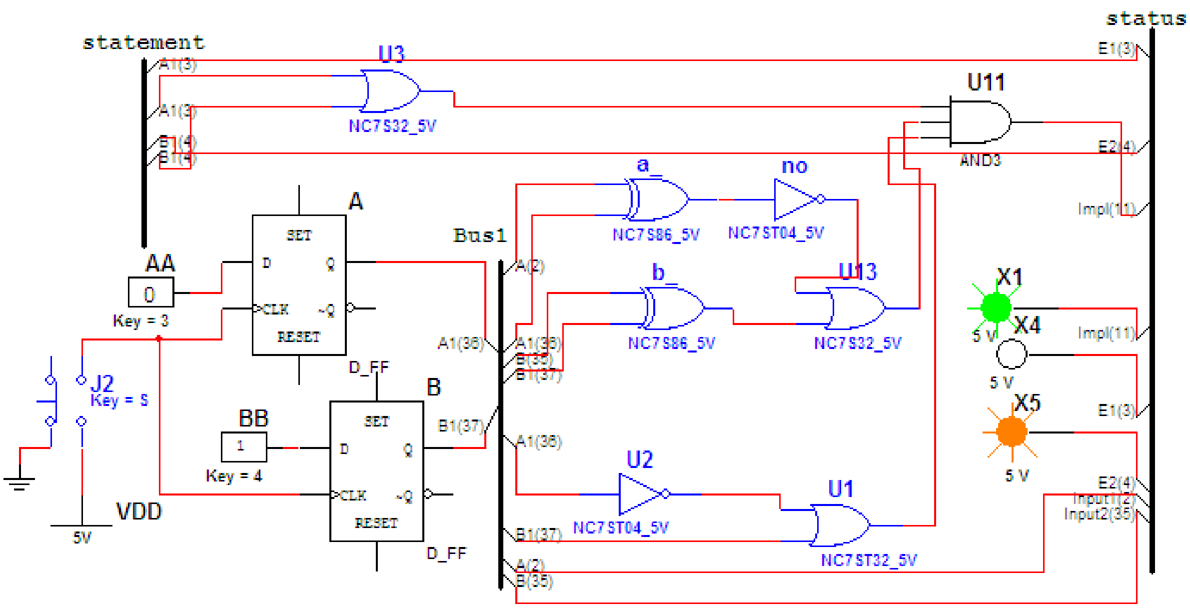
Note that we need axiom detection if one of atomic formulas is active [drawing 2]. If statement of activity for one of two elements is 1 and an axiom exists, then we have to check the truth values from inputted formula and the same formula inside axiom. If we discover formula A with the truth value 1, then the truth value of A inside implication must be also 1. But for formula B the situation must be opposite.

implication exists and is suitable for the Modus Ponens or Modus Tollens. Indicators have been added for the sake of clarity to drawing 2. Here, the indicator X1 depicts the implication signal, X4 the activity of formula A and X5 the activity of formula B. All necessary information for the inference rule is inputted to the bus “status”.

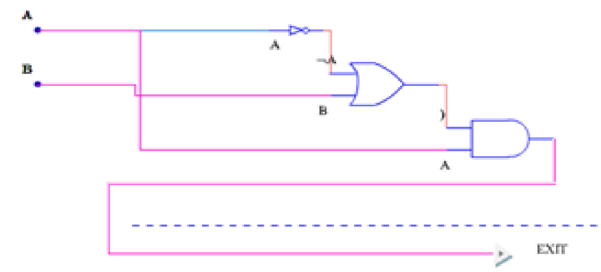
The information about which formula is active and is the relevant axiom available could be used in decoder (table 1).

The same decoding idea of cause may be realized in the system with more rules.

For inference steps we create digital circuits composed of three types of logic gate elements: negation, conjunction and disjunction. Each circuit is divided into two parts, where the first part represents the list of premises and the second part represents the conclusion formula [Matsak 2009c]. The use of the inference step therefore corresponds the moving from the first part of the circuit to the second part.



Drawing 2. Choosing the axiom



Drawing 3. Digital Modus Ponens

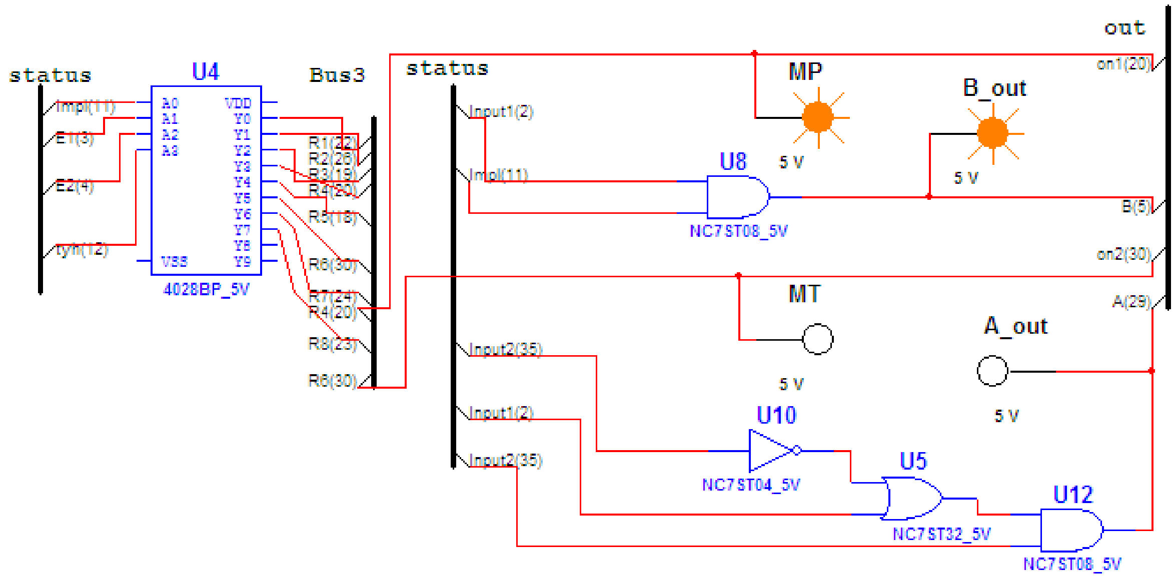
The AND gate with 3 inputs gives the 1 if the

Table 1. Decoding for the rules chosing

Implication	1	1	↑
A	1	0	
B	0	1	
Decimal	3	5	

For the Modus Ponens it is necessary to have formulae A and Implication asactive statements. The system decodes this signal as decimal “3” and moves to the inference step realization block. In this block the system accepts the premise and gives the output through a simple AND gate. Drawing 3 shows what gates are needed for the premise(these elements "come" from the knowledge base and from the axiom part). Bus "status" carries their signals to inference step. Here we need only the conjunction of involved parts and after that is possible to send the signal B to the output [Drawing 4].

The Modus Tollens inference rule can be presented as a combination of the Contrapository rule and Modus Ponens. Actually, if the formula B is active and an axiom is chosen, then we move through the following steps:



Drawing 4. Realization of inference steps

$$\frac{\neg B \qquad \dfrac{A \supset B}{\neg B \supset \neg A}}{\neg A}$$

The digital system also uses two steps for the Modus Tollens realization, and as it shown on drawing 4 for the Contropository rule realization we just have to exchange inputs in the same implication circuit.

For the design of more complex decision-supporting digital systems we can use the cascades of circuits described above. If we would like to increase the effectiveness of the system, then it would be necessary to connect every axiom circuit to the personal block with inferences, in another words split a decision process flow into a multiple parallel flows (see drawing 4).

4. Conclusions

In this work, we have explained how to design and implement decision support systems by means of digital circuits. For that approach, the decisions and necessary steps are presented in a formal way, and are limited by first-order Predicate Calculus. The correct formulas in Predicate Calculus can be obtained by using the correct inference rules. Although the formulae representation by digital circuits is a well-known concept , the realization of inference steps with digital circuits has not received much attention in previous research. The major contribution of this work is the implementation of several major inference rules with digital circuits. This will hopefully enable the creation of new, largely automated and effective hardware solutions for the decision-making systems.

10. References

[1] M.F. Renzi, P. Vicard, R. Guglielmetti, F. Musella. „Probabilistic expert systems for managing information to improve services." *The TQM Journal*. Emerald Group Publishing Limited. Volume 21, Issue 4, Pages 429 – 442. 2009

[2] Guilan Kong , Dong-Ling Xu , Xinbao Liu and Jian-Bo Yang . „Applying a belief rule-base inference methodology to a guideline-based clinical decision support system." *Expert Systems. The Journal of Knowledge Engineering.* Volume 26 Issue 5, Pages 391 – 408. 2009

[3] Martin Rinard. „Integrated Reasoning and Proof Choice Point Selection in the Jahob System (Mechanisms for Program Survival)". *Lecture Notes In Artificial Intelligence.* Vol. 5663. Springer. 2009

[4] J. McCarthy and P.J. Hayes. „Some philosophical problems from the standpoint of Artificial Intelligence"

Machine Intelligence, Vol. 4, eds. B. Meltzer and D. Michie pp. 463–502. Edinburgh University Press, 1969

[5] R. Kowalski and M. Sergot. „A logic-based calculus of events“. *New Generation Computing* 4 p.67–95. 1986

[6] Vipan Kakkar. „Comparative Study on Analog and Digital Neural Networks“. *IJCSNS International Journal of Computer Science and Network Security*, VOL.9 No.7. 2009.

[7] (a). Matsak E. "Representing logical inference steps with digital circuits". *Lecture Notes in Computer Science: HCI International 2009*. 5618 Springer Berlin, p. 178 – 184. 2009.

[8] (b). Matsak E. "On the logic module in intelligent systems". *Proceedings of the International Conference on Artificial Intelligence. IC – AI'2009*. Volume II p. 594 – 599. Las Vegas, Nevada, USA. 2009.

[9] (c). Matsak E. „Discovering Logical Constructs from Estonian Children Language“. *Thesis on Informatics and System Engineering.* TUT Press. Estonia, Tallinn. 2009

[10] Lorents P. „Knowledge and Logic“. *Proceedings of the International Conference on Artificial Intelligence. IC – AI'2009*. Volume II, p. 568 – 570. Las Vegas, Nevada, USA. 2009

[11] Redgers A., Aleksander I.,. „Digital Neural Networks. Neural network applications in control“. *IEE Control Engineering Series*. 1995

[12] Trong Dung Nguyen, Tu Bao Ho, Hiroshi Shimodaira: A visualization tool for interactive learning of large decision trees. ICTAI 2000: 28-35