In: Multi-Agent Applications for Evolutionary Computation and Biologically Inspired Technologies.
Shu-Heng Chen, Yasushi Kambayashi (Eds), 2010, IGI Global, ISBN: 978-1-60566-898-7, pp. 50-76.

Chapter 4
Agents in Quantum and Neural Uncertainty

Germano Resconi
Catholic University Brescia, Italy

Boris Kovalerchuk
Central Washington University, USA

ABSTRACT

This chapter models quantum and neural uncertainty using a concept of the Agent–based Uncertainty Theory (AUT). The AUT is based on complex fusion of crisp (non-fuzzy) conflicting judgments of agents. It provides a uniform representation and an operational empirical interpretation for several uncertainty theories such as rough set theory, fuzzy sets theory, evidence theory, and probability theory. The AUT models conflicting evaluations that are fused in the same evaluation context. This agent approach gives also a novel definition of the quantum uncertainty and quantum computations for quantum gates that are realized by unitary transformations of the state. In the AUT approach, unitary matrices are interpreted as logic operations in logic computations. We show that by using permutation operators any type of complex classical logic expression can be generated. With the quantum gate, we introduce classical logic into the quantum domain. This chapter connects the intrinsic irrationality of the quantum system and the non-classical quantum logic with the agents. We argue that AUT can help to find meaning for quantum superposition of non-consistent states. Next, this chapter shows that the neural fusion at the synapse can be modeled by the AUT in the same fashion. The neuron is modeled as an operator that transforms classical logic expressions into many-valued logic expressions. The motivation for such neural network is to provide high flexibility and logic adaptation of the brain model.

INTRODUCTION

We model quantum and neural uncertainty using the Agent–based Uncertainty Theory (AUT) that uses complex fusion of crisp conflicting judgments of agents. AUT represents and interprets uniformly several uncertainty theories such as rough set theory, fuzzy sets theory, evidence theory, and probability theory. AUT exploits the fact that agents as independent entities can give

DOI: 10.4018/978-1-60566-898-7.ch004

conflicting evaluations of the same attribute. It models conflicting evaluations that are fused in the same evaluation context. If only one evaluation is allowed for each statement in each context (world) as in the modal logic then there is no logical uncertainty. The situation that the AUT models is inconsistent (fuzzy) and is very far from the situation that modeled by the traditional logic that assumes consistency. We argue that the AUT by incorporating such inconsistent statements is able to model different types of conflicts and their fusion known in many-valued logics, fuzzy logic, probability theory and other theories.

This chapter shows how the agent approach can be used to give a novel definition of the quantum uncertainty and quantum computations for quantum gates that are realized by unitary transformations of the state. In the AUT approach, unitary matrices are interpreted as logic operations in logic computations. It is shown, that by using permutation operators that are unitary matrixes any type of complex classical logic expression can be generated. The classical logic has well-known difficulties in quantum mechanics. Now with the quantum gate we introduce classical logic into the quantum domain. We connect the intrinsic irrationality of the quantum system and the non-classical quantum logic with the agents. We argue that Agent-based uncertainty theory (AUT) can help to find meaning for quantum superposition of non-consistent states for which one particle can be at the different points in the same time or the same particle can have spin up and down in the same time.

Next, this chapter shows that the neural fusion at the synapse can be modeled by the AUT. Agents in the neural network are represented by logic input values in the neuron itself. In the ordinary neural networks any neuron is a processor that models a Boolean function. We change the point of view and consider a neuron as an operator that transforms classical logic expressions into many-valued logic expressions or in other words, changes crisp sets into fuzzy sets. This neural network consists of neurons at two layers. At the first one, neurons or agents implement the classical logic operations. At the second layer neurons or nagents (neuron agents) compute the same logic expression with different results. These are many-valued neurons that fuse results provided by different agents at the first layer. They fuse conflicting or inconsistent situations. The network is based on use of the logic of the uncertainty instead of the classical logic. The motivation for such neural network is to provide high flexibility and logic adaptation of the brain model. In this brain model, communication among agents is specified by the fusion process in the neural elaboration.

The probability calculus does not incorporate explicitly the concepts of irrationality or logic conflict of agent's state. It misses structural information at the level of individual objects, but preserves global information at the level of a set of objects. Given a dice the probability theory studies frequencies of the different faces E={e} as independent (elementary) events. This set of elementary events E has *no structure*. It is only required that elements of E are *mutually exclusive* and *complete*, that is no other alternative is possible. The order of its elements is irrelevant to probabilities of each element of E. No irrationality or conflict is allowed in this definition relative to mutual exclusion. The classical probability calculus does not provide a mechanism for modelling uncertainty when agents communicate (collaborates or conflict). Recent work by Halpern (2005) is an important attempt to fill this gap.

This chapter is organized as follows: Sections 2 and 3 provide a summary of the AUT starting from concepts and definitions. Section 4 presents links between quantum mechanics and first order conflicts in the AUT. Section 5 discusses the neural images of the AUT. Section 6 concludes this chapter.

CONCEPTS AND DEFINITIONS

Now we will provide more formal definition of AUT concepts. It is done first for individual agents then for sets of agents. Consider a set of agents $G=\{g_1, g_2, \ldots, g_n\}$. Each agent g_k assigns binary true/false value $v \in \{True, false\}$ to proposition p. To show that v was assigned by the agent g_k we use notation $g_k(p) = v_k$.

Definition. A triple $g = \langle N, A_R, A_A \rangle$ is called an *agent* g if N is label that is interpreted as agent's name and A_R is as set of truth-evaluation actions and A_A is a set of non- truth-evaluation actions associated with name N. For instance agent g called Professor has a set of truth-evaluation actions A_R such as grading students' answer, while delivering a lecture is in another category A_A.

Definition. An agent g is called a *reasoning agent* if g assigns a truth-value v(p) to any proposition p from a set of propositions S and any logical formula based on S.

The actions of a general agent may or may not include truth-evaluation actions. From a mathematical viewpoint a reasoning agent g serves as a mapping,

$$g: p \rightarrow v(p)$$

While natural agents have both AR, and AA, artificial software agents and robots may have only one of them.

Definition. A set of reasoning agents G is called *totally consistent* for proposition p if any agent g from {g} always provides the same truth value for p.

In other words, all agents in G are in concord or logical coherence for the same proposition. Thus, changing the agent does not change logic value v(p) for a totally consistent set of agents. Here v(p) is *global* for G and has *no local variability* that is independent on the individual agent's evaluation. The classical logic is applicable for such set of consistent (rational) agents.

Definition. A set of reasoning agents G is called *inconsistent for* proposition p if there are two subset of agents G_1, G_2 such that agents from them provides different truth values for p.

Multiple reasons lead to agents' inconsistency. The most general one is the context (or hidden variables) in which each agent is evaluating statement p. Even in a relatively well formalized environment, context is not fully defined. Agent g_1 may evaluate p in context "abc", but agent g_2 in context "ab". Agent (robot) g_2 may have no sensor to obtain signal c or abilities to process and reason correctly with input from that sensor (e.g., color blind agent with a panchromatic camera).

Definition. Let S be a set propositions, $S=\{p_1, p_2, \ldots, p_n\}$ then set $\neg S = \{\neg p_1, \neg p_2, \ldots, \neg p_n\}$ is called a *complementary set* of S.

Definition. A set of reasoning agents G is called *S-only-consistent* if agents {g} are consistent only for propositions in $S=\{p_1, p_2, \ldots, p_n\}$ and are inconsistent in the complimentary set $\neg S$.

The evaluations of p is a vector-function $v(p)=(v_1(p), v_2(p)\ldots, v_n(p))$ for a set of agents G that we will represent as follows:

$$\mu(p \vee q) = \frac{w_1(v_1(p) \vee v_1(q)) + .. + w_N(v_N(p) \vee v_N(q))}{N} \tag{1}$$

An example of the logic evaluation by five agent is shown below for p = "$A>B$"

$$\mathbf{v}(p) = \begin{pmatrix} g_1 & g_2 & \cdots & g_{n-1} & g_n \\ v_1 & v_2 & \ldots & v_{n-1} & v_n \end{pmatrix}$$

Here A > B is true for agents g_1, g_3, and g_5 and it is false for the agents g_2, and g_4.

Kolmogorov's axioms of the probability theory are based on a totally consistent set of agents (for a set of statements) on mutual exclusion of elementary events. It follows from the definitions below for a set of events $E=\{e_1.e_2,\ldots,e_n\}$.

Definition. Set $E=\{e_1.e_2,\ldots,e_n\}$ is called a set of *elementary events* (or mutually exclusive events) for predicate El if

$\forall\, e_i\, e_j \in A\ El(e_i) \vee El(e_j)$ =True and $El(e_i) \wedge El(e_j)$=False.

In other words, event e_i is an *elementary event* ($El(e_i)$=true) if for any j, $j \neq i$ events e_i and e_j cannot happen simultaneously, that is probability $P(e_i \wedge e_j) = 0$ and $P(e_i \vee e_j) = P(e_i) + P(e_j)$. Property $P(e_i \wedge e_j)=0$ is the *mutual exclusion axiom* (ME- axiom).

Let $S=\{p_1,p_2,.., p_n\}$ be a set of statements, where $p_i=p(e_i)$=True if and only if event e_i is an elementary event. In probability theory, $p(e_i)$ is not associated with **any** specific agent. It is assumed to be a global property (applicable to all agents). In other words, statements $p(e_i)$ are totally consistent for all agents.

Definition. A set of agent $S = \{g_1.g_2,\ldots,g_n\}$ is called a *ME-rational* set of agents if

$\forall\, e_i\, e_j \in A,\ \forall\, g_i \in S,\ p(e_i) \vee p(e_j)$ =True and $p(e_i) \wedge p(e_j)$=False.

In other words, these agents are *totally consistent* or *rational on Mutual Exclusion.* Previously we assumed that each agent assigns value v(p) and we did not model this process explicitly. Now we introduce a set of criteria $C=\{C_1,C_2,\ldots,C_m\}$ by which an agent can decide if a proposition p is true or false, i.e., now $v(p) = v(p,C_i)$, which means that p is evaluated by using the criterion C_i.

Definition Given a set of criteria C, agent g is in a *self-conflicting state* if

$\exists\, C_i, C_j\ (C_i, C_j \in C)\ \&\ v(p,C_i) \neq v(p,C_j)$

In other words, an agent is in a self–conflicting state if two criteria exist such that p is true for one of them and false for another one. With the explicit set of criteria, the logic evaluation function v(p) is not vector-function any more, but it is expanded to be a matrix function as shown below:

$$f(p) = \begin{pmatrix} g_1 & g_2 & g_3 & g_4 & g_5 \\ true & false & true & false & true \end{pmatrix} \tag{2}$$

For example, four agents using four criteria can produce v(p) as follows:

$$\mathbf{v}(p) = \begin{bmatrix} & g_1 & g_2 & \cdots & g_n \\ C_1 & v_{1,1} & v_{1,2} & \cdots & v_{1,n} \\ C_2 & v_{2,1} & v_{2,2} & \cdots & v_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ C_m & v_{m,1} & v_{m,2} & \cdots & v_{m,n} \end{bmatrix}$$

Note that introduction of a set of criteria C explains self-conflict, but does not remove it. The agent still needs to resolve the ultimate preference contradiction having a goal, say, to buy only one and better car. It also does not resolve conflict among different agents if they need to buy a car jointly.

If agent g can modify criteria in C making them consistent for p then g can resolves self-conflict. The agent can be in a logic conflict state because of inability to understand the complex *context* and to evaluate *criteria*. For example, in the stock market environment, some traders quite often do not understand a complex context and rational criteria of trading. These traders can be in logic conflict exhibiting chaotic, random, and impulsive behavior. They can sell and buy stocks, exhibiting logic conflicting states "sell" = p and "buy" = ¬p in the same market situation that appears as irrational behavior, which means that the statement $p \wedge \neg p$ can be true.

A logical structure of self-conflicting states and agents is much more complex than it is without self-conflict. For m binary criteria $C_1, C_2,\ldots C_m$ that evaluate the logic value for the same attribute, there are 2^m possible states and only two of them (all true or all false values) do not exhibit conflict between criteria.

FRAMEWORK OF FIRST ORDER OF CONFLICT LOGIC STATE

Definition. A set of agents G is in a first order of conflicting logic state (*first order conflict*, for short) if

$\exists\, g_i, g_j\ (g_i, g_j \in G)\ \&\ v(p,g_i) \neq v(p,g_j)$.

In other words, there are agents g_i and g_j in G for which exist different values v_i, v_j in

$$\mathbf{v}(p) = \begin{bmatrix} & g_1 & g_2 & g_3 & g_4 \\ C_1 & true & false & false & true \\ C_2 & false & false & true & true \\ C_3 & true & false & true & true \\ C_4 & false & false & true & true \end{bmatrix}$$

A set of agents G is in the *first order of conflict* if

$G(A>B) \cap G(A<B)= \emptyset$, $G(A>B) \neq \emptyset$, $G(A>B) \cup G(A<B)= G$.

The following definition presents this idea in general terms.

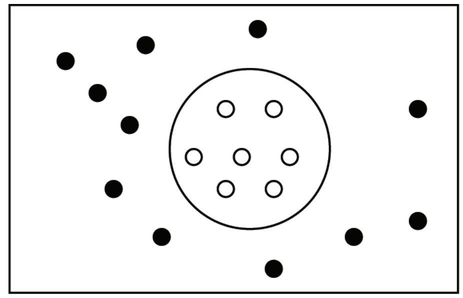
Definition. A set of agents G is in a **First Order Conflict (FOC)** for proposition p if

$G(p) \cap G(\neg p)= \emptyset$, and $G(p) \neq \emptyset$, $G(p) \cup G(\neg p)= G$.

Figure 1 shows a set of 20 agents in the logic conflicting state, where 7 white agents are in the state True and 13 black agents are in the state False for the same proposition p = “A > B”.

Below we show that at the first order of conflicts, AND and OR operations should differ from the classical logic operations and should be **vector operations** in the space of the agents' evaluations (*agents space*). The vector operations reflect a structure of logic conflict among coherent individual agent evaluations.

Figure 1. A set of 20 agents in the first order of logic conflict



Fusion Process

If a single decision must be made at the first order of conflict, then we must introduce a **fusion process** of the logic values of proposition p given by all agents. A basic way to do this is to compute the weighted frequency of logic value given by all agents:

$$\mathbf{v}(p) = \begin{pmatrix} g_1 & g_2 & \dots & g_{n-1} & g_n \\ v_1 & v_2 & \dots & v_{n-1} & v_n \end{pmatrix} \tag{3}$$

where

$$\mu(p) = w_1 v_1(p) + \dots + w_n v_n(p) = \begin{bmatrix} v_1(p) \\ v_2(p) \\ \dots \\ v_n(p) \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{bmatrix}$$

are two vectors in the space of the agents' evaluations. The first vector contains all logic states (True/False) for all agents, the second vector (with property $\begin{bmatrix} v_1(p) \\ v_2(p) \\ \dots \\ v_n(p) \end{bmatrix}$ and $\begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_n \end{bmatrix}$) contains non-negative weights (utilities) that are given to each agent in the fusion process. In a simple frequency case, each weight is equal to 1/n. At first glance, μ(p) is the same as used in the probability and utility theories. However, classical axioms of the probability theory have no references to agents producing initial uncertainty values and do not violate the mutual exclusion. Below we define **vector logic operations** for the first order of conflict logic states **v**(p).

Definition

$$\sum_{k=1}^{n} w_k = 1$$

$$\mathbf{v}(p \wedge q) = v_1(p) \wedge v_1(q), \ldots, v_n(p) \wedge v_n(q),$$

$$\mathbf{v}(p \vee q) = v_1(p) \vee v_1(q), \ldots, v_n(p) \vee v_n(q)$$

where the symbols ∧, ∨, ¬ in the right side of the equations are the classical AND, OR, and NOT operations.

Below these operations are written with explicit indication of agents (in the first row):

$$\mathbf{v}(\neg p) = \neg v_1(p), \ldots, \neg v_n(p),$$

Below we present the important properties of sets of conflicting agents at the first order of conflicts. Let |G(x)| be the numbers of agents for which proposition x is true.

Statement 1 sets up properties of the AND and OR operations for nested sets of conflicting agents.

Statement 1 (general non min/max properties of ∧ and ∨ operations)

If G is a set of agents at the first order of conflicts and

$$|G(q)| \leq |G(p)|$$

then ∧ and ∨ logic operations satisfy the following properties

$$|G(p \wedge q)| = \min(|G(p)|, |G(q)|) - |G(\neg p \wedge q)|, \tag{4}$$

$$|G(p \vee q)| = \max(|G(p)|, |G(q)|) + |G(\neg p \wedge q)|. \tag{5}$$

If G is a set of agents at the first order of conflicts and

$$|G(p)| \leq |G(q)|$$

then ∧ and ∨ logic operations satisfy the following properties

$$|G(p \wedge q)| = \min(|G(p)|, |G(q)|) - |G(\neg q \wedge p)|,$$

$$|G(p \vee q)| = \max(|G(p)|, |G(q)|) + |G(\neg q \wedge p)|.$$

and also

$$|G(p \vee q)| = |G(p) \cup G(q)|, \; G(p \wedge q)| = |G(p) \cap G(q)|$$

Corollary 1 (min/max properties of ∧ and ∨ operations for nested sets of agents)

If G is a set of agents at the first order of conflicts such that G(q) ⊂ G(p) or G(p) ⊂ G(q) then

$$G(\neg p \wedge q)| = \emptyset \text{ or } G(\neg q \wedge p)| = \emptyset$$

$$|G(p \wedge q)| = \min(|G(p)|, |G(q)|)$$

$$|G(p \vee q)| = \max(|G(p)|, |G(q)|)$$

This follows from the statement 1. The corollary presents a well-known condition when the use of min, max operations has the clear justification.

Let $G^c(p)$ is a *complement* of G(p) in G: $G^c(p) = G \setminus G(p)$, $G = G(p) \cup G^c(p)$.

Statement 2. $G = G(p) \cup G^c(p) = G(p) \cup G(\neg p)$.

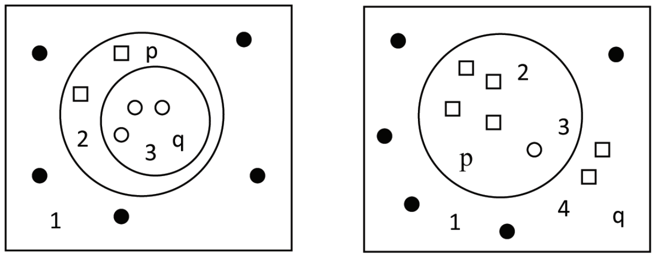
Corollary 2. $G(\neg p) = G^c(p)$

It follows directly from Statement 2.

Statement 3. If G is a set of agents at the first order of conflicts then

$$G(p \vee \neg p) = G(p) \cup G(\neg p) = G(p) \cup G^c(p) = G$$

Figure 2. A set of total 10 agents with two different splits to G(p) and G(q) subsets (a) and (b)



$G(p \wedge \neg p) = G(p) \cap G(\neg p) = G(p) \cap G^c(p) = \emptyset$

It follows from the definition of the first order of conflict and statement 2. In other words, $G(p \wedge \neg p) = \emptyset$ corresponds to the contradiction $p \wedge \neg p$, that is always false and $G(p \vee \neg p)= G$ corresponds to the tautology $p \vee \neg p$, that is always true in the first order conflict.

Let $G_1 \oplus G_2$ be a *symmetric difference* of sets of agents G_1 and G_2,

$G_1 \oplus G_2 = (G_1 \cap G_2^c) \cup (G_1^c \cap G_2)$and let $p \oplus q$ be the *exclusive or* of propositions p and q,

$p \oplus q = (p \wedge \neg q) \vee (\neg p \wedge q)$.

Consider, a set of agents $G(p \oplus q)$. It consists of agents for which values of p and q differ from each other, that is

$G(p \oplus q)= G((p \wedge \neg q) \vee (\neg p \wedge q))$.

Below we use the number of agents in set $G(p \oplus q)$ to define a measure of difference between statements p and q and a measure of difference between sets of agents G(p) a G(q).

<u>Definition.</u> A measure of difference D(p,q) between statements p and q and a measure of difference D(G(p),G(q)) between sets of agents G(p) a G(q) are defined as follows:

$D(p,q) = D(G(p),G(q))= |G(p) \oplus G(q)|$

<u>Statement 4</u>. D(p,q) = D(G(p),G(q)) is a distance, i.e., it satisfies distance axioms

$D(p,q) \geq 0$

$D(p,q) = D(q,p)$

$D(p,q)+D(q,h) \geq D(p,h)$.

This follows from the properties of the symmetric difference $\oplus$ (e.g.., Flament 1963).

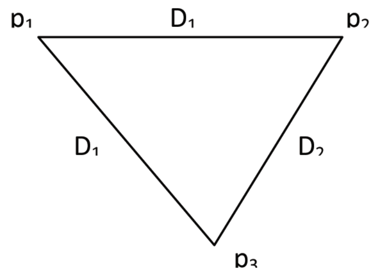
Figure 2 illustrates a set of agents G(p) for which p is true and a set of agents G(q) for which q is true. In Figure 2(a) the number of agents for which truth values of p and q are different, $(\neg p \wedge q) \vee (p \wedge \neg q)$, is equal to 2. These agents are represented by white squares. Therefore the distance between G(p) and G(q) is 2. Figure 2(b) shows other G(p) and G(q) sets with the number of the agents for which $\neg p \wedge q) \vee (p \wedge \neg q$ is true equal to 6 (agents shown as white squares and squares with the grid). Thus, the distance between the two sets is 6.

In Figure 2(a), set 2 consists of 2 agents $|G((p \wedge \neg q)| = 2$ and set 4 is empty, $|G((\neg p \wedge \neg q)| = 0$, thus D(Set2, Set4)=2. This emptiness means that a set of agents with true p includes a set of agents with true q.

In Figure 2(b), set 2 consists of 4 agents $|G((p \wedge \neg q)| = 4$ and set 4 consists of 2 agents, $|G((\neg p \wedge \neg q)| = 2$, thus D(Set2,Set4)=6.

These splits produce different distances between G(p) and G(q). The distance in the case (a) is equal to 2; the distance in the case (b) is equal to 6. Set 1 (black circles) consists of agents for which both p and q are false, Set 2 (white squares) consists of agents for which p is true but q is false. Set 3 (white circles) consists of agents for which p and q are true, and Set 4 (squares with grids) consists of agents for which p is false and q is true,

Figure 3. Graph of the distances for three sentences p_1, p_2, and p_3



For complex computations of logic values provided by the agents we can use a *graph of the distances* among the sentences p_1, p_2, ….,p_N. For example, for three sentences p_1, p_2, and p_3, we have a graph of the distances shown in Figure 3.

This graph has can be represented by a distance matrix

$$\mathbf{v}(p \wedge q) = \begin{pmatrix} g_1 & g_2 & \dots & g_{n-1} & g_n \\ v_1(p) \wedge v_1(q) & v_2(p) \wedge v_2(q) & \dots & v_{n-1}(p) \wedge v_{n-1}(q) & v_n(p) \wedge v_n(q) \end{pmatrix}$$

$$\mathbf{v}(p \vee q) = \begin{pmatrix} g_1 & g_2 & \dots & g_{n-1} & g_n \\ v_1(p) \vee v_1(q) & v_2(p) \vee v_2(q) & \dots & v_{n-1}(p) \vee v_{n-1}(q) & v_n(p) \vee v_n(q) \end{pmatrix}$$

$$\mathbf{v}(\neg p) = \begin{pmatrix} g_1 & g_2 & \dots & g_{n-1} & g_n \\ v_1(\neg p) & v_2(\neg p) & \dots & v_{n-1}(\neg p) & v_n(\neg p) \end{pmatrix}$$

which has a general form of a symmetric matrix,

$$D = \begin{bmatrix} 0 & D_{1,2} & D_{1,3} \\ D_{1,2} & 0 & D_{2,3} \\ D_{1,3} & D_{2,3} & 0 \end{bmatrix}$$

Having distances D_{ij} between propositions we can use them to compute complex expressions in agents' logic operation. For instance, using $0 \leq D(p, q) \leq G(p) + G(q)$ and

$$\begin{aligned} G(p \wedge q) &\equiv G(p \vee q) - G((\neg p \wedge q) \vee (p \wedge \neg q)) \\ &\equiv G(p \vee q) - D(p, q) \end{aligned}$$

Evidence Theory and Rough Set Theory with Agents

In the evidence theory any subset $A \subseteq U$ of the universe U is associated with a value m(A) called a basic assignment probability such that

$$D = \begin{bmatrix} 0 & D_{1,2} & \dots & D_{1,\mathrm{N}} \\ D_{1,2} & 0 & \dots & D_{2,\mathrm{N}} \\ \dots & \dots & \dots & .. \\ D_{1,\mathrm{N}} & D_{2,\mathrm{N}} & \dots & 0 \end{bmatrix}$$

Respectively, the belief Bel(Ω) and plausibility Pl(Ω) measures for any set Ω are defined as

$$\sum_{k=1}^{2^N} m(A_k) = 1$$

Thus, Bel measure includes all subsets that inside Ω, The Pl measure includes all sets with non-empty intersection with Ω. In the evidence theory as in the probability theory we associate one and only one agent with an element.

Figure 4 shows set A at the border of the set Ω, which is divided in two parts: one inside Ω and another one outside it. For the belief measure, we exclude the set A, thus we exclude the false state (*f,f*) and a logical self-conflicting state (*t,f*). But for the plausibility measure we accept the (*t, t*) and the self-conflicting state (*f,t*). In the belief measure, we exclude any possible self-conflicting state, but in the plausibility measure we accept self-conflicting states.

There are two different criteria to compute the belief and plausibility measures in the evidence theory. For belief, C_1 criterion is related to set Ω, thus C_1 is true for the cases inside Ω. The second criterion C_2 is related to set A, thus C_2 is false for the cases inside A. Now we can put in evidence a logically self-conflicting state (*t, f*), where we eliminate A also if it is inside Ω . The same is applied to the plausibility, where C_2 is true for cases inside A. In this situation, we accept a

Figure 4. Example of irrational agents for the belief theory

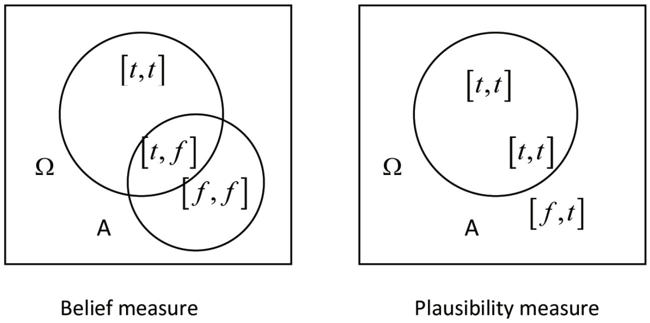
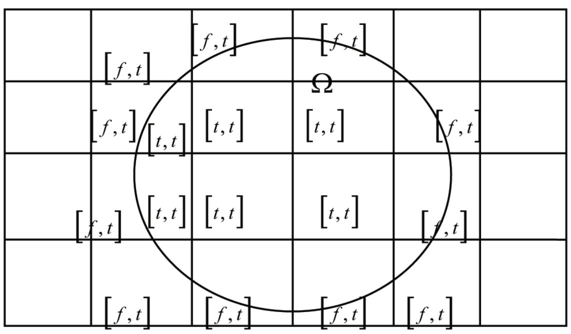


Figure 5. Example of irrational agents for the Rough sets theory



logically self-conflicting situation (*f, t*), where cases in Ω are false but cases in A are true. We also use the logical self-conflict to study the roughs set as shown in Figure 5.

Figure 5 shows set Ω that includes the logically self-conflicting states (f, t).

The internal and external part of Ω is defined by a non empty frontier, where the couples [t, t]are present and as well as the self-conflicting couples [f, t].

QUANTUM MECHANICS: SUPERPOSITION AND FIRST ORDER CONFLICTS

The Mutual Exclusive (ME) principle states that an object cannot be in two different locations at the same time. We call this locality phenomenon. In quantum mechanics, we have *non-locality phenomenon* for which the same particle can be in many different positions at the same time. This is a clear violation of the ME principle for location. The non-locality is essential for quantum phenomena. Given proposition p = "The particle is in position x in the space", agent g_i can say that p is true, but agent g_j can say that the same p is false.

Individual states of the particle include its position, momentum and others. The *complete state* of the particle is a *superposition* of the quantum states of the particle w_i at different positions x_i in the space, This superposition can be a global wave function:

$$Bel(\Omega) = \sum_{A_k \subseteq \Omega} m(A_k), \;\; Pl(\Omega) = \sum_{A_k \cap \Omega \neq \varnothing} m(A_k)$$

where $\psi = w_1 \left| x_1 \right\rangle + w_2 \left| x_2 \right\rangle + ... + w_n \left| x_n \right\rangle$ denotes the state at the position x_i (D'Espagnat, 1999). If we limit our consideration by an individual quantum state of the particle and ignore other states then we are in Mutual Exclusion situation of the classical logic (the particle is at x_i or not). In the same way in AUT, when we observe only one agent at the time the situation collapses to the classical logic that may not adequately represent many-valued logic properties. Having multiple states associated with the particle, we use AUT multi-valued logic as a way to model the multiplicity of states. This situation is very complex because measuring position x_i of the particle changes the superposition value ψ. However, this situation can be modelled by the first order conflict, because each individual agent has no self-conflict.

Quantum Computer

The modern concept of the quantum computer is based on following statements (Abbott, Doering, Caves, Lidar, Brandt, Hamilton, Ferry, Gea-Banacloche, Bezrukov, & Kish, 2003; DiVincenzo, 2000; DiVincenzo, 1995; Feynman, 1982; Jaeger, 2006; Nielsen & Chuang, 2000; Benenti, 2004; Stolze & Suter, 2004; Vandersypen, Yannoni, & Chuang, 2000; Hiroshi & Masahito, 2006):

1. Any state denoted as $\left|x_i\right\rangle$ is a field of complex numbers on the reference space (position and time). In our notation, the quantum state is a column of values for different points (objects).
2. Any combination of states $\left|n\right\rangle$ is a product of fields.

Given two atoms with independent states of energy $\left|nm\right\rangle = \left|n\right\rangle\left|m\right\rangle$ the two atoms have the state $\left|n\right\rangle$ and $\left|m\right\rangle$ that is the product of the separate state for atom 1 and atom 2.

3. The space H is the space which dimension is the number of elementary fields or states.
4. In quantum mechanics, we compute the probability that the qubit takes values 1 or 0 in the all space time.

A qubit has some similarities to a classical bit, but is overall very different. Like a bit, a qubit can have two possible values—normally a 0 or a 1. The difference is that whereas a bit *must* be either 0 or 1, a qubit can be 0, 1, or a superposition of both.

Explanation:

The classical digital computer operates with bits that is with 1 and 0. The quantum computer operates with two states $\left|nm\right\rangle = \left|n\right\rangle\left|m\right\rangle$ In quantum mechanics, the state $\left|1\right\rangle, \left|0\right\rangle$ is associated with a field of probability that the bit assumes the value 1 in the space time. Thus, in the classical sense the bit is 1 in a given place and time. In quantum mechanics, we have the distribution of the probability that a bit assumes the value 1 in the space time. Therefore we have a field of uncertainty for value 1 as well as a field of uncertainty for value 0. This leads to the concept of the qubit.

5. Any space of 2^m - 1 dimensions is represented by H and UH is the unitary transformation of H by which computations are made in quantum physics.

Explanation:

H is a matrix of qubits that can assume different values. For simplicity we use only the numbers 1 and 0, where 1 means qubit that assume the value 1 in the all space time and the same for 0.

In quantum mechanics, the qubit value can be changed only with particular transformations U or unitary transformations. Only the unitary transformations can be applied. Given the unitary transformation

$$\left|1\right\rangle$$

such that $U^T U = I$ and

$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

with $2^2 = 4$ qubits we have

$$H = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{1}$$

Also

$$\mathrm{UH} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$$

is the XOR that can be written as $c = a \oplus b$.

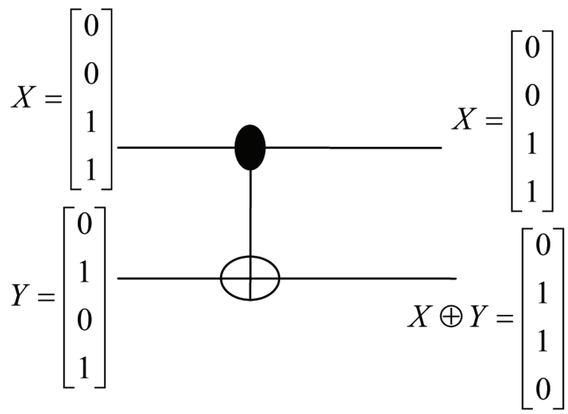
In quantum mechanics, (1) is represented by the Deutch circuit shown in Figure 6

Now we want to clarify and extend the Deutch's interpretation of the quantum computer to introduce not only a Boolean algebra in the quantum computer, but also a many-valued logic. This is more in line with the quantum mechanics intrinsic inconsistency due to the superposition of mutually exclusive states. In quantum mechanics a particle can have two opposite spins at the same time and different positions again at the same time. Here we cannot apply the mutual exclusive states of the classical physics where we have logical consistency that is any particle has one and only one position.

Permutations, Unitary Transformation and Boolean Algebra

In quantum mechanics any physical transformation is governed by special transformations of the Hilbert space of the states. For the unitary transformation U the probability of the quantum system is invariant. The unitary transformation has property U $U^T = 1$ and is extensively used in the quantum computer. Now we establish a

Figure 6. Quantum Computer circuit that represents the transformation



connection between unitary transformation and permutation. In fact we have

$$\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

where

$$\mathrm{Q} = \mathrm{UH} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$$

and

$$U = \begin{pmatrix} P_1 & P_2 & P_3 & P_4 \\ P_1 & P_2 & P_4 & P_3 \end{pmatrix}$$

Now for

$$(\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \mathrm{P}_4) = ((0,0), (0,1), (1,0), (1,1))$$

and

$$Q_1(X,Y)=\begin{bmatrix}0\\0\\1\\1\end{bmatrix}=X\ ,Q_2(X,Y\ )=\begin{bmatrix}0\\1\\1\\0\end{bmatrix}$$

Similarly, for the negation we have

$$Q_2(X,1\ )=\begin{bmatrix}1\\0\end{bmatrix}=\neg X$$

$$\neg X=\begin{pmatrix}P_1 & P_2 & P_3 & P_4\\ P_1 & P_2 & P_4 & P_3\end{pmatrix}=(3,4)$$

and

$$\mathrm{Q}=\mathrm{UH}=\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}0&0\\0&1\\1&0\\1&1\end{bmatrix}=\begin{bmatrix}0&1\\0&0\\1&0\\1&1\end{bmatrix}$$

and

$$Q_2(X,1\ )=\begin{bmatrix}0\\1\end{bmatrix}=X$$

$$X=\begin{pmatrix}P_1 & P_2 & P_3 & P_4\\ P_2 & P_1 & P_3 & P_4\end{pmatrix}=(1,2)$$

for

$$\neg X=\left\{(1,2)\right\}^C=\left\{(3,4)\right\}$$

Therefore,

$$\mathrm{Q}=\mathrm{UH}=\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\begin{bmatrix}0&0\\0&1\\1&0\\1&1\end{bmatrix}=\begin{bmatrix}0&1\\0&0\\1&1\\1&0\end{bmatrix}$$

$$Q_2(X,1\ )=\begin{bmatrix}0\\0\end{bmatrix}=\ false$$

$$\neg X\wedge X=\begin{pmatrix}P_1 & P_2 & P_3 & P_4\\ P_2 & P_1 & P_4 & P_3\end{pmatrix}=(1,2)(3,4)$$

and

$$false=\neg X\wedge X=\left\{(1,2)\right\}\cup\left\{(3,4)\right\}=\left\{(1,2),(3,4)\right\}=Universe$$

Next we have

$$\mathrm{Q}=\mathrm{UH}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}0&0\\0&1\\1&0\\1&1\end{bmatrix}=\begin{bmatrix}0&0\\0&1\\1&0\\1&1\end{bmatrix}$$

$$Q_2(X,1\ )=\begin{bmatrix}1\\1\end{bmatrix}=true$$

and

$$\neg X\vee X=\begin{pmatrix}P_1 & P_2 & P_3 & P_4\\ P_1 & P_2 & P_3 & P_4\end{pmatrix}=\varnothing$$

The quantum computer circuit can be explained as follows. Having

c = X ⊕ Y = (X ∧ ¬ Y) ∨ (¬ X ∧ Y) we infer that

Y = 0 ⇒ X ⊕ Y = X

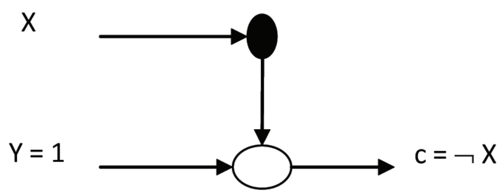
Y = 1 ⇒ X ⊕ Y = ¬X

Thus, we have what is shown in Figure 7

Now we use a common symbolic representation of elementary states $false=\neg X\vee X=\left\{(1,2)\right\}\cap\left\{(3,4)\right\}=\left\{\ \right\}=\varnothing$ $\left|0\right\rangle$ as unitary vectors in 2-D Hilbert space

$\left|1\right\rangle$

Figure 7. Quantum Computer circuit by NOT operation or "¬ "



Any vector in this 2-D space is a superposition $\left|1\right\rangle = \begin{bmatrix}1\\0\end{bmatrix}$, $\left|0\right\rangle = \begin{bmatrix}0\\1\end{bmatrix}$ that is symbolically represented as

$$\psi = \alpha\left|1\right\rangle + \beta\left|0\right\rangle$$

Thus,

$$\psi = \begin{bmatrix}\alpha\\ \beta\end{bmatrix}$$

Points in 2-D Hilbert space for the four states are

$$\psi = \left|1\right\rangle = \begin{bmatrix}1\\0\end{bmatrix} \quad , \psi = \left|0\right\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

and

$$H = \begin{bmatrix}\left|0\right\rangle\left|0\right\rangle\\ \left|0\right\rangle\left|1\right\rangle\\ \left|1\right\rangle\left|0\right\rangle\\ \left|1\right\rangle\left|1\right\rangle\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 1\\ 1 & 0\\ 1 & 1\end{bmatrix}$$

In the classical Boolean algebra, complex expressions are obtained by composition using the elementary operations (NOT, AND, and OR). In quantum computer, we cannot use the same elementary operations and must change our point of view. The quantum mechanics is based on the idea of states, superposition of states, product of states (entanglement or relationship among states) and the Hilbert space of the states with unitary transformation. Thus, the mathematical formalism is closer to the vector space than to the Boolean algebra. We remark that permutations can be taken as generators of mathematical groups. This sets up an interesting link between *groups of symmetry* and *Boolean algebra*, which can lead to a deeper connection with the Boolean logic. In fact, given the permutation with a set of points P in 2-D space, every permutation of the states can generate only the following set of Boolean functions

$$\left|11\right\rangle = \left|1\right\rangle\left|1\right\rangle = \begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\begin{bmatrix}1\\0\end{bmatrix}\\ 0\begin{bmatrix}1\\0\end{bmatrix}\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix} \quad , \quad \left|10\right\rangle = \left|1\right\rangle\left|0\right\rangle = \begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}1\begin{bmatrix}0\\1\end{bmatrix}\\ 0\begin{bmatrix}0\\1\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix} ,$$

$$\left|01\right\rangle = \left|0\right\rangle\left|1\right\rangle = \begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\begin{bmatrix}1\\0\end{bmatrix}\\ 1\begin{bmatrix}1\\0\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix} \quad , \quad \left|00\right\rangle = \left|0\right\rangle\left|0\right\rangle = \begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\begin{bmatrix}0\\1\end{bmatrix}\\ 1\begin{bmatrix}0\\1\end{bmatrix}\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

To obtain the Boolean function NAND we must extend the space of the objects and introduce a 3-D attribute space. Thus, we have the Toffoli transformation and the permutation

$$F = \left\{ X, \neg X \text{ , } \neg X \wedge X \text{ , } \neg X \vee X \text{) } \right\}$$

For

$$\mathrm{U} = \begin{pmatrix}\mathrm{P}_1 & \mathrm{P}_2 & \mathrm{P}_3 & \mathrm{P}_4 & \mathrm{P}_5 & \mathrm{P}_6 & \mathrm{P}_7 & \mathrm{P}_8\\ \mathrm{P}_1 & \mathrm{P}_2 & \mathrm{P}_3 & \mathrm{P}_4 & \mathrm{P}_5 & \mathrm{P}_6 & \mathrm{P}_8 & \mathrm{P}_7\end{pmatrix} = (7,8)$$

and

$$\mathrm{U} = \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0\end{bmatrix}$$

and

$$X=\begin{bmatrix}0\\0\\0\\0\\1\\1\\1\\1\end{bmatrix},Y=\begin{bmatrix}0\\0\\1\\1\\0\\0\\1\\1\end{bmatrix},Z=\begin{bmatrix}0\\1\\0\\1\\0\\1\\0\\1\end{bmatrix},H=\begin{bmatrix}0&0&0\\0&0&1\\0&1&0\\0&1&1\\1&0&0\\1&0&1\\1&1&0\\1&1&1\end{bmatrix}=\begin{bmatrix}X&Y&Z\end{bmatrix}$$

Now we put

$$Q\,(X,Y,Z)=UH=\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{bmatrix}\begin{bmatrix}0&0&0\\0&0&1\\0&1&0\\0&1&1\\1&0&0\\1&0&1\\1&1&0\\1&1&1\end{bmatrix}=\begin{bmatrix}0&0&0\\0&0&1\\0&1&0\\0&1&1\\1&0&0\\1&0&1\\1&1&1\\1&1&0\end{bmatrix}$$

Thus,

$$Q_1(X,Y,Z)=\begin{bmatrix}0\\0\\0\\0\\1\\1\\1\\1\end{bmatrix}=X\ ,\ Q_2(X,Y,Z)=\begin{bmatrix}0\\0\\1\\1\\0\\0\\1\\1\end{bmatrix}=Y\ ,\ Q_3(X,Y,Z)=\begin{bmatrix}0\\1\\0\\1\\0\\1\\1\\0\end{bmatrix}$$

Therefore,

$$Q_3\begin{bmatrix}X=0&Y=0&Z=1\\X=1&Y=0&Z=1\\X=0&Y=1&Z=1\\X=1&Y=1&Z=1\end{bmatrix}=\begin{bmatrix}1\\1\\1\\0\end{bmatrix}=\neg(X\wedge Y)=X\,NAND\;Y$$

The permutation of (7,8) introduces a zero, this Q_3 (1, 1, 1) = 0 .

For the permutation we have

$$Q_3(X,Y,1)=\neg(X\wedge Y)=XNAND\;Y$$

and

$$Y=\begin{pmatrix}P_1&P_2&P_3&P_4&P_5&P_6&P_7&P_8\\P_2&P_1&P_4&P_3&P_5&P_6&P_7&P_8\end{pmatrix}=(1,2)(3,4)$$

or

$$UP=\begin{bmatrix}0&1&0&0&0&0&0&0\\1&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\end{bmatrix}\begin{bmatrix}0&0&0\\0&0&1\\0&1&0\\0&1&1\\1&0&0\\1&0&1\\1&1&0\\1&1&1\end{bmatrix}=\begin{bmatrix}0&0&1\\0&0&0\\0&1&1\\0&1&0\\1&0&0\\1&0&1\\1&1&0\\1&1&1\end{bmatrix}$$

Also

$$Q_3\begin{bmatrix}X=0&Y=0&Z=1\\X=1&Y=0&Z=1\\X=0&Y=1&Z=1\\X=1&Y=1&Z=1\end{bmatrix}=\begin{bmatrix}0\\0\\1\\1\end{bmatrix}=Y$$

or

$$X=\begin{pmatrix}P_1&P_2&P_3&P_4&P_5&P_6&P_7&P_8\\P_2&P_1&P_3&P_4&P_6&P_5&P_7&P_8\end{pmatrix}=(1,2)(5,6)$$

and

$$UP=\begin{bmatrix}0&1&0&0&0&0&0&0\\1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&1\end{bmatrix}\begin{bmatrix}0&0&0\\0&0&1\\0&1&0\\0&1&1\\1&0&0\\1&0&1\\1&1&0\\1&1&1\end{bmatrix}=\begin{bmatrix}0&0&1\\0&0&0\\0&1&0\\0&1&1\\1&0&1\\1&0&0\\1&1&0\\1&1&1\end{bmatrix}$$

For the operation X ∧ Y, we have the permutation

$$Q_3\begin{bmatrix} X=0 & Y=0 & Z=1 \\ X=1 & Y=0 & Z=1 \\ X=0 & Y=1 & Z=1 \\ X=1 & Y=1 & Z=1 \end{bmatrix}=\begin{bmatrix}0\\1\\0\\1\end{bmatrix}=X$$

because

$$X \wedge Y=\begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \\ P_2 & P_1 & P_4 & P_3 & P_6 & P_5 & P_7 & P_8 \end{pmatrix}=(1,2)(3,4)(5,6)$$

or

$$X=(1,2)(5,6) \text{ and } Y=(1,2)(3,4)$$
$$X \wedge Y=\left\{(1,2),(5,6)\right\} \cup \left\{(1,2),(3,4)\right\}=\left\{(1,2),(3,4),(5,6)\right\}$$

and

$$UP=\begin{bmatrix} 0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}\begin{bmatrix} 0&0&0 \\ 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{bmatrix}=\begin{bmatrix} 0&0&1 \\ 0&0&0 \\ 0&1&1 \\ 0&1&0 \\ 1&0&1 \\ 1&0&0 \\ 1&1&0 \\ 1&1&1 \end{bmatrix}$$

For the operation X ∨ Y, we have the permutation

$$Q_3\begin{bmatrix} X=0 & Y=0 & Z=1 \\ X=1 & Y=0 & Z=1 \\ X=0 & Y=1 & Z=1 \\ X=1 & Y=1 & Z=1 \end{bmatrix}=\begin{bmatrix}0\\0\\0\\1\end{bmatrix}=X \wedge Y$$

and because

$$X \vee Y=\begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \\ P_2 & P_1 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \end{pmatrix}=(1,2)$$

and

$$X=\left\{(1,2),(5,6)\right\}, X=\left\{(1,2),(3,4)\right\}$$
$$X \vee Y=\left\{(1,2),(5,6)\right\} \cap \left\{(1,2),(3,4)\right\}=\left\{(1,2)\right\}$$

or

$$UP=\begin{bmatrix} 0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix}\begin{bmatrix} 0&0&0 \\ 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{bmatrix}=\begin{bmatrix} 0&0&1 \\ 0&0&0 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{bmatrix}$$

For the negation operation, we generate a complementary set of permutations and

$$Q_3\begin{bmatrix} X=0 & Y=0 & Z=1 \\ X=1 & Y=0 & Z=1 \\ X=0 & Y=1 & Z=1 \\ X=1 & Y=1 & Z=1 \end{bmatrix}=\begin{bmatrix}0\\1\\1\\1\end{bmatrix}=X \vee Y$$

or

$$\neg X=\begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \\ P_1 & P_2 & P_4 & P_3 & P_5 & P_6 & P_8 & P_7 \end{pmatrix}=\{(3,4)(7,8)\}$$
$$X=\left\{(1,2),(5,6)\right\}$$
$$\neg X=\left\{(1,2),(5,6)\right\}^C=\left\{(3,4),(7,8)\right\}$$

$$UP=\begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \end{bmatrix}\begin{bmatrix} 0&0&0 \\ 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{bmatrix}=\begin{bmatrix} 0&0&0 \\ 0&0&1 \\ 0&1&1 \\ 0&1&0 \\ 1&0&0 \\ 1&0&1 \\ 1&1&1 \\ 1&1&0 \end{bmatrix}$$

For contradiction we have

$$Q_3 \begin{bmatrix} X=0 & Y=0 & Z=1 \\ X=1 & Y=0 & Z=1 \\ X=0 & Y=1 & Z=1 \\ X=1 & Y=1 & Z=1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \neg X$$

Thus, we have the Unitary transformation

$$Q_3 \begin{bmatrix} X=0 & Y=0 & Z=1 \\ X=1 & Y=0 & Z=1 \\ X=0 & Y=1 & Z=1 \\ X=1 & Y=1 & Z=1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \neg X \wedge X$$

and

$$\text{UP}=\begin{bmatrix} 0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \end{bmatrix}\begin{bmatrix} 0&0&0 \\ 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{bmatrix} = \begin{bmatrix} 0&0&1 \\ 0&0&0 \\ 0&1&1 \\ 0&1&0 \\ 1&0&1 \\ 1&0&0 \\ 1&1&1 \\ 1&1&0 \end{bmatrix}$$

This is due to

$$X \wedge \neg X = \begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \\ P_2 & P_1 & P_3 & P_3 & P_6 & P_5 & P_8 & P_7 \end{pmatrix} = (1,2)(3,4)(5,6)(7,8)$$

and

$$X = \left\{(1,2),(5,6)\right\}, \qquad \neg X = \left\{(3,4),(7,8)\right\}$$

Next by using the De Morgan rule for negation

$$X \wedge \neg X = \left\{(1,2),(5,6)\right\} \cup \left\{(3,4),(7,8)\right\} = \left\{(1,2),(3,4),(5,6),(7,8)\right\}$$

we get

$$\neg(X \wedge Y) = (\neg X \vee \neg Y)$$

Thus,

$$\neg(X \wedge Y) = \begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \\ P_2 & P_1 & P_4 & P_3 & P_6 & P_5 & P_7 & P_8 \end{pmatrix} = (7,8) = XNANDY$$

$$\neg(X \wedge Y) = \left\{(1,2),(3,4),(5,6)\right\}^C = \left\{(7,8)\right\}$$

or

$$\text{UP}=\begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \end{bmatrix}\begin{bmatrix} 0&0&0 \\ 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&0 \\ 1&1&1 \end{bmatrix} = \begin{bmatrix} 0&0&0 \\ 0&0&1 \\ 0&1&0 \\ 0&1&1 \\ 1&0&0 \\ 1&0&1 \\ 1&1&1 \\ 1&1&0 \end{bmatrix}$$

Now having permutations and rules to compute AND, OR and NOT we can generate any function with two variables X and Y. For example,

$$\neg X = \begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \\ P_1 & P_2 & P_4 & P_3 & P_5 & P_6 & P_8 & P_7 \end{pmatrix} = \{(3,4),(7,8)\}$$

$$\neg Y = \begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \\ P_1 & P_2 & P_3 & P_4 & P_6 & P_5 & P_8 & P_7 \end{pmatrix} = \{(5,6),(7,8)\}$$

$$\neg X \vee \neg Y = \left\{(3,4),(7,8)\right\} \cap \left\{(5,6)(7,8)\right\} = \left\{(7,8)\right\} = \neg(X \wedge Y)$$

Finally in quantum computer, given set of elementary permutation

$$\Gamma = \{(1,2), (3,4), (5,6), (7,8)\}$$

any Boolean function of two variables is a subset of Γ.

Agents theory AUT in the quantum interpretation of logic. For the quantum mechanics the true and false value or qbit is given by the elementary vectors

$$\neg X = \begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \\ P_1 & P_2 & P_4 & P_3 & P_5 & P_6 & P_8 & P_7 \end{pmatrix} = \{(3,4),(7,8)\}$$

$$Y = \begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 & P_7 & P_8 \\ P_1 & P_2 & P_3 & P_4 & P_6 & P_5 & P_8 & P_7 \end{pmatrix} = \{(1,2),(3,4)\}$$

$$X \rightarrow Y = \neg X \vee Y = \left\{(3,4),(7,8)\right\} \cap \left\{(1,2)(3,4)\right\} = \left\{(3,4)\right\}$$

where

$$true = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; false = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

We define a quantum agent as any entity that can define the quantum logic values $\left|\psi\right\rangle = \alpha\left|true\right\rangle + \beta\left|false\right\rangle$

now $\left|\psi\right\rangle$ is function of the parameters $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$

If these parameters are equal to $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ then ,

$\left|\psi\right\rangle = \left|true\right\rangle$

If these parameters are equal to $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ then

$\left|\psi\right\rangle = \left|false\right\rangle$

where false can be spin down and true can be spin up. Now for different particles a set of logic values v(p) can be defined by using quantum agents,

$$\left|\psi\right\rangle = \left|true\right\rangle \;,\; \left|\psi\right\rangle = \left|false\right\rangle$$

and p is a logic proposition, that is p = " the particle is in the state $v(p) = \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ \nu_n \end{bmatrix}$ where $\nu_k \in \{true, false\} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$ ". For example the particle can be in the state up for the spin or the particle has the velocity v and so on.

<u>Definition.</u>A qagent is an agent that performs a quantum measure.

The quantum agents or "qagent" and the evaluation vector can be written in the language of AUT in this usual way

$\left|\psi\right\rangle$

where

$$v(p) = \begin{bmatrix} qagent_1 & qagent_2 & \dots & qagent_n \\ v_1 & v_2 & \dots & v_n \end{bmatrix}$$

$$v_k \in \{(1,2),(5,6)\}$$

and

$$v(q) = \begin{bmatrix} qagent_1 & qagent_2 & \dots & qagent_n \\ \eta_1 & \eta_2 & \dots & \eta_n \end{bmatrix}$$

Now we use the previous definition of the logic in quantum mechanics, thus

$$\eta_k \in \{(1,2),(3,4)\}$$

where

$$v(p \wedge q) = \begin{bmatrix} qagent_1 & qagent_2 & \dots & qagent_n \\ v_1 \wedge \eta_1 & v_2 \wedge \eta_2 & \dots & v_n \wedge \eta_n \end{bmatrix}$$

Now for

$$v_k \wedge \eta_k \in \{(1,2),(5,6)\} \cup \{(1,2),(3,4)\} = \{(1,2),(3,4),(5,6)\}$$

where

$$v(p \vee q) = \begin{bmatrix} qagent_1 & qagent_2 & \dots & qagent_n \\ v_1 \vee \eta_1 & v_2 \vee \eta_2 & \dots & v_n \vee \eta_n \end{bmatrix}$$

Finally, we have the logic vector

Table 1. Logic operation AND by qagents with quantum states 1 and 0

p ∧ q	$\lvert 1\wedge 1\rangle, \lvert 0\wedge 0\rangle, \lvert 1\wedge 0\rangle, \lvert 0\wedge 1\rangle$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$
$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$
$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$
$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$
$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$

$$v_k \vee \eta_k \in \left\{(1,2),(5,6)\right\} \cap \left\{(1,2),(3,4)\right\} = \left\{(1,2)\right\}$$

where $S_1, \ldots, S_n$ are logic value true and false defined by logic operation

For example we have

$$v(p \bullet q) = \begin{bmatrix} qagent_1 & qagent_2 & .... & qagent_n \\ S_1 & S_2 & .... & S_n \end{bmatrix}$$

for which p • q = p ∨ q, that is represented by the subset S = { (1, 2)}, $S_1, \ldots, S_n$ are different logic values of the operation union generated by the unitary matrix in quantum mechanics and the quantum measure. Now with the agent interpretation of the quantum mechanics, we can use the quantum superposition principle and obtain the aggregation of the qagents logic values.

$$v(p \vee q) = \begin{bmatrix} qagent_1 & qagent_2 & .... & qagent_n \\ S_1 & S_2 & .... & S_n \end{bmatrix}$$

For instance, for two conflicting qagents we have the superposition

$$\mu(p \bullet q) = \lvert p \bullet q\rangle = w_1 \lvert S_1\rangle + w_2 \lvert S_2\rangle + ... + w_n \lvert S_n\rangle$$

Thus, the AUT establishes a bridge between the quantum computer based on the Boolean logic with the many-valued logic and conflicts based on the quantum superposition phenomena. These processes are represented by fusion process in AUT. The example of many-valued logic in quantum computer is given in Table 1. In the table 1 any qagent can assume states $\mu(p) = \lvert p\rangle = w_1 \lvert S_1\rangle + w_2 \lvert S_2\rangle = w_1 \lvert 1\rangle + w_2 \lvert 0\rangle$ for a single proposition. For two different propositions p and q, a qagent can assume one of the following possible states:

$$\lvert 1\rangle \text{ or } \lvert 0\rangle$$

Also for the superposition we have Table 2.

Now we can assign a fractional value to the true logic value as shown in Table 3.

Table 2. Quantum superposition as logic fusion

p ∧ q	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$
$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	$\alpha \lvert 1\rangle + \beta \lvert 1\rangle$	$\alpha \lvert 1\rangle + \beta \lvert 0\rangle$	$\alpha \lvert 0\rangle + \beta \lvert 1\rangle$
$\alpha \lvert 0\rangle + \beta \lvert 0\rangle$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\alpha \lvert 1\rangle + \beta \lvert 0\rangle$	$\alpha \lvert 1\rangle + \beta \lvert 0\rangle$	$\alpha \lvert 0\rangle + \beta \lvert 0\rangle$
$\alpha \lvert 0\rangle + \beta \lvert 0\rangle$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	$\alpha \lvert 0\rangle + \beta \lvert 1\rangle$	$\alpha \lvert 0\rangle + \beta \lvert 0\rangle$	$\alpha \lvert 0\rangle + \beta \lvert 1\rangle$
$\alpha \lvert 0\rangle + \beta \lvert 0\rangle$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\alpha \lvert 0\rangle + \beta \lvert 0\rangle$	$\alpha \lvert 0\rangle + \beta \lvert 0\rangle$	$\alpha \lvert 0\rangle + \beta \lvert 0\rangle$

Table 3. Many-valued logic in the quantum system

p ∧ q	$\alpha \lvert 0\rangle + \beta \lvert 0\rangle$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$
$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	True	½ True	½ True	False
$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	½ True	½ True	False	False
$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 1\rangle & \nu_2 = \lvert 0\rangle \end{pmatrix}$	½ True	False	½ True	False
$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = \lvert 0\rangle & \nu_2 = \lvert 1\rangle \end{pmatrix}$	False	False	False	False

Correlation (entanglement) in quantum mechanics and second order conflict. Consider two interacting particles as agents. These agents are interdependent. Their correlation is independent from any physical communication by any type of fields. We can say that this correlation is rather a characteristic of a logical state of the particles than dependence. We can view the quantum correlation as a conflicting state because we know from quantum mechanics that there is a correlation but when we try to measure the correlation, we cannot check the correlation itself. The measurement process destroys the correlation. Thus, if the spin of one electron is up and the spin of another

Figure 8. Many-valued logic in the AUT neural network

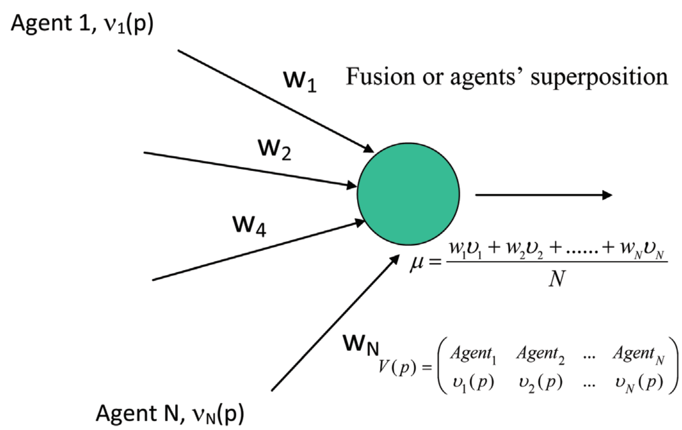
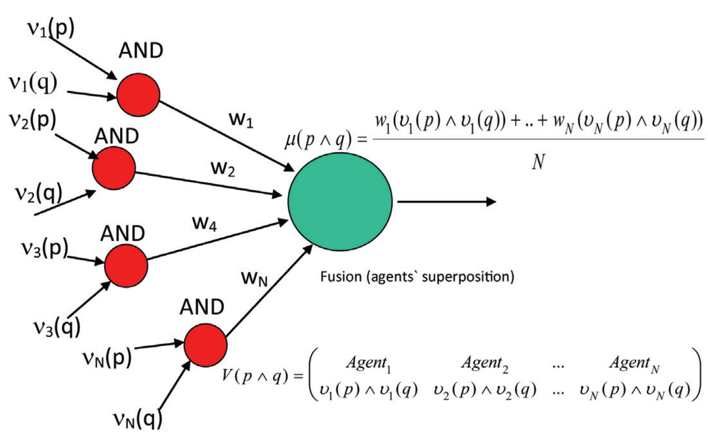


Figure 9. Many-valued logic operation AND in the AUT neural network



electron is down the first spin is changed when we have correlation or entanglement. It generates the change of the other spin instantaneously and this is manifested in a statistic correlation different from zero. For more explanation see D'Espagnat (1999).

NEURAL IMAGE OF AUT

In this section, we show the possibility for a new type of neural network based on the AUT. This type of the neural network is dedicated to computation of many-valued logic operations used to model uncertainty process. The traditional neural networks model Boolean operations and classical logic. In a new fuzzy neural network, we combine two logic levels (classical and fuzzy) in the same neural network as presented in Figure 8.

Figures 9-11 show that at the first level we have the ordinary Boolean operations (AND, OR, and NOT). At the second level, the network fuses results of different Boolean operations. As a result, a many value logic value is generated as Figure 8 shows.

Figure 10. AUT Many-valued logic operation OR in the AUT neural network

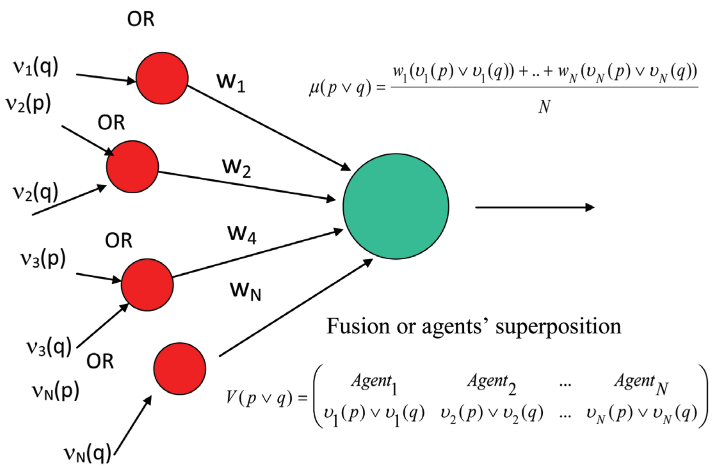
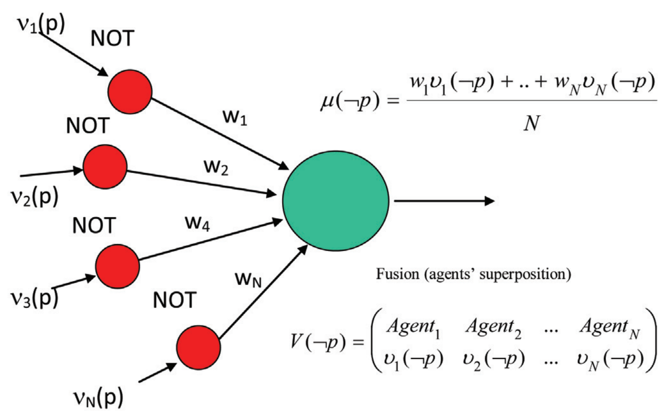


Figure 11.Many-valued logic operation NOT in the AUT neural network



Now we show an example of many-valued logic operation AND with the agents and fusion in the neural network. Table 4 presents the individual AND operation for a single agent in the population of two agents.

$$\begin{pmatrix} qagent_1 & qagent_2 \\ \nu_1 = |0\rangle & \nu_2 = |0\rangle \end{pmatrix}$$

With the fusion process in AUT we have the many-valued logic in the neural network.

In Table 5 we use the aggregation rule that can generate a many-valued logic structure with the following three logic values:

$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$ with equivalent notations

$$\Omega = \left\{ true, \frac{true}{2} = \frac{false}{2}, false \right\}$$

Now, having the commutative rule we derive,

$$\frac{false + false}{2} = false$$

The previous composition rule can be written in the simple form shown in Table 6 where different

Table 4. Agent Boolean logical rule

p ∧ q	$V(p \vee q) = \begin{pmatrix} Agent_1 & Agent_2 & \dots & Agent_N \\ v_1(p) \vee v_1(q) & v_2(p) \vee v_2(q) & \dots & v_N(p) \vee v_N(q) \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ true & true \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ true & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & true \end{pmatrix}$
$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ true & true \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ true & true \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ true & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & true \end{pmatrix}$
$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ true & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ true & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ true & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$
$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & true \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & true \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & true \end{pmatrix}$
$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$	$\begin{pmatrix} nagent_1 & nagent_2 \\ false & false \end{pmatrix}$

Table 5. Neuronal fusion process

p ∧ q	$\frac{true + false}{2} = \frac{false + true}{2} = \frac{true}{2} = \frac{false}{2} = \frac{1}{2} true = \frac{1}{2} false$	$\frac{true + true}{2} = true$	$\frac{true + false}{2} = \frac{true}{2}$	$\frac{false + true}{2} = \frac{true}{2}$
$\frac{false + false}{2} = false$	$\frac{true + true}{2} = true$	$\frac{true + true}{2} = true$	$\frac{true + false}{2} = \frac{true}{2}$	$\frac{false + true}{2} = \frac{true}{2}$
$\frac{false + false}{2} = false$	$\frac{true + false}{2} = \frac{true}{2}$	$\frac{true + false}{2} = \frac{true}{2}$	$\frac{true + false}{2} = \frac{true}{2}$	$\frac{false + false}{2} = false$
$\frac{false + false}{2} = false$	$\frac{false + true}{2} = \frac{true}{2}$	$\frac{false + true}{2} = \frac{true}{2}$	$\frac{false + false}{2} = false$	$\frac{false + true}{2} = \frac{true}{2}$
$\frac{false + false}{2} = false$	$\frac{false + false}{2} = false$	$\frac{false + false}{2} = false$	$\frac{false + false}{2} = false$	$\frac{false + false}{2} = false$

results for the same pair of elements are located in the same cell.

Table 6 contains two different results for the AND operation,

$$\frac{false + false}{2} = false$$

for p = ½ true and q = ½ true.

In this case we have no criteria to choose one or the other. Here the operation AND is not uniquely defined, we have two possible results one is false and the other is ½ true. The first one is shown in Table 7 and the second one is shown in Table 8.

The neuron image of the previous operation is shown in Figure 12.

Table 6. Simplification of the Table 5

p ∧ q	$\frac{true+false}{2}=\frac{false+true}{2}=\frac{true}{2}$	$\frac{true+true}{2}=true$	$\frac{true}{2}$
$\frac{false+false}{2}=false$	$\frac{true+true}{2}=true$	$\frac{true+true}{2}=true$	$\frac{true+false}{2}=\frac{true}{2}$
$\frac{false+false}{2}=false$	$\frac{true}{2}$	$\frac{false+true}{2}=\frac{true}{2}$ $\frac{false+false}{2}=false$	$\frac{false+true}{2}=\frac{true}{2}$
$\frac{false+false}{2}=false$	$\frac{false+false}{2}=false$	$\frac{false+false}{2}=false$	$\frac{false+false}{2}=false$

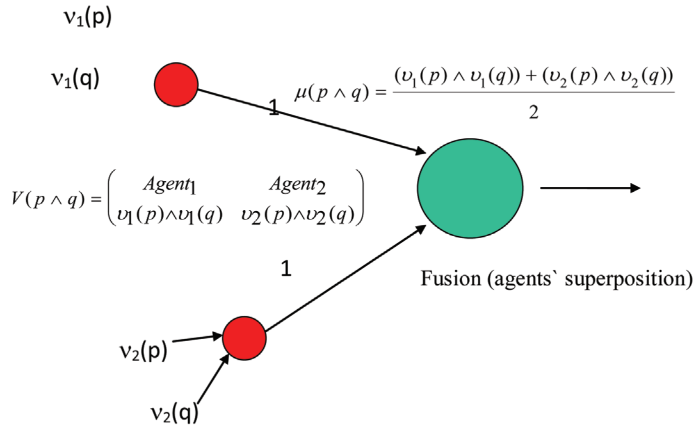
Table 7. First operation

p ∧ q	$p\wedge q=\frac{true}{2}\wedge\frac{true}{2}=\left\{false,\ \frac{true}{2}\right\}$	$\frac{true+true}{2}=true$	$\frac{true}{2}$
$\frac{false+false}{2}=false$	$\frac{true+true}{2}=true$	$\frac{true+true}{2}=true$	$\frac{true+false}{2}=\frac{true}{2}$
$\frac{false+false}{2}=false$	$\frac{true}{2}$	$\frac{false+true}{2}=\frac{true}{2}$	$\frac{false+false}{2}=false$
$\frac{false+false}{2}=false$	$\frac{false+false}{2}=false$	$\frac{false+false}{2}=false$	$\frac{false+false}{2}=false$

Table 8. Second operation

p ∧ q	$\frac{false+false}{2}=false$	$\frac{true+true}{2}=true$	$\frac{true}{2}$
$\frac{false+false}{2}=false$	$\frac{true+true}{2}=true$	$\frac{true+true}{2}=true$	$\frac{true+false}{2}=\frac{true}{2}$
$\frac{false+false}{2}=false$	$\frac{true}{2}$	$\frac{false+true}{2}=\frac{true}{2}$	$\frac{true}{2}$
$\frac{false+false}{2}=false$	$\frac{false+false}{2}=false$	$\frac{false+false}{2}=false$	$\frac{false+false}{2}=false$

Figure 12. The neuron image of the operation presented in Table 6



CONCLUSION

In this chapter, we have summarized the Agent-based Uncertainty Theory (AUT) and had shown how it can model the inconsistent logic values of the quantum computer and represent a new many-valued logic neuron.

We demonstrated that the AUT modeling of the inconsistent logic values of the quantum computer is a way to solve this old problem of inconsistency in quantum mechanics and quantum computer with mutually exclusive states. The general classical approach is based on states and differential equations without introduction of elaborated logic analysis. In the quantum computer, the unitary transformation has been introduced in the literature as media to represent classical logic operations in the form of Boolean calculus in the quantum phenomena. In the quantum computer, superposition gives us a new type of logic operation for which the mutual exclusion principle is not true. The same particle can be in two different positions in the same time. This chapter explained in a formal way how the quantum phenomena of the superposition can be formalized by the AUT

Many uncertainty theories emerged such as fuzzy set theory, rough set theory, and evidence theory. The AUT allows reformulating them in a similar way as we did for the quantum mechanics and quantum computer in this chapter. Now with the AUT, it is possible to generate neuron models and fuzzy neuron models that deal with intrinsic conflicting situations and inconsistency. The AUT opens the opportunities to rebuild previous models of uncertainty with the explicit and consistent introduction of the conflicting and inconsistent phenomena by the means of the many-valued logic.

REFERENCES

Abbott, A., Doering, C., Caves, C., Lidar, D., Brandt, H., & Hamilton, A. (2003). Dreams versus Reality: Plenary Debate Session on Quantum Computing. *Quantum Information Processing*, *2*(6), 449–472. doi:10.1023/B:QINP.0000042203.24782.9a

Atanassov, K. T. (1999). *Intuitionistic Fuzzy Sets, Physica Verlag*. Heidelberg: Springer.

Baki, B., Bouzid, M., Ligęza, A., & Mouaddib, A. (2006). A centralized planning technique with temporal constraints and uncertainty for multi-agent systems. *Journal of Experimental & Theoretical Artificial Intelligence*, *18*(3), 331–364. doi:10.1080/09528130600906340

Benenti, G. (2004). *Principles of Quantum Computation and Information* (*Vol. 1*). New Jersey: World Scientific.

Carnap, R., & Jeffrey, R. (1971). *Studies in Inductive Logics and Probability* (*Vol. 1*, pp. 35–165). Berkeley, CA: University of California Press.

Chalkiadakis, G., & Boutilier, C. (2008). Sequential Decision Making in Repeated Coalition Formation under Uncertainty, In: Proc. of 7th Int. Conf. on Autonomous Agents and Multi-agent Systems (AA-MAS 2008), Padgham, Parkes, Müller and Parsons (eds.), May, 12-16, 2008, Estoril, Portugal, http://eprints.ecs.soton.ac.uk/15174/1/BayesRLCF08.pdf

Colyvan, M. (2004). The Philosophical Significance of Cox's Theorem. *International Journal of Approximate Reasoning*, *37*(1), 71–85. doi:10.1016/j.ijar.2003.11.001

Colyvan, M. (2008). Is Probability the Only Coherent Approach to Uncertainty? *Risk Analysis*, *28*, 645–652. doi:10.1111/j.1539-6924.2008.01058.x

D'Espagnat, B. (1999). *Conceptual Foundation of Quantum mechanics* (2nd ed.). Perseus Books.

DiVincenzo, D. (1995). Quantum Computation. *Science*, *270*(5234), 255–261. doi:10.1126/science.270.5234.255

DiVincenzo, D. (2000). The Physical Implementation of Quantum Computation. *Experimental Proposals for Quantum Computation*. arXiv:quant-ph/0002077

Edmonds, B. (2002). Review of Reasoning about Rational Agents by Michael Wooldridge. *Journal of Artificial Societies and Social Simulation*, *5*(1). Retrieved from http://jasss.soc.surrey.ac.uk/5/1/reviews/edmonds.html.

Fagin, R., & Halpern, J. (1994). Reasoning about Knowledge and Probability. *Journal of the ACM*, *41*(2), 340–367. doi:10.1145/174652.174658

Ferber, J. (1999). *Multi Agent Systems*. Addison Wesley.

Feynman, R. (1982). Simulating physics with computers. *International Journal of Theoretical Physics*, *21*, 467. doi:10.1007/BF02650179

Flament, C. (1963). *Applications of graphs theory to group structure*. London: Prentice Hall.

Gigerenzer, G., & Selten, R. (2002). *Bounded Rationality*. Cambridge: The MIT Press.

Halpern, J. (2005). *Reasoning about uncertainty*. MIT Press.

Harmanec, D., Resconi, G., Klir, G. J., & Pan, Y. (1995). On the computation of uncertainty measure in Dempster-Shafer theory. *International Journal of General Systems*, *25*(2), 153–163. doi:10.1080/03081079608945140

Hiroshi, I., & Masahito, H. (2006). *Quantum Computation and Information*. Berlin: Springer.

Hisdal, E. (1998). *Logical Structures for Representation of Knowledge and Uncertainty*. Springer.

Jaeger, G. (2006). *Quantum Information: An Overview*. Berlin: Springer.

Kahneman, D. (2003). Maps of Bounded Rationality: Psychology for Behavioral Economics. *The American Economic Review*, *93*(5), 1449–1475. doi:10.1257/000282803322655392

Kovalerchuk, B. (1990). Analysis of Gaines' logic of uncertainty, In I.B. Turksen (Ed.), *Proceedings of NAFIPS '90* (Vol. 2, pp. 293-295).

Kovalerchuk, B. (1996). Context spaces as necessary frames for correct approximate reasoning. *International Journal of General Systems*, *25*(1), 61–80. doi:10.1080/03081079608945135

Kovalerchuk, B., & Vityaev, E. (2000). *Data mining in finance: advances in relational and hybrid methods*. Kluwer.

Montero, J., Gomez, D., & Bustine, H. (2007). On the relevance of some families of fuzzy sets. *Fuzzy Sets and Systems*, *16*, 2429–2442. doi:10.1016/j.fss.2007.04.021

Nielsen, M., & Chuang, I. (2000). *Quantum Computation and Quantum Information*. Cambridge: Cambridge University Press.

Priest, G., & Tanaka, K. Paraconsistent Logic. (2004). Stanford Encyclopedia of Philosophy. http://plato.stanford.edu/entries/logic-paraconsistent.

Resconi, G., & Jain, L. (2004). *Intelligent agents*. Springer Verlag.

Resconi, G., Klir, G. J., Harmanec, D., & St. Clair, U. (1996). Interpretation of various uncertainty theories using models of modal logic: a summary. *Fuzzy Sets and Systems*, *80*, 7–14. doi:10.1016/0165-0114(95)00262-6

Resconi, G., Klir, G. J., & St. Clair, U. (1992). Hierarchical uncertainty metatheory based upon modal logic. *International Journal of General Systems*, *21*, 23–50. doi:10.1080/03081079208945051

Resconi, G., Klir, G.J., St. Clair, U., & Harmanec, D. (1993). The integration of uncertainty theories. *Intern. J. Uncertainty Fuzziness knowledge-Based Systems,* 1, 1-18.

Resconi, G., & Kovalerchuk, B. (2006). The Logic of Uncertainty with Irrational Agents In *Proc. of JCIS-2006 Advances in Intelligent Systems Research, Taiwan*. Atlantis Press

Resconi, G., Murai, T., & Shimbo, M. (2000). Field Theory and Modal Logic by Semantic field to make Uncertainty Emerge from Information. *International Journal of General Systems*, *29*(5), 737–782. doi:10.1080/03081070008960971

Resconi, G., & Turksen, I. B. (2001). Canonical Forms of Fuzzy Truthoods by Meta-Theory Based Upon Modal Logic. *Information Sciences*, *131*, 157–194. doi:10.1016/S0020-0255(00)00095-5

Ruspini, E. H. (1999). A new approach to clustering. *Information and Control*, *15*, 22–32. doi:10.1016/S0019-9958(69)90591-9

Stolze, J., & Suter, D. (2004). *Quantum Computing*. Wiley-VCH. doi:10.1002/9783527617760

Sun, R., & Qi, D. (2001). Rationality Assumptions and Optimality of Co-learning, In *Design and Applications of Intelligent Agents* (LNCS 1881, pp. 61-75). Berlin/Heidelberg: Springer.

van Dinther, C. (2007). *Adaptive Bidding in Single-Sided Auctions under Uncertainty: An Agent-based Approach in Market Engineering (Whitestein Series in Software Agent Technologies and Autonomic Computing)*. Basel: Birkhäuser.

Vandersypen, L.M.K., Yannoni, C.S., & Chuang, I.L. (2000). *Liquid state NMR Quantum Computing*.

Von-Wun Soo. (2000). Agent Negotiation under Uncertainty and Risk In *Design and Applications of Intelligent Agents* (LNCS 1881, pp. 31-45). Berlin/Heidelberg: Springer.

Wooldridge, M. (2000). *Reasoning about Rational Agents*. Cambridge, MA: The MIT Press.

Wu, W., Ekaette, E., & Far, B. H. (2003). Uncertainty Management Framework for Multi-Agent System, Proceedings of ATS http://www.enel.ucalgary.ca/People/far/pub/papers/2003/ATS2003-06.pdf

KEY TERMS AND DEFINITIONS

Logic of Uncertainty: A field that deals with logic aspects of uncertainty modeling.

Conflicting Agents: Agents that have self-conflict or conflict with other agents in judgment of truth of specific statements.

Fuzzy Logic: A field that deals with modeling uncertainty based on Zadeh's fuzzy sets.

The Agent–Based Uncertainty Theory: (AUT): A theory that model uncertainty using the concept of conflicting agents.

Neural Network: Used in this chapter to denote any type of an artificial neural network.

Quantum Computer: Used in this chapter to denote any systems that provide computations based on qubits.

Fusion: Used in this chapter to denote a fusion of multi-dimensional conflicting judgments of agents.