Thematic Oral Session

METHODOLOGICAL ISSUES

# ESSAYING THE PROLOG LANGUAGE TO OBTAIN THE COMPUTATIONAL EVALUATION OF THE COHERENCE OF REASONING

Alfredo O. **López Alonso** * and María Herminia **Del Rey** **

Resumen

Se presenta una descripción de la automatización intentada para evaluar el *Test de Coherencia de Razonamiento* (TCR, López Alonso, 1981, 1988, 1996, 2000) a través de programas lógicos. En primer lugar, se presentan las aplicaciones hechas utilizando el programa Prolog. Si bien no es secuencial, Prolog es un programa que provee hechos y reglas de derivación de conclusiones y procedimientos que son propios del razonamiento deductivo y que son utilizados para resolver problemas de programación. Sus características operativas surgen de cláusulas que son propias de la lógica de predicados. Esta lógica utiliza los mismos caminos inferenciales que el razonamiento expresado en términos lingüísticos. En consideración a estas características se llevaron a cabo muchos ensayos para automatizar la evaluación del TCR. Algunos de estos ensayos empíricos y sus resultados se exponen y discuten brevemente. También se analizan sucintamente los ajustes y desajustes de programa, encontrados hasta la fecha entre el lenguaje Prolog y los supuestos básicos del TCR.

* Psychological Doctor. Researcher of Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET). Director of Instituto de Investigaciones Psicológicas de la Universidad del Salvador (IIPUS). E-Mail: alalonso@ciudad.com.ar

** Specialist in Scientific Computation. Support Professional of Instituto de Investigaciones Psicológicas de la Universidad del Salvador (IIPUS).

*Palabras clave*: Test de Coherencia de Razonamiento - evaluación automatizada de tests - tests cognitivos - test de isomorfismo entre el razonamiento personal y la lógica formal.

## Abstract

Automation in evaluating the *Test of Reasoning Coherence* (TRC, López Alonso, 1981, 1988, 1996, 2000) is attempted by using logical programs. First in being applied is the Prolog program. Although non-sequential, Prolog provides facts, and rules deriving into conclusions and uses deductive reasoning to solve programming problems. Its working characteristics stem from clauses of predicate logic. This logic uses the same paths as reasoning when linguistically carried out. Considering these characteristics many essays were attempted to automate the TRC evaluation. Some empirical essays and results are briefly exposed and discussed. Program adjustments, and maladjustment found up-to-date between Prolog language, and TRC basic assumptions are succinctly analyzed.

*Key words*: Test of Reasoning Coherence - tests automated evaluation - cognitive test - test of isomorphism between personal reasoning and formal logic.

The idea to evaluate the coherence of reasoning comes from the employment of the Test of Reasoning Coherence (TRC - López Alonso, 1981, 1988, 1996, 2000; Orsi, 1988) an instrument that has been devised using modal expressions of the copula, such as *has (have) to be, cannot be,* and *may be* (see Figure 1). These three relationships permit to state both propositional, and set relations between terms by using modal expressions of the copula.

The relationship *Has to be* (Figure 2) implies the set inclusion relation, or well, the propositional logical implication. For example, "*A has to be B*" is graphically represented as "*A is included in B*", or as "*A implies B*". As

we can see in the graphs of Figure 2, there are two ways of diagramming this relationship, as the separate or embedded Euler diagrams on the upper part of the display, or well as the overlapping similar-typed Venn diagrams below where the external rectangle represents the universal set, and the (+), and the (-) signs define the resultant non-empty, and empty intersections, respectively. In terms of modal logic this relationship implies the necessity of *A to be B*, or that *A must be B*, or *A is B* necessarily. The relationship *Has to be* is noted *1*; and, for the case, *"A1B"* is the symbol brief notation corresponding to the above expression.

The relationship *Cannot be* (Figure 3) implies a set exclusion or an incompatibility relationship. In modal terms, it implies the impossibility of *A to be B*, and reversely the impossibility of *B to be A*. But in terms of modal logic, *Cannot be* also implies the necessity of *A to be non-B*, and of *B to be non-A*. Again, using our diagramming system, we have the Euler-type separate diagrams above, and the Venn-type overlapping diagrams below where the empty set corresponds to the overlapping intersection, given the (-) sign, in the diagram. The relationship *Cannot be* is noted *0*; for the case being,*"A0B"* is the corresponding symbol brief notation.

The relationship *May be* (Figure 4) implies that both "*A is B*", and "*A is not-B*" are possible, or the case of inclusive disjunction in proposition logic, where both can be true, although not both false. There is no necessity nor impossibility or incompatibility relationships in this modal expression between *A* and *B*. It implies the union of intersections *A.B,* and *A.-B*. Then, in terms of modal logic, it implies the *possibility* or the *contingency* of *A* to be *B* or of *A* to be *non-B*. So, the relationship "*may be*" is noted *P* and the expression above is reduced to "*APB*" in the corresponding brief notation.

So, an instrument as the TRC permits to detect the internal contradictions hidden in the reasoner's inferential representations as formed from those modal relations.

How can we manage to device a logical computational program involving the above modal rules and inferential procedures? What is Prolog to this purpose? Up to what point can Prolog help us to define the program in order to evaluate logically and computationally the TRC subjects' answers?

Spivey (1996) states that a more encouraging attempt to this purpose is to find theories, and programming paradigms that link together different ways of understanding programs, and computer systems, and the purpose of his book is to explore up to what extent logic programming provides such a theory. Through the medium of logic, we can separate the task of capturing

the problem from the task of finding an effective way to solve it. He also states that the implementation of Prolog can be viewed as carrying out symbolic reasoning with logical formulas, and its correctness is expressed in the fact that it faithfully realizes the inference rule of resolution, which is itself sound with respect to the declarative meaning of the program.

A program is a collection of instructions for carrying out some computing task. Then, high-level languages like Fortran and Algol 60 were invented to ease the programming task, and the successors of those languages, including Pascal, C, and Ada, are still in use today. The question "What kind of thing is a program?" stems from languages like Lisp and like Prolog -this one as of special interest for us in this case. Prolog is used for writing computer programs that model human thinking. Prolog is a programming language developed in the early 1970s by Alain Colmerauer (University of Marseilles) and standardized by the ISO (International Organization for Standardization) in 1995. Prolog exemplifies *logic programming*, a kind of programming developed by Robert Kowalski of the University of London.

In ordinary programming, a program describes the steps that a computer is to work through in order to solve a problem. In logic programming, the program gives the computer facts about the problem, plus rules by means of which other facts can be inferred. In the case of the TRC the three relationships defined above provide the rules, while the options of the subjects given as a determined series of *1, 0*, and *P* relationships in the order in which they have been chained along by the subject coherently, constitute the computer facts about the problem.

The distinguishing feature of these *declarative programming languages*, at least in their pure formats, is that programs are made up not of commands to be executed, but of definitions and statements about the problem itself to be solved. Grammatically, they are in the declarative mood, used for ordinary statements in natural language. Contrarily to imperative programs, declarative programs contain no explicit instructions to be followed by the computer that executes them. Instead, the job of the computer is to manipulate the information contained in the program so as to derive the solution of a given problem. In logic programming a program consists of a collection of statements expressed as formulas in symbolic terms. Because rules of inference can be expressed in purely symbolic terms, applying them is the kind of symbol-manipulation that can be carried out by a computer. Some predicates, and extensional, and relational clauses will be essayed.

## Essaying Prolog for the TRC

*Predicates:* (XXXX)

1.- *Extensional Premises:*
Percentage of A = AX% so that ($0 \leq X \leq 100$)
Percentage of B = BX% so that ($0 \leq X \leq 100$)

2.- *Relational Premises: for A : B, A* $\leq$ B
3.- *Immediate Inferences Consequences:* (See Table 1)

*Clauses*

*Extensional and percentual clauses*:

*Has to be (A,B)*: -AX% $\leq$ BX%, ABX% = AX%, A - B% = 0

*Cannot be (A,B)*: -AX% + BX% $\leq$ 100, ABX% = 0 If AX% + B% > 100 then APB

*May be (A,B)*: -AX% > 0, BX% > 0, 0 < AB% < 100, 0 < A - B% < 100

*Relational clauses:*

*Has to be* (A,B): -A1B, that is A $\subseteq$ B, A $\neq$ 0, A.-B = $\varnothing$

*Cannot be* (A,B): -A0B, that is A1- B , A $\subseteq$ -B, A.B = $\varnothing$

*May be* (A,B): -APB, that is A.B $\neq \varnothing$ and A.-B $\neq \varnothing$

*Set of disjoint intersection clauses:*

R intersections of A, B, C, D .....N attributes = $2^n$

R intersections for (N = 4 attributes) $2^4 = 16$

Display of the R disjoint intersections, for N = 4, and R = 16:

| | | | |
|---|---|---|---|
| A -B -C -D | -A B -C -D | -A -B C -D | -A -B -C D |
| A B -C -D | A -B C -D | A -B -C D | -A B C -D |
| -A B -C D | -A -B C D | A B C -D | A B -C D |
| A -B C D | -A B C D | A B C D | -A -B -C -D |

Clauses for the 1, 0, and *P* relations concerning of R disjoint intersections in rectangular diagrams:

- If A1B then all A.-B .....N (intersection) is empty ( - ) = ∅

- If A0B then all A.B.....N (intersection) is empty ( - ) = ∅

- If APB then at least one AB....N and one A -B....N are non-empty, ≠ ∅

Table 1
Possible Series of Consistent 1, 0 and *P* modal relations for the pair attributes A and B

| Equality / Difference Between A,B,U and ϕ | Set Relations as given by Erickson<br>First Relational Premise in modal terms (Relations 1, 0 and *P*)<br>(A1B) (A0B) (APB) | Possible coherent series of immediate inferences between A and B derived from the first relational premise (from second to eighth terms) |
|---|---|---|
| 1) Logical possibilities for: | | |
| A=U, B=U (A=B) | *Id* - - | (B1A)(A0-B)(B0-A)(-A1B)(-B1A)(-A1-B )(-B1-A) |
| A=ϕ, B=U (A≠B) | *Sb.* - - | (B0A)(A1-B)(B1-A)(-A1B)(-B1A)(-A0-B )(-B1-A) |
| U≠A≠ϕ,B=U (A≠B) | *Sb.* - - | (BPA)(A0-B)(BP-A)(-A1B)(-B1A)(-A0-B )(-B1-A) |
| A=U, B≠ϕ (A≠B) | - *(Sp.)* - | (B1A)(A1-B)(B1-A)(-A1B)(-B1A)(-A1-B )(-B0-A) |
| A=ϕ, B=ϕ (A=B) | *Id.* - - | (B1A)(A1-B)(B1-A)(-A0B)(-B0A)(-A1-B )(-B1-A) |
| U≠A≠ϕ,B=ϕ (A≠B) | - *(Sp.)* - | (B1A)(A1-B)(B1-A)(-A0B)(-BPA)(-A1-B )(-BP-A) |
| A=U, U≠B≠ϕ (A≠B) | - - *Sp.* | (B1A)(AP-B)(B0-A)(-A1B)(-B1A)(-A1-B )(-B0-A) |
| A=ϕ, U≠B≠ϕ (A≠B) | *Sb.* - - | (B0A)(A1-B)(B1-A)(-APB)(-B0A)(-AP-B )(-B1-A) |

this table continues

Table 1 (Continuation)
Possible Series of Consistent 1, 0 and *P* modal relations for the pair attributes A and B

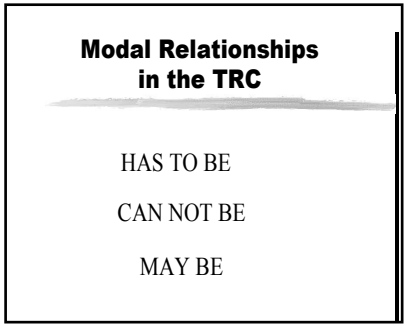
| Equality / Difference Between A,B,U and $\phi$ | Set Relations as given by Erickson<br>First Relational Premise in modal terms (Relations 1, 0 and *P*)<br>(A1B) (A0B) (APB) | | | Possible coherent series of immediate inferences between A and B derived from the first relational premise (from second to eighth terms) |
|---|---|---|---|---|
| 2) Logical possibilities for:<br>$U \neq A \neq \phi$ and $U \neq B \neq \phi$<br>I: For (A=B) | *Id.* | - | - | (B1A)(A0-B)(B0-A)(-A0B)(-B0A)(-A1-B )(-B1-A) |
| II: For ($A \neq B$)<br>II-a: $A \subset B$, $B \not\subset A$ ($AB \neq \phi$) | *Sb.* | - | - | (BPA)(A0-B)(BP-A)(-APB)(-B0A)(-AP-B )(-B1-A) |
| II-b: $A \not\subset B$, $B \subset A$ ($AB \neq \phi$) | - | - | *Sp.* | (B1A)(AP-B)(B0-A)(-A0B)(-BPA)(-A1-B )(-BP-A) |
| II-c: $A \not\subset B$, $B \not\subset A$ ($AB \neq \phi$) | - | - | *Ov.* | (BPA)(AP-B)(BP-A)(-APB)(-BPA)(-AP-B )(-BP-A) |
| II-d: $A \subset$ -B, -$B \not\subset A$($AB = \phi$) | - | *Ex.* | - | (B0A)(A1-B)(B1-A)(-APB)(-BPA)(-AP-B )(-BP-A) |
| II-e: $A \subset$ -B, -$B \subset A$ ($AB = \phi$) | - | *Ex.* | - | (B0A)(A1-B)(B1-A)(-A1B)(-B1A)(-A0-B )(-B0-A) |

*Notation remarks:*
-A and -B is for non-A and non-B sets, respectively.
U is for universal set and $\phi$ is for empty set.

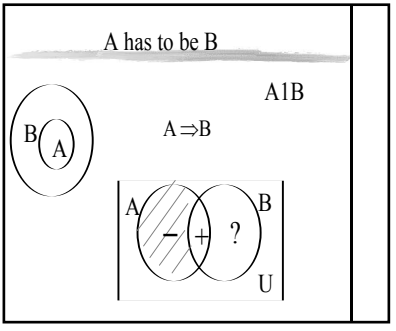
In Erickson (1974, 1978) terms, *Id*. is for identity, *Sb*. for subset, *Sp*. for superset, *Ov*. for overlap, and *Ex.* for exclusion relations between sets; those in parenthe-

ses are debatable.

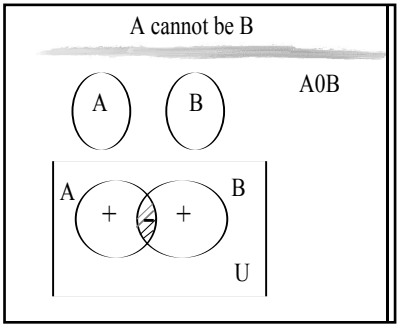
Figures



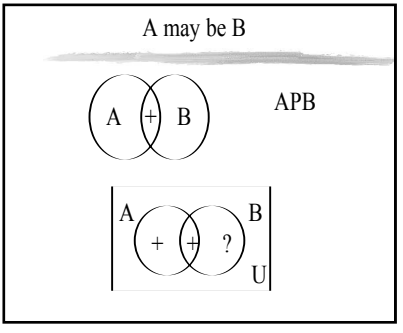

1




2




3




4

## References

Erickson, J.R. (1974). A set analysis theory of behavior in formal syllogistic reasoning tasks. In R.L. Solso (Ed.), *Theories of cognitive psychology: The Loyola Symposium*. Potomac: Erlbaum.

Erickson, J.R. (1978). Research on syllogistic reasoning. In R. Revlin, & R.E. Mayer (Eds.), *Human reasoning.* Washington DC: V.H. Winston.

López Alonso, A.O. (1981). Test de Coherencia [Coherence Test]. *Publicación Nro. 65, CIIPME*.

López Alonso, A.O. (1988). Razonamiento humano: Un test para su consistencia interna [Human reasoning: A test for its inner consistency]. *Revista Signos Universitarios* [Número Especial], *Psicología*, Año 7, *13*, enero/junio, 103-179.

López Alonso, A.O. (1996). La medición de la organización lógica del pensamiento [The measurement of the logic organization thinking's]. In L. Pasquali (Org.), *Teoría e métodos da medida em ciëncias do comportamento* (Cap. 9, pp. 225-261). Edición del Ministerio de Educación y del Deporte de Brasil, Secretaría de Evaluación e Información Educacional, Instituto Nacional de Estudios e Investigaciones Educacionales (SEDIAE/INEP) y Laboratorio de Investigación en Evaluación y Medida - Instituto de Psicología, Universidad de Brasilia: Brasilia, Brasil.

López Alonso, A.O. (2000). Modal reasoning: A cognitive and instrumental approach. *Biennial Review of Research Interchanges between University of Missouri, Saint Louis, and Universidad del Salvador, Buenos Aires, for Psychology, 1*, 79-108.

Orsi, A. (1988). Coherencia de razonamiento y diferencias individuales: Un estudio de sus características e interdependencias [Reasoning coherence and individual differences: A characteristics and interdependences study]. *Revista Signos Universitarios, Psicología,* Año 7, *3*, 69-102.

Prolog Development Center (1997). *Lenguaje tutorial* [Tutorial language]. Copenhagen, Denmark.

Spivey, M. (1996). *An introduction to logic programming through Prolog*. London: Prentice Hall.

*Instituto de Investigaciones Psicológicas*
*de la Universidad del Salvador (IIPUS)*
*Marcelo T. de Alvear 1326*
*1er Piso*
*(C1058AAV) Buenos Aires – Argentina*

Received: April 16, 2001
Accepted: April 30, 2001