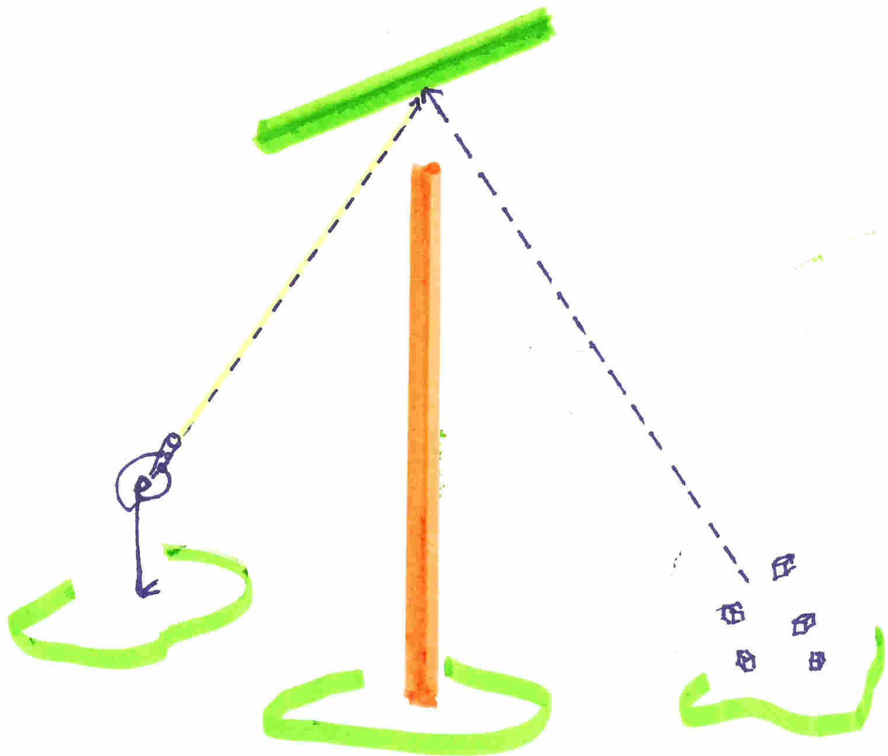
Workshop in A. I. and C. S. in honour of Jacques Calmet

A LOGIC INSPIRED BY RELATIVISTIC AND QUANTUM MECHANICS

Luis M. Laita



p					**E**
	1, 2	2, 2	0, 2	1, 2	2, 2
	1, 1	2, 1	0, 1	1, 1	2, 1
	1, 0	2, 0	0, 0	1, 0	2, 0
	1, 2	2, 2	0. 2	1, 2	2, 2
q	1, 1	2, 1	0, 1	1, 1	2, 1 **t**

DOUBLE LOGIC TRUTH TABLES

- Lukasiewicz's Standard 3-valued and modal logic (N. Rescher, page 23)

¬		□		◊		∧	0	1	2	∨	0	1	2	→	0	1	2	≈	0	1	2
0	2	0	0	0	0	0	0	0	0	0	0	1	2	0	2	2	2	0	2	1	0
1	1	1	0	1	2	1	0	1	1	1	1	1	2	1	1	2	2	1	1	2	1
2	0	2	2	2	2	2	0	1	2	2	2	2	2	2	0	1	2	2	0	1	2

- Changed logic

¬c		□ c		◊c		∧c	0	1	2	∨c	0	1	2	→c	0	1	2	≈c	0	1	2
0	1	0	0	0	0	0	0	0	0	0	0	2	1	0	1	1	1	0	1	2	0
1	2	1	0	1	1	1	0	2	2	1	2	2	1	1	2	1	1	1	2	1	2
2	0	2	1	2	1	2	0	2	1	2	1	1	1	2	0	2	1	2	0	2	1

- The truth values of changed logic comes from multiplying by 2 the truth values of standard logic. Justified by the function $v(p) = (\Box p \wedge 1) \vee \Diamond(p \wedge \neg p)$, which sends 0, 1 and 2 to 0, 2 and 1.

Polynomial translation of the basic logic formulae

- **Standard logic**

¬X is translated into the polynomial: $2+2x$

◇ X is translated into the polynomial $2x^2$

□X is translated into the polynomial x^2+2x

X∨Y is translated into the polynomial $x^2y^2+x^2y+xy^2+2xy+x+y$

X∧Y is translated into the polynomial $2x^2y^2+2x^2y+2xy^2+xy$

X→Y is translated into the polynomial $2x^2y^2+2x^2y+2xy^2+xy+2x+2$

X↔Y is translated into the polynomial $x^2y^2+x^2y+xy^2+2xy+2x+2y+2$

- **Changed logic**

¬X is translated into the polynomial $1+x$

◇ X is translated into the polynomial x^2

□X is translated into the polynomial $2x^2+x$

X∨Y is translated into the polynomial $2x^2y^2+2x^2y+2xy^2+xy+2x+2y$

X∧Y is translated into the polynomial $x^2y^2+x^2y+xy^2+2xy$

X→Y is translated into the polynomial $x^2y^2+x^2y+xy^2+2xy+x+1$

X↔Y is translated into the polynomial $2x^2y^2+2x^2y+2xy^2+xy+x+y+1$

Tautologies,undefinologies, contradictions and tautological and undefinological consequences

- **“Tautologies" are the standard logical formulae such that their truth-value is 2 for any assignation of truth-values to its propositional variables. An example of tautology in standard logic.**

◇(x[1]∨¬x[1])):

the output for NF(POS(OR1(x[1],NEG(x[1]))),I); is -1 **(that is 2), a tautology,**

- **"Undefinologies", are the changed logic formulae such that their truth-value is 1 for any assignation of truth-values to its propositional variables. An example of undefinology in changed logic is:**

◇c(x[1] ∨c ¬c x[1])) .

The output for: NF(POSC(OR1C(x[1],NEGC(x[1]))),I); is 1, an undefinology.

- **"Contradictions" are the standard/changed logic formulae such that their truth-values are 0 for any assignation of truth-values to its propositional variables. Examples of contradictions are:**

Standard logic contradiction: ◻(x[1]∧¬x[1]).

The output for NF(NEC(AND1(x[1],NEG(x[1]))),I); is 0, a standard logic contradiction.

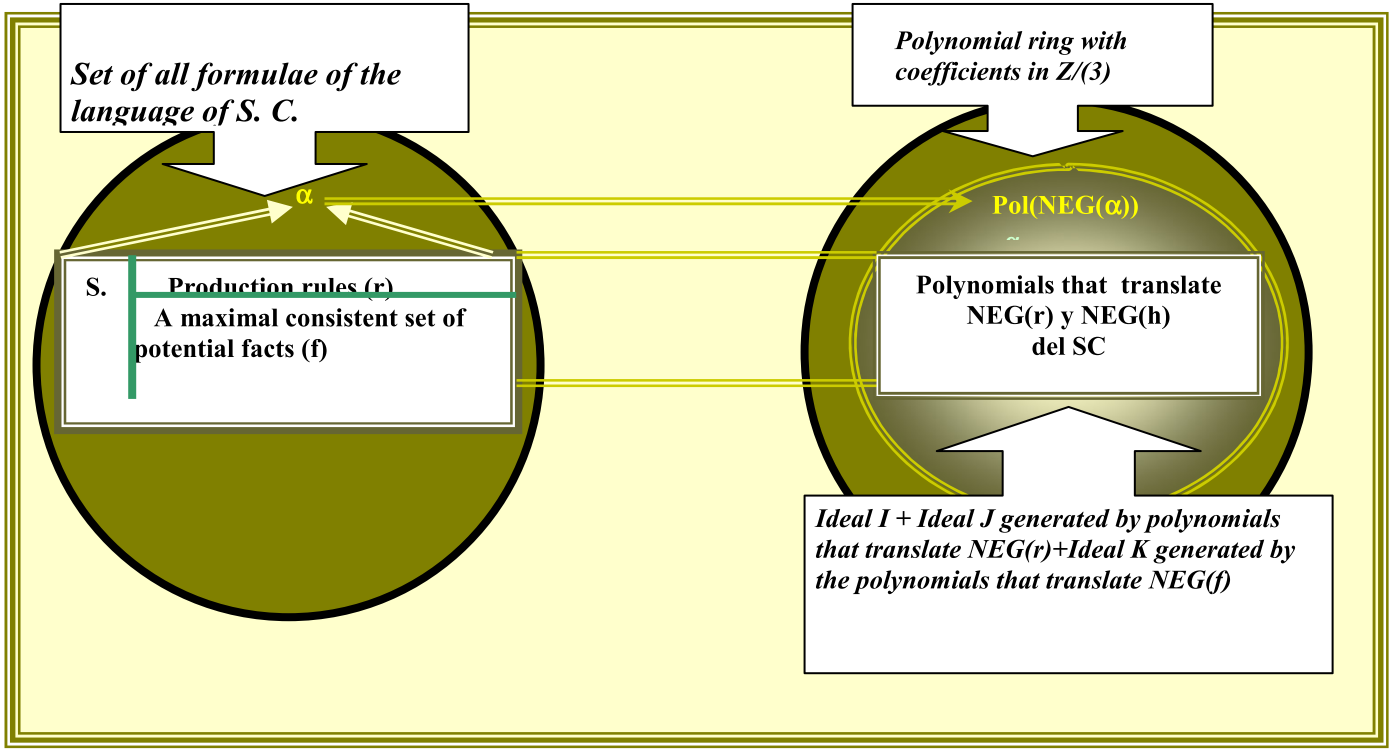
Changed logic contradiction:

◻c ¬c (x[1] ∧c ¬c x[1])).

The output for NF(NECC(NEGC(AND1C(x[1],NEGC(x[1])))),I); is **0**, a changed logic contradiction.

- Tautological, undefinological consequence. In standard (respectively changed) logic, a formula α (respectively δ) is tautological (respectively undefinological), consequence of a set Γ (respectively Θ) of formulae, whenever in all cases in which, the formulae in Γ have truth value 2 (respectively 1), α has truth-value 2 (respectively 1). That α (respectively δ) is tautological (respectively undefinological) consequence of the set $\Gamma=\{\beta_1, \beta_2, \ldots, \beta_n\}$ (respectively $\Theta=\{\gamma_1, \gamma_2, \ldots, \gamma_m\}$) is denoted as: $\beta_1, \beta_2, \ldots, \beta_n \models \alpha$ (respectively $\gamma_1, \gamma_2, \ldots, \gamma_m \models_c \delta$).

The main theorem, relating tautological-consequences with an ideal membership problem (standard logic):



PART COMMON TO ALL CONSISTENCY CHECKINGS (STANDARD AND CHANGED LOGICS)

Definition of the ring A and of the basic ideal I

```
A ::= Z/(3)[x[1..3],a[1..3]];

I:=Ideal(x[1]^3-x[1], x[2]^3-x[2], x[3]^3-x[3], a[1]^3-a[1],
         a[2]^3-a[2], a[3]^3-a[3]);
```

- **CoCoA commands for the translations of the polynomials corresponding to the basic connectives FOR STANDARD LOGIC AND CHANGED LOGIC**

  * **Standard logic**

```
NEG(M):=NF(2-M,I);
POS(M):=NF(2*M^2,I);
NEC(M):=NF(M^2+2*M,I);
OR1(M,N):=NF(M^2*N^2+M^2*N+M*N^2+2*M*N+M+N,I);
AND1(M,N):=NF(2*M^2*N^2+2*M^2*N+2*M*N^2+M*N,I);
IMP(M,N):=NF(2*M^2*N^2+2*M^2*N+2*M*N^2+M*N+2*M+2,I);
IFF(M,N):=NF(M^2*N^2+M^2*N+M*N^2+2*M*N+2*M+2*N+2,I);
```

  * **Changed logic**

```
NEGC(M):=NF(1+M,I);
POSC(M):=NF(M^2,I);
NECC(M):=NF(2*M^2+M,I);
OR1C(M,N):=NF(2*M^2*N^2+2*M^2*N+2*M*N^2+M*N+2*M+2*N,I);
AND1C(M,N):=NF(M^2*N^2+M^2*N+M*N^2+2*M*N,I);
IMPC(M,N):=NF(M^2*N^2+M^2*N+M*N^2+2*M*N+M+1,I);
IFFC(M,N):=NF(2*M^2*N^2+2*M^2*N+2*M*N^2+M*N+M+N+1,I);
```

- **Program CONSIST**

```
Define CONSIST(J_,N_)
   N_:=N_-1;
   GB_:=[0];
   While N_ < Len(J_) And Not GB_=[1] Do
      N_:=N_+1;
      GB_:=GBasis(Ideal( First(J_,N_) ));
      If GB_=[1]
         Then Print 'Hasta el elemento '; Print N_; Print 'hay';
              PrintLn 'INconsistencia'
         Else Print 'Hasta el elemento '; Print N_; Print 'hay';
              PrintLn 'CONsistencia'
         End;
   End;
End;
```

STANDARD LOGIC CONSISTENCY CHECKING

Production rules and potential facts

```
R1:=NF(IMP(AND1(x[1],NEG(x[2])),OR1(NEC(a[1]),POS(NEG(a[2])))),I);
R2:=NF(IMP(x[1],NEC(NEG(a[1]))),I);
R3:=NF(IMP(AND1(AND1(x[1],NEG(x[2])),x[3]),AND1(NEC(a[2]), POS(a[3]))),I);

F1:=x[1];  F1N:= NEG(x[1]);
F2:=x[2];  F2N:= NEG(x[2]);
F3:=x[3];  F3N:= NEG(x[3]);
```

Definitions of the ideals for production rules and potential facts F1, F2N and F3

```
J:=Ideal(NEG(R1),NEG(R2),NEG(R3));

K:=Ideal(NEG(F1),NEG(F2N),NEG(F3));
```

Consistency checking

Consistency checking using Gröbner bases

--Input

```
GBasis(I+K+J);
```

--Output

```
-------------------------------
[1]
-------------------------------
```

Consistency checking element by element with program CONSIST

--Input.

```
CONSIST(List(I+K+J),1);
```

--Output

```
-------------------------------
Up to element 1 there is CONsistency
Up to element 2 there is CONsistency
.....................
Up to element 11 there is CONsistency
Up to element 12 there is INconsistency
```

****

6 elements in ideal I (basic ideal), 3 in ideal K (generated by the subset {F1, F2N, F3}, 3 in ideal J (production rules). **Thus, element 12 (RULE 3) produces inconsistency.**

CONTRADICTIONS IN CHANGED LOGIC

- **Changed logic production rules and potential facts**

```
R1C:=NF(IMPC(AND1C(x[1],NEGC(x[2])),OR1C(NECC(a[1]),POSC(NEGC(a[2])))),I);
R2C:=NF(IMPC(x[1],NECC(NEGC(a[1]))),I);
R3C:=NF(IMPC(AND1C(AND1C(x[1],NEGC(x[2])),x[3]),AND1C(NECC(a[2]),
POSC(a[3]))),I);

F1C:=x[1];  F1CN:= NEGC(x[1]);
F2C:=x[2];  F2CN:= NEGC(x[2]);
F3C:=x[3];  F3CN:= NEGC(x[3]);
```

- **Definitions of the ideals for production rules and potential facts**

Production rules R1C. R2C, R3C

```
JC:=Ideal(NEGC(R1C), NEGC(R2C),NEGC (R3C));
```

Ideal for potential facts F1C, F2CN, F3C

```
KC:=Ideal(NEGC(F1C),NEGC(F2CN),NEGC(F3C));
```

Consistency checking using Gröbner bases

--Input.

```
GBasis(I+KC+JC);
GBasis(I+JC+KC);
```

--Output

```
-------------------------------
[1]
-------------------------------
[1]
-------------------------------
```

Consistency checking element by element with program CONSIST

--Input 1

```
CONSIST(List(I+KC+JC),1);
```

--Output 1.

```
Up to element 1 there is CONsistency
Up to element 2 there is CONsistency
..................
Up to element 11 there is CONsistency
Up to element 12 there is INconsistency
```

--Input 2

```
CONSIST(List(I+JC+KC),1);
```

--Output 2

```
Up to element 1 there is CONsistency
..................
Up to element 8 there is CONsistency
Up to element 9 there is INconsistency
```

--Input 3

```
CONSIST(List(I+JC),1);
```

--Output 3

Up to element 1 there is CONsistency

…………..

Up to element 8 there is CONsistency

Up to element **9 there is IN consistency**

RULE 3 IS THE FORMULA THAT PRODUCES INCONSISTENCY, AS FOR STANDARD LOGIC

Another approach to comparison of inconsistency between standard and changed logic

- Apply POS to the three changed logic formulae R1C, R2C and R3C. It can be checked that POS applied to changed logic connectives = POS applied to the corresponding standard logic connectives: the result is the following one.

```
R1:=NF(POS(IMP(POS(AND1(x[1],POS(NEG(x[2])))),POS(OR1(POS(NEC(a[1])),
       POS(POS(NEG(a[2]))))))),I);
R2:=NF(POS(IMP(x[1],POS(NEC(POS(NEG(a[1])))))),I);
R3:=NF(POS(IMP(POS(AND1(POS(AND1(x[1], POS(NEG(x[2])))),x[3])),
       POS(AND1(POS(NEC(a[2])), POS(POS(a[3])))))),I);
```

That is, POS has sent changed logic to standard logic.

- Similarly POSC sends standard logic connectives into the corresponding changed logic connectives, being the result the following one.

```
R1C:=NF(POSC(IMPC(POSC(AND1C(x[1],POSC(NEGC(x[2])))),
        POSC(OR1C(POSC(NECC(a[1])),POSC(POSC(NEGC(a[2]))))))),I);
R2C:=NF(POSC(IMPC(x[1],POSC(NECC(POSC(NEGC(a[1])))))),I);
R3C:=NF(POSC(IMPC(POSC(AND1C(POSC(AND1C(x[1],POSC(NEGC(x[2])))),x[3])),
        POSC(AND1C(POSC(NECC(a[2])), POSC(POSC(a[3])))))),I);
```

That is, POSC has sent standard logic to changed logic.

System R1, R2 and R3.

- **Potential facts for STANDARD logic**

```
F1:=x[1];  F1N:= NEG(x[1]);
F2:=x[2];  F2N:= NEG(x[2]);
F3:=x[3];  F3N:= NEG(x[3]);
```

- **Ideals for Production rules and potential facts**

```
J:=Ideal(NEG(R1),NEG(R2),NEG(R3))
K:=Ideal(NEG(F1),NEG(F2N),NEG(F3))
```

- **Consistency checking with Gröbner bases**

--Inputs

```
GBasis(I+K+J);
```

--Outputs

```
-------------------------------
[1]   Inconsistency
-------------------------------
```

- **Second method, Consistency element by element**

  --Input

```
CONSIST(List(I+K+J),1);
```

  --Output

```
-------------------------------
Up to element 1 there is CONsistency
...............
Up to element 11 there is CONsistency
Up to element 12 there is INconsistency
```

System R1C, R2C and R3C.

- **Potential facts for changed logic**

```
F1C:=x[1];  F1CN:= NEGC(x[1]);
F2C:=x[2];  F2CN:= NEGC(x[2]);
F3C:=x[3];  F3CN:= NEGC(x[3]);
```

- **Definitions of the ideals for production rules and potential facts**

```
JC:=Ideal(NEGC(R1C), NEGC(R2C),NEGC (R3C);
KC:=Ideal(NEGC(F1C),NEGC(F2CN),NEGC(F3C));
```

- **Consistency checking with Gröbner bases.**

```
GBasis(I+KC+JC);
-------------------------------
[1]  Inconsistency
-------------------------------
```

- **Second method, Consistency element by element.**

--Input

```
CONSIST(List(I+KC+JC),1);
```

--Output

```
--------------------------------
Up to element 1 there is CONsistency
```

....

```
Up to element 11 there is CONsistency
Up to element 12 there is Inconsistency
```

Undefinological consequences translated into tautological consequences by POS

Ring A, main ideal I and commands for basic connectives, as always

- Consider the undefinological consequence:

  N1, N2, N4 ⊧ c N3

  Where N1 to N4 are the next formulae

```
N1:= NF(POSC(x[1]),I);
N2:= NF(NEGC(x[2]),I);
N3:= NF(POSC(x[3]),I);
N4:= NF(IMPC(AND1C(N1,N2),N3), I);
```

Consider the ideal generated by the negation NEGC in changed logic of N1, N2 and N4

```
I5:= Ideal(NEGC(N1), NEGC(N2), NEGC(N4));
```

The output is the ideal:

```
Ideal(x[1]^2 + 1, x[2] - 1,  - x[1]^2x[2]^2x[3]^2 + x[1]^2x[2]x[3]^2 - x[1]^2x[2]^2 -
      x[1]^2x[3]^2 + x[1]^2x[2] - x[1]^2 - 1)
```

Consider the ideal I6 that is defined as I5 enlarged with NEG(N3)

```
I6:= Ideal(NEGC(N1), NEGC(N2), NEGC(N4), NEGC(N3));
```

The output is the ideal

```
Ideal(x[1]^2 + 1, x[2] - 1,  - x[1]^2x[2]^2x[3]^2 + x[1]^2x[2]x[3]^2 - x[1]^2x[2]^2 -
      x[1]^2x[3]^2 + x[1]^2x[2] - x[1]^2 - 1, x[3]^2 + 1)
```

IMPORTANT RESULT

Gbasis (I + I5) = Gbasis (I + I6) = [1]

This means that the ideals generated by the NEGC of antecedents of the UNDEFINOLOGICAL CONSEQUENCES and the one enlarged with the NEGC of the consequent coincide with the whole ring A. THE MAIN THEOREM DOESN'T HOLD, AND THE MEANING OF CONTRADICTION CHANGES IN CHANGED LOGIC.

- LET US EXCHANGE THE CHANGED LOGIC UNDEFINOLOGICAL CONSEQUENCE TO TAUTOLOGICAL CONSEQUENCE BY APPLYING POS.

```
P1:= NF(POS(x[1]),I);
P2:= NF(POS(NEG(x[2])),I);
P3:= NF(POS(x[3]),I);
P4:= NF(POS(IMP(POS(AND1(P1,P2)),P3)),I);
```

Ideals J5 and J6

```
J5:= Ideal(NEG(P1), NEG(P2), NEG(P4));
```

Output

```
Ideal(x[1]^2 - 1, x[2]^2 - x[2], x[1]^2x[2]^2x[3]^2 - x[1]^2x[2]x[3]^2 - x[1]^2x[2]^2
      + x[1]^2x[3]^2 + x[1]^2x[2] - x[1]^
```

```
J6:= Ideal(NEG(P1),NEG(P2),NEG(P4),NEG(P3));
```

Output

```
Ideal(x[1]^2 - 1, x[2]^2 - x[2], x[1]^2x[2]^2x[3]^2 - x[1]^2x[2]x[3]^2 - x[1]^2x[2]^2
      + x[1]^2x[3]^2 + x[1]^2x[2] - x[1]^2, x[3]^2 - 1)
```

IMPORTANT RESULTS

```
GBasis(I+J5) = GBasis(I+J6) = [x[1]^2 - 1, x[2]^2 - x[2], x[3]^2 - 1]
```

Normal form in standard logic:

```
NF(NEG(P3), I + J5) = 0
```

- **Thus, when translating undefinological consequences into standard logic, although the ideals in changed logic coincide with the whole ring, the main theorem holds for the tautological consequence which is the standard logic translation of the mentioned undefinological consequence.**

CONCLUSSIONS

- **"CONTRADICTIONS" IN CHANGED LOGIC STAND AT THE SAME PLACES (IN AN EXPERT SYSTEM) AS FOR STANDARD LOGIC, BUT THE MEANING OF CHANGED LOGIC CONTRADICTION IS DIFFERENT.**

- **NEVERTHELESS, IF TRANSLATING (BY "POS") THE CHANGED LOGIC FORMULAE INTO STANDARD LOGIC FORMULAE, THE MAIN THEOREM APPLIES.**

- **BY CONTRAPOSITION, IF THE ORIGINAL STANDARD LOGIC FORMULAE ARE MOVED TO CHANGED LOGIC BY "POS", THE CORRESPONDING "CONTRADICTION" APPEARS IN CHANGED LOGIC.**