Miguel Sánchez-Mazas

IDEAS PRELIMINARES SOBRE UN CALCULO AUTOMATICO DE LAS "CUALIDADES"

1. *Perspectivas y alcance de un cálculo puramente "cualitativo"*

Es posible concebir un tipo de cálculo automático cuyos últimos argumentos —constantes o variables— ya no representen cantidades, sino cualidades.

Hay que entender aquí 'cualidad' ante todo en oposición a 'cantidad', así como a 'relación', como un modo de ser que puede ser afirmado o negado de un sujeto,[1] el cual puede, a su vez, ser una cualidad.

---

1. Nuestro concepto de 'cualidad' se aproxima al de 'propiedad' ('property'), utilizado por Carnap (Véase Rudolf Carnap: *Meaning and necessity, A study in semantics and modal logic.* Chicago, University of Chicago Press, 1947, en especial las secciones: 4. Classes and Properties, y 5. Extensions and Intensions), siempre que no se olvide que nosotros incluimos también, entre las cualidades, como casos límite, por un lado, a las "cualidades individuales" —colmando, por consiguiente, el abismo escolástico entre la *species specialissima* y el individuo y aceptando prácticamente el "principio de los indiscernibles" de Leibniz, en virtud del cual no es admisible la idea de dos individuos distintos de cualidad idéntica— y, por otro, a la "cualidad universal" y a la "cualidad contradictoria". Véanse también el artículo "Qualité" en el *Vocabulaire technique et critique de la Philosophie,* ed. André Lalande, sexta edición, Paris, Presses Universitaires de France, 1951, y el artículo "Cualidad" en el *Diccionario de Filosofía,* ed. José Ferrater Mora, quinta edición, Buenos Aires, Editorial Sudamericana, 1965.

Mas precisamente, llamaremos 'cualidad' a la conjunción de todos los caracteres y propiedades aplicables a todos los individuos de una clase dada —lo que se denomina la comprehensión o intensión de dicha clase—, evocada por cualquier término que pueda figurar como sujeto o predicado de una proposición cualquiera.

En la práctica, se aceptará como expresión válida de una cualidad, en el contexto de una lengua natural, cualquier adjetivo calificativo (o proposición adjetiva de valor equivalente, como las utilizadas por Boole para definir a ciertas clases), así como cualquier nombre, ya sea común o propio. El nombre propio que designa a un individuo u objeto individual será admitido como caso límite para la expresión de las cualidades, siendo su comprehensión una 'cualidad individual'.

No es necesario que insistamos aquí sobre otros medios de expresión de las cualidades, tanto en las lenguas naturales como en los lenguajes artificiales, y ya tratándose de toda la esfera lingüística, ya de una esfera o rama científica particular. Nos limitaremos a precisar en lo que sigue las formas de expresión algebraica o numérica de las cualidades en un lenguaje de programación (el FORTRAN).

En cierta medida, también los *descriptores*, empleados en algunos intentos de tratamiento automático de la documentación bibliográfica y recuperación de la información,[2] podrían, a nuestro juicio, quedar reducidos a las cualidades y tratarse en consecuencia.

Las dificultades de todo cálculo lógico basado en el aspecto puramente *cualitativo* de los términos —es decir, en su comprehensión— son bien conocidas desde el comienzo mismo de la lógica matemática con Leibniz hasta nuestros días, pasando por De Morgan y Boole, cuya *álgebra lógica*, construida

2. En relación con estos intentos, véanse, por ejemplo, los estudios sobre tratamiento automático de la documentación sobre problemas económicos, sociales, laborales y sindicales que viene realizando en Ginebra desde hace algunos años, utilizando "descriptores" y basando en ellos la orientación bibliográfica, la Oficina Internacional del Trabajo (Servicio I.S.I.S. —Integrated Scientific Information Service—). La última lista de "descriptores" para la esfera indicada ha sido publicada en agosto de 1966.

a partir de las relaciones extensionales entre las clases, se halla hoy en la base del funcionamiento de los ordenadores.

Estas dificultades, sin embargo, no nos han parecido nunca —y así lo hemos venido afirmando desde hace bastantes años en distintos trabajos—[3] esenciales y definitivas. De lo que se trata es, simplemente, de superarlas, y para lograrlo hemos constatado, ante todo, que es indispensable situarse en una perspectiva formal muy diferente a la de la extensión, abandonando la falsa idea de que una y otra (la perspectiva de la intensión y la de la extensión) son simétricas y, en el fondo, equivalentes, sobre todo en lo relativo a la negación y a la exclusión, cuyo sentido, en comprehensión (la *incompatibilidad*) es algo enteramente nuevo, con propiedades muy peculiares.

Nuestro modo de abordar el problema se basa, en la práctica, en la consideración de dos tipos de *memorias* diferentes y estrechamente vinculadas entre sí: la *memoria de las cualidades* y *la memoria de las proposiciones*. Una y otra pueden ser concebidas de un modo abstracto o realizadas en el ordenador.

La primera memoria permite efectuar una enumeración y clasificación inicial, siempre perfectible, de las categorías, conceptos y objetos de una ciencia o esfera particular —sus *cualidades* específicas— y definir ciertas *funciones* de esas cualidades. En la segunda memoria quedan fijadas y son tratadas las relaciones que se dirán 'aceptadas', 'admitidas' o 'afirmadas' —en forma de proposiciones válidas en el sistema— entre las cualidades y funciones citadas, deduciéndose las eventuales consecuencias.

Una vez establecido este marco general, nos hallamos aún, como es natural, bastante lejos de haber previsto el modo de resolver todos los problemas planteados por la definición y la realización automática de los distintos tipos de *funciones de cualidad* necesarios para describir la estructura de una es-

3. Véase, en especial, nuestro libro: *Fundamentos matemáticos de la Lógica Formal*. Caracas, Universidad Central de Venezuela, 1963 (Premio "Menéndez y Pelayo" del Consejo Superior de Investigaciones Científicas de España).

fera particular de conceptos, así como de los criterios de deducción válidos en dicha esfera. Algunos de estos problemas se hallan, por otra parte, estrechamente vinculados a los eventuales resultados de la investigación lingüística en curso sobre las funciones y las estructuras sintácticas, a través de las gramáticas generativas y transformacionales, en la medida en que tales resultados puedan facilitar la construcción de un lenguaje artificial, intermediario universal entre las lenguas naturales,[4] con un grado de formalización bastante elevado para que sea posible extraer del mismo una expresión susceptible de tratamiento automático de todas las funciones de cualidad presentes en cada esfera.

Con todo, en nuestro trabajo demostramos ya, de un modo efectivo, la posibilidad de empezar a llenar este marco vacío, de una amplitud y complejidad impresionantes, con un tratamiento automático riguroso de algunos tipos elementales de funciones lógicas y de proposiciones, desde el ángulo cualitativo, comprehensivo (o, si se prefiere, intensional).

Como ejemplo de la validez de nuestro enfoque, demostramos cómo puede fundamentarse en un tipo de cálculo extraordinariamente claro y sencillo la programación de la deducción automática, en el ordenador, de los 24 modos válidos (incluyendo los de conclusión atenuada) del silogismo categórico, cuya complejidad es bien conocida. El método utilizado puede abrir el camino a otros intentos de simplificación análoga de distintos tipos de deducción.

Por lo que sabemos, es la primera vez, desde que la silogística fue establecida por Aristóteles hace 23 siglos, que se ha logrado calcular, de modo estrictamente algebraico, todos los modos del silogismo, obteniéndose el *número de la conclusión* —o, si se quiere, el número que, en virtud de la correspondencia inicial en que se funda la aritmetización, iden-

4. En relación con el problema de un lenguaje intermediario, especialmente como instrumento auxiliar para la traducción automática, consúltese, por ejemplo, el estudio de N. D. Andreyev: "The intermediary language as the focal point of machine translation", publicado con otros trabajos sobre la traducción automática en el volumen *Machine translation*, de A. D. Booth, Amsterdam, North-Holland Publishing Company, 1967.

tifica por completo el tipo y los términos componentes de la proposición que es legítimo deducir de las premisas—, en forma de *producto aritmético* de los *números de las premisas.*

Durante toda su vida —como es bien sabido—, Leibniz habían intentado, a través de una sucesión increíble de sistemas diferentes, proceder de esa forma, sin conseguirlos jamás.[5] Los críticos más eminentes de sus concepciones lógicas —y, a la cabeza de ellos, Couturat—[6] habían sentenciado como definitivamente condenados al fracaso todos los eventuales intentos en esa dirección, en especial cuando éstos pretendan apoyarse en la perspectiva que llamamos 'cualitativa', en oposición a la de la extensión, esencialmente cuantitativa; y esto debido sobre todo a las dificultades *insuperables* (para los citados críticos) con que tropieza la formalización de la versión comprehensiva de la *exclusión* (es decir, la *incompatibilidad*, "esa especie de repulsión moral" que desconcertaba a Couturat).

Asistimos pues, una vez más, al hundimiento de un prejuicio. Las consecuencias teóricas de este hecho —para la lógica, el álgebra y el tratamiento automático de la información— y hasta las consecuencias, por así decirlo, pedagógicas, son a nuestro juicio, innegables.

2. *La memoria de las cualidades*

En esta memoria podrá introducirse —por los procedimientos definidos a continuación— las *cualidades elementales* pertenecientes a un sistema de cualidades elegido para la descripción de una esfera determinada (álgebra, geometría, electrónica, genética, psicología, sociología...) y definirse los *operadores de cualidad* y las *funciones de cualidad aplicables* ya a las cualidades, en general, ya a las cualidades específicas de la esfera considerada.

---

5. La historia dramática de estos repetidos ensayos está registrada principalmente en los "*Opuscules et fragments inédits de Leibniz*", *extraits des manuscrits de la Bibliothèque royale de Hanovre*, por Louis Couturat. París, 1903. Nuestra obra citada en la nota 3 contiene un análisis sistemático de los más importantes de dichos ensayos.

6. Véase Louis Couturat: *La Logique de Leibniz d'après des documents inédits*. París, Félix Alcan, 1901, y también L. Couturat - L. Leau: *Histoire de la Langue Universelle*. París, Hachette, 1903.

2.1. *Cualidades elementales*

En la fase que precede a su introducción en la memoria del ordenador, las cualidades elementales escogidas para la descripción de una esfera determinada quedarán expresadas mediante términos de una lengua natural (francés, inglés, español...) o, en ciertos casos, mediante grupos de términos, acompañados o no de sus respectivas definiciones o restricciones relativas a su empleo en la esfera considerada, así como de sus eventuales equivalencias en otras lenguas. Podrán también quedar expresadas por fórmulas químicas o por expresiones de cualquier otra lengua artificial.

Nombres como 'poliedro', 'partícula', 'planeta', 'polinomio', 'matriz', 'organización', 'virus', 'figura', 'imagen', 'hidrógeno'; adjetivos como 'regular', 'cargado(a)', 'rectangular', 'mundial', 'sindical', 'social', 'vegetal', 'artístico(a)', 'metálico(a)', 'líquido(a)', y hasta nombres propios como 'España', 'Venus', 'O.N.U.', acompañados o no de definiciones o de restricciones, podrán ser utilizados normalmente como medio válido de expresión de las cualidades elementales, antes de su introducción en la memoria, y al margen del tratamiento automático de que puedan ser objeto en la misma.

En ésta, las cualidades elementales serán introducidas ya en forma de variables, ya en forma de constantes, mediante una *declaración* del tipo de las declaraciones utilizadas en lenguaje FORTRAN para notificar el tipo de variable evocado por cada identificador de variable y, como consecuencia, el tipo de elemento de memoria (entero, real, complejo, lógico de doble precisión...) que se asigna a dicho identificador. En nuestro caso, la mencionada declaración adoptará una de las dos formas siguientes:

a) La expresión 'QUAL' seguida de una lista de identificadores de variable, separados por comas, cada uno de los cuales estará formado por 1 hasta 6 caracteres alfabéticos FORTRAN, siendo necesariamente estos caracteres una de las letras A, B, C,...M;

b) La expresión 'QUAL' seguida de una lista de números primos, separados por comas.

Una declaración de tipo a) especifica que a los identificadores de variable que ella enumera quedan asociadas durante todo el programa *variables de cualidad*, es decir, variables que no pueden adoptar otros valores que números primos expresivos de cualidades determinadas, de acuerdo con una asignación convenida. Una declaración de tipo b) especifica que los números primos que ella enumera expresan durante todo el programa cualidades determinadas, pertenecientes al sistema elegido para describir una esfera determinada.

He aquí ejemplos de las declaraciones de los tipos a) y b):

a) QUAL A, B, C, ALMA, CECA, AELE

b) QUAL 3, 5, 13, 43, 191, 257, 347

En cualquier momento de un programa abierto por una declaración del tipo a), que enumera identificadores —pongamos como ejemplo—, A, B, JL, KC..., pueden figurar fórmulas aritméticas FORTRAN de la forma elemental 'A-5', 'B-13', 'JL-43', 'KC-347'..., que indican que los números primos 5, 13, 43, 347..., que expresan cualidades determinadas, son atribuidos, respectivamente, a las variables A, B, JL, KC..., como valores de las mismas, y colocados en los elementos de memoria correspondientes.

2.2. *Las cualidades límite: la cualidad universal U y la cualidad contradictoria Z*

Además de las cualidades elementales ya definidas, cada sistema de cualidades incluye las dos *cualidades límite:* U (*cualidad universal*)[7] y Z (*cualidad contradictoria*),[8] a las

7. A esta "cualidad universal" corresponde, en la perspectiva extensional, el "Universo del Discurso" de Boole.
8. A esta "cualidad contradictoria" corresponde, en el cálculo extensional de las clases, la "clase vacía" (*null class*).

que quedan asignados siempre, respectivamente, los valores numéricos siguientes:

$$U = 1 \qquad Z = -1$$

2.3. *Los operadores de cualidad*

En cada sistema de cualidades es posible introducir *operadores* cuyos argumentos serán cualidades elementales. La aplicación de estos operadores a una, dos,... n cualidades elementales permite la construcción de *funciones de cualidad* monádicas, diádicas,...n-ádicas. Cada operador de cualidad quedará definido, a los efectos del cálculo automático, por la operación que el mismo efectúa sobre constantes (números primos) o variables que expresan cualidades elementales.

2.3.1. *Operadores monádicos: el operador de inversión negativa*

Como ejemplo de operador monádico (u operador cuyo argumento es una única cualidad elemental), definiremos aquí el operador de *inversión negativa* '–1/' (en el cual el espacio en blanco que sigue al signo '/' puede ser ocupado por la expresión de cualquier cualidad elemental): en el cálculo automático, este operador efectúa la operación aritmética que, aplicada a la constante o variable que expresa la cualidad dada (su argumento), produce como resultado el valor inverso o recíproco, cambiado de signo, de ese argumento. El operador está escrito en caracteres FORTRAN.

2.3.2. *Operadores diádicos: el operador de composición*

Como ejemplo de operador diádico (u operador cuyos argumentos son dos cualidades elementales distintas), definiremos aquí el operador de *composición*'*' (en el cual cada uno de los espacios en blanco —el que precede y el que sigue, respectivamente, al signo '*'— puede ser ocupado por la expresión de una cualidad elemental cualquiera): en el cálculo automático, el operador de composición tiene el sentido aritmético de la multiplicación, es decir, que efectúa la operación aritmética cuya aplicación a las dos constantes o a las dos variables que expresan las cualidades dadas (entre sí distintas), produce como resultado el producto aritmético de esos dos argumentos. Escribimos dicho operador utilizando el mismo carácter empleado en FORTRAN para designar la multiplicación aritmética (el asterisco '*').

2.4. *Las funciones de cualidad*

Una función de cualidad es una función —en el sentido matemático general— cuyos argumentos son cualidades elementales. Quedará expresada, en el cálculo automático, por una expresión algebraica que tomará un valor aritmético para cada juego posible de valores aritméticos de sus argumentos. En el caso de las funciones que definimos a continuación, los valores que puede adoptar cada función son siempre números enteros o fraccionarios, positivos o negativos, y, por consiguiente, valores que pueden ser colocados en una *memoria real* del ordenador y tratados por éste según el *modo real.*

Toda función de cualidad será definida y construida utilizando ya operadores de cualidad, ya otras funciones de cualidad anteriormente definidas, ya operadores y funciones a la vez.

2.4.1. *Funciones monádicas: la negación de una cualidad A*

La *negación de una cualidad elemental A* será definida y construida mediante la aplicación del operador de inversión negativa '−1/' a la cualidad en cuestión A. Escribiremos:

Para toda cualidad elemental A: N(A) = −1/A (definición)

2.4.2. *Funciones de varios argumentos*

En cualquier sistema de cualidades elementales, se podrán definir funciones de cualidad de 2, 3, 4,..n argumentos. Consideramos aquí, ante todo, las *funciones diádicas* o de dos argumentos, y como ejemplo de estas funciones definiremos la *compatibilidad*, la *incompatibilidad*, la *independencia* y la *dependencia* de dos cualidades A, B (tomadas en este orden).

2.4.2.1. *La compatibilidad de dos cualidades A, B*

Para todo par de cualidades elementales A, B:

COMP(A, B) = A*B (definición)

2.4.2.2. *La incompatibilidad de dos cualidades A, B*

Para todo par de cualidades elementales A, B:

INCOMP(A, B) = N(COMP(A, B)) = N(A*B) =
= –1/(A*B) (definición)

2.4.2.3. *La independencia de una cualidad A, respecto de otra cualidad B*

Para todo par de cualidades elementales A, B:

INDEP(A, B) = COMP(A, N(B)) = A*(–1/B) =
= –A/B (definición)

2.4.2.4. *La dependencia de una cualidad A, respecto de otra cualidad B*

Para todo par de cualidades elementales A, B:

DEP(A, B) = N(INDEP(A, B)) = N(COMP(A, N(B))) =
N(A*(–1/B)) = N(–A/B)
= –1/(–A/B) = B/A (definición)

2.4.3. *Algunas propiedades de las funciones de cualidad antes definidas*

a) La compatibilidad y la incompatibilidad son funciones conmutativas. Esto significa que las igualdades siguientes:

COMP(A, B) = COMP(B, A) e INCOMP(A, B) =
= INCOMP(B, A)

son siempre válidas;

b) La independencia y la dependencia son funciones no-conmutativas. Esto significa que las igualdades siguientes:

INDEP(A, B) = INDEP(B, A) y DEP(A, B) = =DEP(B, A)

no son válidas, en general;

c) La función compatibilidad tomaría, si se extendiera su aplicación a las negaciones de cualidades elementales y a la cualidad universal U, los valores siguientes para los pares (A,N(A)), (A,U) y (U,U) —siendo A una cualidad elemental cualquiera—:

COMP(A,N(A)) = A*N(A) = A*(–1/A) = –1 = Z (valor equivalente a la cualidad contradictoria)

COMP(A,U) = A*U = A*1 = A (valor equivalente a la cualidad dada A)

COMP(U,U) = U*U = 1*1 = 1 = U (valor equivalente a la cualidad universal)

2.4.4. *Posibilidad de introducir en el sistema funciones más complejas*

En el marco, apenas esbozado, de la memoria de las cualidades, es posible, a nuestro juicio, introducir ulteriormente funciones de cualidad diferentes y más complejas que las que acabamos de definir. Con este fin, creemos que sería útil aplicar los recursos siguientes:

a) Reservar series bien determinadas de números primos (que dejarían de utilizarse entonces para representar las cualidades elementales de la esfera particular descrita), como por ejemplo, los números primos cuya expresión decimal termina por la cifra '9' (o cualquier otra cifra o grupo de cifras), para la construcción de *operadores especiales* (los más simples de los cuales serían de los tipos '19*', '(–1/29)*', etc.) que expresaran el sentido de las preposiciones ('de', 'con', 'sin', 'para', etc.), de ciertas flexiones y de otros *operadores lingüísticos*.

b) Utilizar *exponentes* eventuales (en el sentido aritmético de la palabra), aplicados ya a constantes o variables expresivas de cualidades elementales, ya a expresiones de funciones de cualidad, para marcar un *orden* —lineal o incluso ramificado— de sucesión de las cualidades o de las funciones precitadas en la construcción de funciones de cualidad más complejas (sometiéndose de este modo al carácter generalmente no-conmutativo de la mayoría de las funciones lingüísticas, si exceptuamos las que se expresan mediante ciertas conjunciones, como 'y', 'o', 'ni', etc.).

En este sentido, he aquí un ejemplo de expresión aritmética, fundado en la elección adecuada de los exponentes, de una función compleja F7, de estructura ramificada, construida a partir de un conjunto de funciones de cualidad más simples F1, F2, F3, F4, F5, F6:

Sea la representación ramificada de F7 la siguiente:

$$F7 = F1 \left< \begin{array}{l} F2 \left< \begin{array}{l} F4 \\ F5 \text{——} F6 \end{array} \right. \\ F3 \end{array} \right.$$

Entonces, utilizando el método apuntado, la expresión aritmética de la función compleja F7 sería del tipo siguiente:

$$F7 = F1 * F2^{2} * F4^{2*2} * F5^{2*3} * F6^{2*3*3} * F3^{3}$$

o, en lenguaje FORTRAN:

```
F7 = F1*(F2**2)*(F4**(2*2))*(F5**(2*3))*(F6**
                                  (2*3*3))*(F3**3)
```

Se puede constatar que, si las funciones F1, F2, F3, F4, F5, F6 son, como las funciones de cualidad hasta aquí definidas, funciones que han de tomar necesariamente *valores reales* cuando las variables que expresan cualidades elementales adoptan *valores enteros* (o, más exactamente, iguales a ciertos *números primos*), también las *funciones ramificadas*, del tipo de la F7 considerada, tomarán necesariamente *valores reales*. Y, si se completa el método imponiendo ciertas previsibles restricciones en la construcción de las funciones de cualidad, puede llegar a obtenerse el siguiente resultado: *Los valores adoptados por dos funciones de cualidad diferentes al atribuir valores numéricos a las variables de cualidad elemental serán siempre dos números reales diferentes.*

He aquí el principio de una *aritmetización* de las funciones de cualidad, aritmetización que, en la medida en que las citadas funciones de cualidad se acercaran a cubrir toda la esfera semántica de una rama científica particular, podría permitir un verdadero tratamiento automático de las estructuras características de la mencionada esfera.[9]

3. *La memoria de las proposiciones*

En esta memoria podrán introducirse las *proposiciones válidas* en una rama científica o esfera particular en forma de aceptación, admisión o afirmación de la validez de algunas de las funciones de cualidad anteriormente definidas, a partir de las cualidades elementales elegidas para la descripción de la mencionada esfera particular, y cuya expresión aritmética o algebraica ocupa la memoria de las cualidades. El paso de una función de cualidad de la memoria de las cualidades a la memoria de las proposiciones representa, pues, en el ordenador, el paso de una determinada relación del *nivel hipotético* de las relaciones lógicamente posibles al *nivel científico* de las relaciones aceptadas como efectivamente válidas en una esfera determinada.

3.1. *La aceptación de las funciones de cualidad en la memoria de las proposiciones*

Las funciones de cualidad F1, F2...FN, que se haya decidido aceptar como válidas en una esfera particular, serán

9. Se puede constatar que la aritmetización concebida utiliza, en lo que respecta a la traducción aritmética de un orden de sucesión por cierto orden de aplicación de los exponentes (descompuestos en sus factores primos) a las bases, criterios muy diferentes a los que fueron utilizados por Gödel, en sus intentos de aritmetización de la meta-matemática (Kurt Gödel, "Ueber formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I", *Monatshefte für Mathematik und Physik*, 38, 1931, pp. 173-198). En efecto, para Gödel, el orden estaba marcado por las bases, mientras que para nosotros lo está por lo exponente. Nuestro método hace posible, como se ve, la representación de una ramificación en la construcción de una función, el de Gödel no.

introducidas en la memoria de las proposiciones mediante una *declaración* del tipo de las declaraciones utilizadas en lenguaje FORTRAN, de la forma siguiente:

PROP F1, F2,...FN

Una vez declaradas como tales en la forma aludida, las funciones en cuestión serán tratadas como *proposiciones* a lo largo de todo el programa y podrán ser designadas utilizando *identificadores de variable proposicional*. Estos identificadores estarán formados por 1 hasta 6 caracteres alfabéticos FORTRAN, pero estos caracteres —que, en el caso de los identificadores de variable de cualidad elemental, eran necesariamente una de las letras A, B, C,...M— serán aquí necesariamente una de las letras P, Q, R, S, T.

Podrán escribirse, pues, fórmulas del tipo siguiente:

P = COMP(A,B) Q = DEP(C,F) S = INCOMP(7,29)

RTT = INDEP(5,41)

cuyo sentido, a los efectos del cálculo automático, consistirá en colocar, respectivamente, en los elementos de memoria del ordenador asignados a las variables proposicionales P, Q, S y RTT, las funciones de cualidad escritas, en las citadas fórmulas, a la derecha del signo '='.

3.2. *Funciones proposicionales: el producto de dos proposiciones P, Q*

Una función proposicional es una función —en el sentido matemático general— cuyos argumentos son proposiciones. Quedará expresada, en el cálculo automático, por una expresión algebraica que tomará un valor aritmético para cada juego posible de valores aritméticos de sus argumentos.

Por el momento, definiremos y construiremos una sola función proposicional —diádica, es decir, de dos argumentos—, la función 'producto de dos proposiciones P, Q,' utilizando para ello, como operador proposicional, el mismo operador diádico '*' ya utilizado, en su aplicación a las cualidades elementales, como operador de composición de cualidades. El sentido de este operador, a los efectos del cálculo, será siempre el de la multiplicación aritmética.

La definición del producto de dos proposiciones P,Q será, pues, la siguiente:

PROD(P,Q) = P*Q

3.3. *Regla fundamental de deducción*

Llamaremos 'regla de deducción', en el cálculo automático de cualidades, a toda regla, aplicable en forma automática, en virtud de la cual, a partir de una o varias funciones de cualidad ya aceptadas como proposiciones en la memoria de las proposiciones, se deduce la inmediata aceptación de otra u otras funciones de cualidad como proposiciones en la referida memoria.

Entre las reglas de deducción que es posible establecer en el cálculo de cualidades —unas, de validez general; otras, válidas en una esfera científica determinada—, nos limitaremos aquí a considerar la *regla fundamental de deducción*, que enunciaremos del modo siguiente:

Si dos funciones de cualidad cualesquiera F1 y F2 de las que una, al menos, ha de ser positiva, han sido aceptadas como proposiciones en la memoria de las proposiciones (asignándoseles, eventualmente, dos elementos de esa memoria, designados por los identificadores de variable proposicional P, Q, mediante fórmulas del tipo 'P=F1', 'Q=F2') *y si, al mismo tiempo, el resultado del producto F1*F2 presenta la forma de una función de cualidad definida en el sistema, entonces ese resultado será también aceptado como proposición en la memoria de las proposiciones* (pudiéndosele asignar también un elemento de esa memoria designado por un nuevo identificador de variable R, mediante una fórmula del tipo 'R=F1*F2').

Las deducciones efectuadas en virtud de la regla fundamental de deducción que acabamos de enunciar podrán presentarse en el cálculo automático utilizando una de las cuatro formas a), b), c), d) que indicamos a continuación:

a) PROP F1, F2
PROP F1*F2

b) P = F1
Q = F2
R = F1*F2

c) P
Q
R = P*Q

d) P*Q = R

3.4. *El axioma de autocompatibilidad*

En cada uno de los desarrollos posibles de un cálculo de cualidades —efectuado con vistas a su aplicación concreta a una esfera particular—, es posible establecer y utilizar *reglas de deducción específicas,* cuya validez puede depender, en ciertos casos, de la previa aceptación de determinados *axiomas.*

Entre los axiomas posibles, cuya aceptación o no aceptación en el cálculo de cualidades puede caracterizar esencialmente un tipo determinado de desarrollo de éste, por depender de la misma la validez y posible utilización de ciertas reglas especiales de deducción, consideraremos el *axioma de autocompatibilidad,* de cuya aceptación depende la validez de las *reglas de subalternación.*

Para poder formular el axioma de autocompatibilidad, es indispensable, a su vez, que se haya autorizado expresamente —al definir el *operador de composición*— la posible aplicación de dicho operador de dos argumentos *no distintos entre sí,* sino idénticos, es decir, expresión de *una misma cualidad elemental A.* La función resultante de tal aplicación será denominada *'función de autocompatibilidad'* de una cualidad elemental A y expresada del modo siguiente:

AUTOCOMP(A) = COMP(A,A) = A*A = A**2
(definición)

El axioma de autocompatibilidad se enunciará entonces como sigue:

Para toda cualidad elemental A aceptada en la memoria de las cualidades, la función AUTOCOMP(A) será aceptada como proposición en la memoria de las proposiciones.

Este axioma nos permitirá, pues, escribir:

a) Para toda cualidad elemental A: PROP AUTOCOMP(A) o, si se quiere:

b) Para toda cualidad elemental A: PROP COMP(A,A) o también:

c) Para toda cualidad elemental A: PROP A*A o finalmente:

d) Para toda cualidad elemental A: PROP A**2

3.5. *Las reglas de subalternación*

Las *reglas de subalternación* se deducen, como teoremas, del axioma de autocompatibilidad y de la regla fundamental de deducción.

I. La primera regla de subalternación se enuncia del modo siguiente:

Si una función de dependencia DEP(A,B) de una cualidad elemental A respecto de otra cualidad elemental B ha sido aceptada como proposición en la memoria de las proposiciones, entonces también la correspondiente función de compatibilidad COMP(A,B) entre esas dos mismas cualidades A, B será aceptada como proposición.

Dando como aceptado el axioma de autocompatibilidad y aplicando la regla fundamental de deducción, en su forma b), podemos deducir, a partir de cualquier proposición P de aceptación de una función de dependencia DEP(A,B) arbitrariamente elegida, una proposición R de aceptación de la correspondiente función de compatibilidad COMP(A,B), de la manera siguiente:

P = DEP(A,B) = B/A (proposición de partida)

Q = AUTOCOMP(A) = COMP(A,A) = A*A = A**2
(axioma de autocompatibilidad, aplicado a la cualidad elemental A)

R = P*Q = (B/A) * (A**2) = B*A = A*B =
= COMP(A,B)

Como hemos elegido arbitrariamente la función de dependencia DEP(A,B), vemos que, una vez aceptado el axioma de autocompatibilidad, la regla I de subalternación se aplica siempre automáticamente, como consecuencia de ese axioma y de la aplicación de la regla fundamental de deducción.

Las deducciones efectuadas en virtud de la regla I de subalternación podrán presentarse en el cálculo automático utilizando una de las tres formas a), b), c) que indicamos a continuación:

a) PROP DEP(A,B)
PROP COMP(A,B)

b) P = DEP(A,B)
Q = COMP(A,B)

c) P = B/A
Q = A*B

II. Análogamente, la segunda regla de subalternación se enuncia del modo siguiente:

Si una función de incompatibilidad INCOMP(A,B) de dos cualidades A, B ha sido aceptada como proposición en la memoria de las proposiciones, entonces también la correspondiente función de independencia INDEP(A,B) de la cualidad A referida respecto de la referidad cualidad B será aceptada como proposición.

La validez de esta regla II de subalternación, una vez aceptado el axioma de autocompatibilidad, se deduce en forma análoga a la utilizada para la regla I:

P = INCOMP(A,B) = –1/(A*B) (proposición de partida)
Q = AUTOCOMP(A) = COMP(A,A) = A*A = A**2
(ax. de autocompatibilidad)
R = P*Q = (–1/(A*B)) * (A**2) = –A/B =
= INDEP(A,B)

Las deducciones efectuadas en virtud de la regla II de subalternación podrán presentarse en el cálculo automático utilizando una de las tres formas: a), b), c) que indicamos a continuación:

a) PROP INCOMP(A,B)
PROP INDEP(A,B)
b) P = INCOMP(A,B)
Q = INDEP(A,B)
c) P = –1/(A*B)
Q = –A/B

4. *Aplicación del cálculo de cualidades a la deducción automática de todos los modos válidos del silogismo categórico por multiplicación de los números o expresiones algebraicas característicos de las premisas*

Es posible interpretar lógicamente la admisión como proposiciones de las cuatro funciones diádicas de cualidad del modo siguiente:

PROP COMP(A,B)	o P = A*B	'Algún A es B'	(P.A.)
PROP INCOMP(A,B)	o P = –1/(A*B)	'Ningún A es B'	(U.N.)
PROP INDEP(A,B)	o P = –A/B	'Algún A no es B'	(P.N.)
PROP DEP(A,B)	o P = B/A	'Todo A es B'	(U.A.)

Basándose en esa interpretación, la deducción automática por el ordenador de los 24 modos válidos del silogismo categórico se presenta del modo siguiente:

I. Para los 15 modos fundamentales, que no exigen la aplicación de las reglas de subalternación —de acuerdo con la teoría clásica— (y por lo tanto tampoco del axioma de autocompatibilidad), mediante la aplicación de la regla fundamental de deducción, que permite obtener la expresión (numérica o algebraica) de la conclusión, *por simple multiplicación de las expresiones respectivas de las premisas:*

Cálculo aritmético o algebraico	Conclusión lógica correspondiente	Modo
1. (B/M) * (M/A) = B/A	DEP(M,B) * DEP(A,M) = DEP(A,B)	BARBARA
2. (–1/(M*B)) * (M/A) = –1/(A*B)	INCOMP(M,B) * DEP(A,M) = INCOMP(A,B)	CELARENT
3. (B/M) * (A*M) = A*B	DEP(M,B) * COMP(A,M) = COMP(A,B)	DARII
4. (–1/(M*B)) * (A*M) = –A/B	INCOMP(M,B) * COMP(A,M) = INDEP(A,B)	FERIO
5. (M/B) * (–1/(A*M)) = –1/(A*B)	DEP(B,M) * INCOMP(A,M) = INCOMP(A,B)	CAMESTRES
6. (–1/(B*M)) * (M/A) = –1/(A*B)	INCOMP(B,M) * DEP(A,M) = INCOMP(A,B)	CESARE
7. (M/B) * (–A/M) = –A/B	DEP(B,M) * INDEP(A,M) = INDEP(A,B)	BAROCO
8. (–1/(B*M)) * (A*M) = –A/B	INCOMP(B,M) * COMP(A,M) = INDEP(A,B)	FESTINO
9. (B/M) * (M*A) = A*B	DEP(M,B) * COMP(M,A) = COMP(A,B)	DATISI
10. (–1/(M*B)) * (M*A) = –A/B	INCOMP(M, B) * COMP(M,A) = INDEP(A,B)	FERISON
11. (–M/B) * (A/M) = –A/B	INDEP(M,B) * DEP(M,A) = INDEP(A,B)	BOCARDO
12. (M*B) * (A/M) = A*B	COMP(M,B) * DEP(M,A) = COMP(A,B)	DISAMIS
13. (M/B) * (–1/(M*A)) = –1/(A*B)	DEP(B,M) * INCOMP(M,A) = INCOMP(A,B)	CALEMES
14. (–1/(B*M)) * (M*A) = –A/B	INCOMP(B,M) * COMP(M,A) = INDEP(A,B)	FRESISON
15. (B*M) * (A/M) = A*B	COMP(B,M) * DEP(M,A) = COMP(A,B)	DIMATIS

II. Para los cuatro modos que, según la teoría clásica se deducen aplicando la 'conversio per accidens' a una de las premisas (indicada por la inclusión de la letra 'P' en el nombre tradicional del modo), también nuestro cálculo opera la deducción *aplicando la regla I de subalternación*, lo cual equivale a dar por supuesta la validez del axioma de autocompatibilidad (cuya interpretación lógica, de acuerdo con el esquema antes señalado, es la afirmación de la validez general, para toda cualidad A, de la proposición 'Algún A es A').[10] La deducción se presenta del modo siguiente:

16. PROP B/M, A/M — DARAPTI
PROP B/M, M*A
(B/M) * (M*A) = A*B (se reduce a DATISI)

17. PROP –1/(M*B), A/M — FELAPTON
PROP –1/(M*B), M*A
(–1/(M*B)) * (M*A) = –A/B (se reduce a FERISON)

18. PROP M/B, A/M — BAMALIP
PROP B*M, A/M
(B*M) * (A/M) = A*B (se reduce a DIMATIS)

19. PROP –1/(B*M), A/M — FESAPO
PROP –1/(B*M), M*A
(–1/(B*M)) * (M*A) = –A/B (se reduce a FRESISON)

---

10. La validez de las reglas de subalternación, al igual que la de nuestro axioma de autocompatibilidad, estrechamente ligado a ellas, puede ser discutida, y es posible concebir un sistema lógico más estricto que no las admita. Aceptar tales reglas puede considerarse equivalente a excluir de toda proposición el caso de la clase nula o vacía, en la perspectiva extensional, y de la cualidad contradictoria, en la perspectiva intensional (véase, a propósito de esa y de otras equivalencias de las reglas de subalternación el Apéndice a la Parte IV de nuestra obra ya citada, pp. 129-130). En efecto, el axioma de autocompatibilidad no se aplica a la "cualidad contradictoria". Véase también la observación del lógico de Lovaina Joseph Dopp, en este sentido: "Si on ne présupposait pas que les concepts sont vérifiés par un objet, au moins, 9 modes cesseraient d'être valables" (exactamente los 9 modos para cuya deducción nuestro cálculo precisa la aceptación previa de las reglas de subalternación o del axioma de autocompatibilidad), en sus *Notions de Logique Formelle*. Paris - Louvain, 1965.

III. Para los cinco modos de conclusión atenuada (Barbari, Celaront, Cesarop, Camestrop y Calemop) se aplican, respectivamente, la primera (al primero de los modos citados) y la segunda (a los cuatro restantes) regla de subalternación a las conclusiones de los modos Barbara, Celarent, Cesare, Camestres y Calemes, de un modo enteramente análogo al que hemos utilizado anteriormente.[11]

11. Es posible demostrar en nuestro cálculo que, fuera de los 24 modos válidos del silogismo categórico, que acabamos de deducir automáticamente, ningún otro de los 232 modos teóricamente imaginables, cuya conclusión sería falsa, no puede ser deducido. Por otra parte, es útil recordar aquí, para situar históricamente este problema, la observación del gran lógico polaco Jan Lukasiewicz sobre la primitiva interpretación aritmética de la silogística efectuada por Leibniz en 1679 (interpretación mucho más complicada que la que ofrecemos en este trabajo, habiendo nosotros desarrollado completamente también aquélla en la obra ya citada): "Leibniz once said that scientific and philosophic controversies could always be settled by a calculus. It seems to me that his famous 'calculemus' is connected with the above arithmetical interpretation of the syllogictis rather than with his ideas on mathematical logic" (Jan Lukasiewicz, *Aristotle's syllogistic from the standpoint of modern formal logic*. Oxford, Clarendon Press, 1951, p. 129).