Lógica matemática: Álgebra de Boole y análisis práctico

4a Edició Premi Poincaré 2007

Segon premi

Lógica matemàtica. Álgebra de Boole y análisis práctico

Per les diferents aplicacions pràctiques que fa de les quals destaquem la presentació acurada en diferents formats, incloent una pàgina web.

Autor: Román Sandoval Villamora

Centre: Aula Escola Europea (Barcelona)

Tutor: Sr. Joan Alemany

Trabajo de investigación

Agradecimientos,

Me gustaría agradecer a una serie de personas su colaboración con este trabajo de investigación.

En primer lugar, me gustaría agradecer a los tres tutores que he tenido en este trabajo: la Sra.Granados, la Sra.García-Matos, y, finalmente, al Sr. Alemany. Especialmente agradezco a este último su dedicación en el trabajo, dado que ha sido con el que más tiempo he tenido para desarrollar el trabajo. Todos ellos me han prestado sus conocimientos para que yo pudiera realizar mi investigación.

En segundo lugar, agradezco al Sr. Leguizamon sus correcciones y anotaciones en la parte más filosófica del trabajo.

En tercer lugar, agradezco al Sr. Valls su colaboración en este trabajo de investigación. Además, le agradezco que me cediera todos los instrumentos necesarios para realizar las prácticas.

La lógica es invencible, porque para combatir a la lógica se usa a la lógica

Pierre Boutroux

Las matemáticas, cuyas ideas básicas de contar, medir y deducir han acompañado al hombre a lo largo de su evolución como una de sus máximas creaciones: constituyen la base de la ciencia y la tecnología actuales y nos conducirán a las estrellas.(...)El gran desarrollo de las matemáticas durante los últimos siglos ha permitido establecer una lógica formal, desligada de las imprecisiones semánticas, que ha recibido el nombre de lógica matemática.

Matemáticas, Lógica, conjuntos, geometría y estadística

Miguel Angel Sainz

La lógica muestra un devenir histórico muy interesante, naciendo de la fuerte formalización de las matemáticas de los griegos, que fue impactada, como muchas ciencias, por el pensamiento de la Edad Media, donde la religión se anteponía a todo; pero, el ímpetu de la mente de los filósofos renacentistas ayudó a retomar su desarrollo. No cabe duda que la lógica tiene impactó fundamental, como ciencia de las ciencias, en el pensamiento contemporáneo, y que el nacimiento de la tecnología computacional deba mucho al desarrollo del formalismo lógico de principios de siglo.

http://www.xtec.es/~jdomen28/article101.htm

Índice

1. Introducción..........9

1.1. Interés previo y motivaciones..........11

1.2. Composición del trabajo..........12

1.3. Tesis..........13

1.4. Objetivos..........13

2. Introducción a la lógica..........14

2.1. ¿Qué es la lógica?..........14

2.2. Lenguaje preposicional..........16

2.2.1. Términos de enlace..........18

2.2.2. La forma de las proposiciones moleculares..........19

2.3. Breve reseña histórica de la lógica..........19

2.4. Valoración personal..........22

3. La lógica matemática..........25

3.1. ¿Qué es la lógica matemática?..........26

3.2. Tabla de verdad..........27

3.3. Nexos lógicos principales..........29

3.3.1. Nexo lógico "Y"..........30

3.3.2. Nexo lógico "O"..........31

3.3.3. Operación lógica "No"..........33

3.4. Circuitos eléctricos..........35

3.5. Simbología en puertas lógicas..........38

3.6. Autómatas..........38

3.7. Práctica de los conceptos previamente explicados..........39

3.7.1. Persiana..........39

3.7.2. Puerta de garaje..........43

3.7.3. Regulación por intermitencia..........44

3.7.4. "Scalextric" calculadora..........47

3.8. Puertas lógicas derivadas..........48

3.8.1. Puerta lógica derivada "XOR" 48
3.8.2. Puerta lógica derivada "NOR" 49
3.8.3. Puerta lógica derivada "NAND" 50
3.8.4. Puerta lógica derivada "XNOR" 51
3.9. Valoración personal 52

4. Álgebra de Boole 54

4.1. Introducción 55
4.2. Propiedad conmutativa 55
4.3. Propiedad distributiva 57
4.4. Elementos cero y unidad 59
4.5. Complementación 59
4.6. Principio de dualidad 60
4.7. Leyes de Morgan 60
4.7.1. Demostración de la primera ley de Morgan 63
4.7.2. Demostración de la segunda ley de Morgan 64
4.8. Teoría de conjuntos 65
4.9. Posibilidades de resultado con nexos lógicos 72
4.10. Valoración personal 74

5. Decibilidad 75

5.1. Concepto de decibilidad 76
5.2. Problema de satisfacibilidad booleana (SAT) 80
5.3. Buscaminas 81
5.3.1. En qué consiste el juego 81
5.3.2. Azar en el juego 84
5.3.3. Relación entre el programa y el ¿P=NP? 85
5.3.4. Algoritmos relacionados con el buscaminas 86
5.3.5. Consistencia del buscaminas 87
5.3.6. Resolución de un buscaminas 87
5.3.7. Análisis comparativo entre nuestro algoritmo y el de Windows 89
5.4. Valoración personal 91

6. Incidencias 93

7. Posibles líneas de investigación 98

8. Conclusiones 101

9. Glosario...104

10. Bibliografía...108

1.Introducción

En este primer apartado se expondrán las motivaciones, tesis y objetivos planteados durante la realización del trabajo de investigación.

Hace dos años, cuando ni me había planteado la idea de cómo tenía que realizar mi trabajo de investigación, me surgían muchas dudas. La más frecuente, si realmente sería capaz de realizar un trabajo de investigación como éste. Pues ahora puedo decir, una vez acabado el trabajo, que sí. He sido capaz, y además he acabado orgulloso de mi trabajo. No obstante, sé que el trabajo que yo haya podido realizar, será un grano de arena al lado de tantos otros trabajos anteriormente realizados sobre el tema.

Así pues, y la primera idea que quiero dejar clara, es que en ningún caso tengo la intención de hacer de este trabajo, un libro de ciencia, dado que libros sobre este tema, como ya he dicho, hay muchos técnicamente más completos. No obstante, este trabajo tiene en primer lugar, la intención de resumir todos los libros y estudios que he leído y analizado durante el tiempo de realización del mismo. De esta manera, quiero dar un nuevo enfoque a un tema que, normalmente, los libros explican de manera muy técnica y difícil de entender. Por otra parte, y la diferencia más notable con los libros científicos, es la personalidad con la que he dotado a este trabajo: es un trabajo muy personal, con apuntes propios dentro de cada apartado.

Otro aspecto a destacar es que durante la realización de este trabajo de investigación, me he ido adentrando en un mundo complejo, pero al mismo tiempo un mundo interesante, fascinante incluso. Gracias a este interés por la tecnología, y a que me divertía haciendo maquetas, leyendo sobre el tema, he podido profundizar sobre muchos aspectos necesarios para entender este tema .Ahora me pregunto como no podía saber cosas como de la misma manera que siempre nos han dicho que todo es química, me atrevería a decir, que todo es lógica matemática, dado que actualmente vivimos en una sociedad que usa como primer recurso para realizar cualquier acción nuevas tecnologías, que a su vez, funcionan todas con 0 y 1. Obviamente es una afirmación muy generosa, pero si lo miramos de una manera más global, teniendo en cuenta el gran campo que abarca la lógica matemática, nos daremos cuenta que es cierta.

En este trabajo, y al principio de cada gran tema, habrá un pequeño prólogo en “cursiva” para introducir al lector al tema que se va a tratar en el siguiente apartado. Por otra parte, se ha preparado un anexo donde el lector encontrará una ampliación de los temas tratados en este trabajo, así como las fotografías de las prácticas. Además, se ha creado una página web donde se podrá encontrar la parte multimedia del trabajo, así como la parte teórica del trabajo. La dirección web es:

http://www.logicamatematica.net

1.1 Interés previo y motivaciones

Después de haberme planteado la posibilidad de realizar diferentes trabajos de investigación, este me interesó particularmente interesante para profundizar en los conocimientos sobre sistemas de numeración binaria, que había estudiado previamente en Informática. Además, quería analizar las aplicaciones que tiene esta parte de la matemática en las tecnologías de la información y comunicación. En un primer lugar, me había planteado temas demasiado ambiciosos, y cuando parecía que ya había encontrado el que me parecía más interesante, me dijeron que era un tema demasiado usual para repetirlo. Finalmente, encontré el tema que me parecía ideal para mi trabajo: la lógica matemática.

Aunque no tenía mucha conciencia de en el mundo en qué me estaba adentrando, me pareció acertado escoger este tema, dado que vivimos en un mundo basado en las nuevas tecnologías. Actualmente pocas cosas se hacen si uso de éstas. De esta manera, me quería interesar en un mundo fascinante, pero a la vez complejo y desconocido por la mayoría.

Cuando empecé a leer libros y a recoger información sobre el tema, me di cuenta que me estaba adentrando en un mundo fundamental para nuestra cultura, pero a la misma vez, muy complejo. Así pues, tuve que hacer una selección de información para ver qué era importante y qué era prescindible, dado que había una gran cantidad de estudios, tesis y libros, sobre la lógica matemática.

El campo de la lógica matemática es una disciplina que todos tenemos muy cerca, pero no todo el mundo sabe que existe. Actualmente, todo funciona con computadores lógicos, desde la persiana que hay en nuestras casas, hasta los trenes, pasando por Internet y los relojes. En general todas las nuevas tecnologías funcionan con esta parte de las matemáticas.

Me llama particularmente la atención investigar los orígenes de una parte de la matemática que inicialmente parecía que no tenían ninguna utilidad, y que con el tiempo se ha convertido en una pieza clave del desarrollo tecnológico. Así pues, es un trabajo más práctico que teórico, y hay temas muy interesantes de esta parte de la matemática, que no se van a tratar en el trabajo. En el caso de la reseña histórica de la lógica, podríamos haber profundizado mucho más si hubiéramos querido realizar un trabajo más teórico. No obstante, creemos que una simple, aunque no sencilla, reseña histórica es suficiente para entender el apartado, y poder centrarnos en el trabajo de investigación de la lógica matemática.

Si anteriormente afirmamos que inicialmente la lógica matemática no iba a servir para nada, es dado que muchas veces los descubrimientos de teorías y de fórmulas, son de manera involuntaria y en un principio no sirven para lo que podrán servir en un futuro. No podemos saber, si una persona que ha planteado una teoría sobre la composición de los electrones, podrá demostrarla, o simplemente la tendrá que dejar de lado a causa de que no la puede probar. No obstante, en el caso de Boole, no inventó la lógica matemática para el sistema computacional que utilizamos actualmente, pero debido a la evolución de las matemáticas sus teorías nos han servido para llegar a crear una sociedad basada en esta ciencia.

1.2 Composición de trabajo

Por otra parte, este trabajo de investigación consistirá en hacer, en primer lugar, un estudio teórico de la lógica matemática y binaria, y después, aplicar los conocimientos adquiridos en una parte experimental. En la parte teórica trataremos temas como los antecedentes históricos de la lógica matemática, introduciremos los elementos fundamentales de este tipo de lógica matemática, veremos ejemplos con proposiciones lógicas, entre muchas otras cosas. Para entender mejor estos conceptos, iremos haciendo pequeños experimentos, como es el caso de una persiana, o una bomba de agua. A lo que se refiere a la parte práctica de este trabajo, intentaremos, mediante los conocimientos previamente adquiridos en la parte anterior del trabajo, hacer dos experimentos: en primer lugar, y el más sencillo, será de hacer una puerta de ascensor, que utilice un sistema lógico de apertura y cerrada. La segunda parte del experimento, consistirá en programar un autómata programable para que haga ciertas funciones lógicas.

1.3 Tesis

La tesis de mi trabajo será mostrar que la lógica binaria, una teoría matemática abstracta, tiene una aplicación práctica y concreta en el control automático. Además, intentaremos aplicar la lógica binaria, juntamente con las tecnologías de la computación electrónica, para controlar "procesos secuenciales", que son procesos que convierten secuencialmente las instrucciones de un programa.

Para entender la tesis que me he planteado en este trabajo, se ha planteado una serie de objetivos. Cuando empecé a planteármelos, encontré algunos de muy ambiciosos, otros muy interesantes pero poco ambiciosos, otros poco interesantes pero ambiciosos. Empecé haciéndome una lista de los objetivos que me planteaba en un primer momento, y me di cuenta que a medida que iba avanzando con el trabajo, y que las investigaciones y lecturas se iban haciendo más extensas, me iba planteando otros que se sobreponían a los que ya tenía. Así pues, tuve que hacer una especie de selección de objetivos y de ideas que quería tratar en este trabajo de síntesis y de investigación para que me saliera el mejor trabajo posible. Llegué a la decisión de seleccionar los siguientes objetivos.

1.4 Objetivos

- Entender las operaciones lógicas básicas
- Entender qué es la lógica y la lógica matemática
- Analizar el Álgebra de Boole
- Entender porqué es tan importante esta parte de las matemáticas
- Ver qué aplicaciones tiene
- Hacer una parte experimental que resuma todas las ideas anteriormente planteadas

2. Introducción a la Lógica

En este primer apartado del trabajo analizaremos detalladamente qué es la lógica. En primer lugar, veremos cual es el significado de este concepto. En segundo lugar, estudiaremos el origen de los sistemas lógicos estructurados, y haremos un breve recorrido por su historia. Finalmente, llegaremos a la formulación de la lógica matemática, a sus orígenes y sus objetivos.

2.1. ¿Qué es la lógica?

Desde hace mucho tiempo, la **lógica** es la parte de la filosofía que se ocupa de estudiar los razonamientos y su estructura, distinguiendo así, cuales son correctos y explica por qué lo son. El **razonamiento lógico** es una de las características del pensamiento humano y consiste en encadenar proposiciones lógicas para realizar una argumentación. Está compuesto por conceptos y razonamientos. Los conceptos son las palabras y los razonamientos las frases. A partir de esta estructuración se forman las oraciones lógicas. Existen dos tipos de lógica: la **lógica formal** y la **informal**. La lógica formal se ocupa de estudiar las formas convenientes del razonamiento. No debemos confundir la verdad de la conclusión con la validez del razonamiento. Es por esta razón, que se ha puesto que es la materia que se ocupa de analizar las manera convenientes de razonar, pero estas no han de ser obligatoriamente ciertas, dado que algo conveniente no es siempre verdadero. En esta parte de la lógica, encontramos los **silogismos**. Por otra parte, la lógica informal examina los razonamientos teniendo en cuenta la construcción a veces errónea del pensamiento humano. Así pues, ésta tiene como objetivo fundamental la búsqueda de la verdad, y distinguir entre argumentos correctos y falaces.

La racionalidad es una característica pura del ser humano, que consiste en hacer un uso adecuado de la razón. Los humanos utilizamos la razón para formar nuestras creencias, elegir nuestras finalidades y preferencias, y dirigir nuestras acciones. Las decisiones humanas no son siempre fruto de un uso estricto de la razón y de sus reglas. Cuando hacemos uso de la razón, nuestras creencias, preferencias o acciones se convierten en racionales. Que una creencia, una preferencia o una acción sean racionales significa que han sido formadas, elegidas y dirigidas, respectivamente, por buenas razones. Y, si es posible, por las mejores razones.

En general, nuestras creencias se forman a partir de otras creencias o conocimientos. Así pues, la información que disponemos sirve de base de nuestras nuevas creencias. Nos referimos al hecho de razonar o argumentar. Pero a veces razonamos incorrectamente y nos equivocamos con la opinión que sacamos de la información disponible. Por ejemplo:

Joaquín va a la universidad

Joaquín juega a tenis

Todas las personas de la universidad juegan a tenis

En este caso el problema es la mala formulación del silogismo. Por lo tanto, se ha cometido un error en la forma de razonar. A veces un razonamiento puede estar bien estructurado, pero su conclusión ser falsa, porque uno de los datos de la premisa era erróneo, por ejemplo:

Las cosas verdes son vegetales

El coche es verde

El coche es un vegetal

Si el *como* (razonamiento lógico) llegamos a nuestras creencias es importante, también lo es el *que* (premisas o datos de partida que usamos) creemos. Ambos son imprescindibles si queremos tener verdaderos conocimientos y no simples creencias, quizás equivocadas. No obstante, para la lógica, que recordamos es la ciencia de los razonamientos, lo importante es solo *como* razonamos, *como* formamos nuestras ideas a partir de la que ya disponemos. Así pues, la lógica es la ciencia que estudia los razonamientos correctos o validos y porqué lo son.

El hombre se caracteriza por ser racional, que hace uso de la razón. Así pues, y teniendo en cuenta que la lógica es la ciencia del razonamiento, podemos deducir que el progreso en el conocimiento de la lógica humana, es el progreso del conocimiento del ser humano. El ser humano ha sido un ser lógico durante toda su existencia, no podemos decir que haya habido un momento en que el ser humano empezó a ser un ser racional. En un primer momento, la lógica sirvió para entender y aprovechar la naturaleza. No obstante, podemos afirmar que durante la historia de la humanidad ha habido diferentes momentos clave para la lógica. En primer lugar el concepto de lógica aristotélica. Si seguimos cronológicamente en el tiempo veremos que la revolución científica cambiará por completo la manera de entender el mundo, de razonar. Finalmente, a mediados del siglo XIX las teorías de George Boole provocarán una nueva interpretación de esta parte de la filosofía.

2.2 Lenguaje proposicional

Con el estudio de la lógica se intenta llegar a ser minucioso. La lógica tiene un lenguaje exacto. Para realizar este trabajo se utilizarán **proposiciones** en lengua castellana, de la misma manera que se usa la lengua castellana para explicar las reglas precisas de un juego a alguien que no ha jugado a este juego. Por supuesto, la lógica es algo más que un juego. Puede ayudarnos a aprender una forma de razonar que es exacta y a la vez muy útil.

Para empezar, consideraremos las proposiciones en lengua castellana. Cada proposición tiene una forma lógica a la que se le dará un nombre. En primer lugar, se consideran y simbolizan dos clases de proposiciones en Lógica; unas se denominan proposiciones ***atómicas*** y otras proposiciones ***moleculares***.

En este siglo de la ciencia se utiliza la palabra *atómico* muchas veces. Efectivamente, el significado de esta palabra en el lenguaje de la lógica es análogo a su significado original en las Ciencias físicas. En Lógica, *atómicas* son las proposiciones de forma más simple. Si se juntan una o varias proposiciones *atómicas* con un término de enlace, se tiene una proposición *molecular*. Una proposición *atómica* es una proposición completa sin términos de enlace. Se utilizan términos de enlace para formar proposiciones *moleculares* a partir de proposiciones *atómicas*.

Por ejemplo, si consideramos dos proposiciones *atómicas*:

Hoy es lunes

Hay clase

Al ser las dos proposiciones, atómicas, las podemos unir mediante un término de enlace y se tendrá una proposición molecular:

Hoy es lunes y hay clase

Esta proposición molecular se ha construido con dos proposiciones atómicas y el término de enlace "y". Cuando analizamos una proposición molecular la descomponemos en las más pequeñas proposiciones atómicas completas. En el ejemplo anterior se puede descomponer la proposición molecular en dos proposiciones atómicas. El término de enlace "y" no forma parte de ninguna de las proposiciones atómicas. Se ha añadido a las proposiciones atómicas para construir una proposición molecular.

2.2.1 Términos de enlace

Las palabras de enlace, por cortas que sean, no deben subestimarse, pues son de gran importancia. Tanto es así, que se estudiarán algunas reglas muy precisas para el uso de esta clase de términos. Gran parte de lo que se tratará en el estudió de la lógica matemática se refiere a la manera cuidadosa de cómo se han de utilizar estos términos de enlace. El término de enlace en la proposición del ejemplo "Hoy es sábado y no hay clase" es la palabra "y". Hay otros, pero antes de considerar cada uno de ellos separadamente, les daremos el nombre lógico correcto. Se les denominará ***términos de enlace de proposiciones***. Este nombre será fácil de recordar, porque indica efectivamente cuál es el papel que desempeñan. Enlazan proposiciones. Forman proposiciones moleculares a partir de proposiciones atómicas.

Los términos de enlace que se utilizarán en esta parte son los dos principales: "y" y "o" .Estos dos nexos lógicos, "**y**" y "**o**", se usan para enlazar dos proposiciones atómicas. Sin embargo, tenemos un tercer **elemento lógico**, el "**no**", que se agrega a una sola proposición lógica para formar una proposición molecular. Cuando una sola proposición se le agrega "no" se forma una proposición molecular. Así pues, se puede decir que el término de enlace "no" actúa sobre *una* sola proposición y que los otros dos nexos actúan sobre *dos* proposiciones atómicas a la vez.

A continuación se dan algunos ejemplos de proposiciones moleculares que utilizan los términos de enlace considerados.

- La luna no está hecha de queso verde

En este caso la proporción es una proposición molecular que utiliza el término de enlace "no". En este caso, el término de enlace actúa sobre *una* proporción atómica: "la luna está hecha de queso verde".

- La música es muy suave o la puerta está cerrada

Este es un ejemplo de proposición en la que se utiliza un nexo "o". Este nexo actúa sobre *dos* proposiciones *atómicas*, "la música es muy suave" y "la puerta está cerrada".

2.2.2 La forma de las proposiciones moleculares

Las reglas para el uso de los términos de enlace son las mismas, cualesquiera que sean proposiciones atómicas que enlazan o en las que se han utilizado. La forma de las proposiciones moleculares construidas depende del término de enlace seguido, no del contenido de la proposición o proposiciones atómicas. Es decir, si en una proposición molecular se sustituyen las proposiciones atómicas cualesquiera, la forma de la proposición molecular se conserva. La misma manera de escribir un nuevo término de enlace: "si...entonces..." lo indica. Los puntos suspensivos después de "si" y los puntos suspensivos después de "entonces" ocupan el lugar de las proposiciones. Para formar proposiciones moleculares utilizando este término de enlace basta simplemente sustituir los puntos suspensivos por proposiciones *atómicas* cualesquiera.

En esta primera parte del análisis del lenguaje proposicional en nuestro lenguaje, se ha querido de alguna manera introducir al lector a lo que posteriormente será el lenguaje proposicional en lenguaje matemático. Así pues, en este primer apartado del lenguaje proposicional, se ha de destacar los tres elementos lógicos principales: "y", "o" y "no".

2.3 Breve reseña de la evolución histórica de la lógica

Aristóteles (384-332 a.C.) fue el primero en intentar establecer la lógica como una ciencia, es decir: formular unos procedimientos estructurados, que expliquen como funciona el pensamiento racional humano. Además, fue la primera persona en utilizar el término lógica refiriéndose al estudio de los argumentos dentro del lenguaje natural y se refería a la lógica como "El arte de la argumentación correcta y verdadera". De esta manera, y recordando la definición anteriormente descrita de la lógica informal, podemos ver que él es el *padre* de la lógica informal.

Posteriormente, se sucederá un periodo en el cual la religión se sobrepondrá a la creatividad intelectual. En la Edad media se impuso la idea según la cual el conocimiento era revelado por Dios a través de la Biblia y de los pensadores antiguos aceptados por la iglesia. En el siglo XVI, el espíritu científico, observador y experimentador del Renacimiento permitió que se produjeran grandes avances en los campos de la medicina o la anatomía. No obstante, intelectuales como Copérnico o

Galileo tuvieron grandes dificultades en la difusión de sus ideas a causa de la inflexibilidad de la iglesia. La llamada **revolución científica**, que se extendió a lo largo del siglo XVII, produjo importantes aportaciones en campos muy diversos. La revolución científica supuso un cambio en la mentalidad de la sociedad, dado que ella se funda el método científico, basado en la observación, la experimentación y la demostración de experiencias mediante la razón. Los tres principales protagonistas de la revolución científica fueron el filósofo y matemático René Descartes, el físico Isaac Newton y el también filósofo y matemático Gottfried W. Leibniz

René Descartes (1596-1650), autor del *Discurso del método*, afirmó la superioridad de la razón como fundamento de todo conocimiento. Intentó aplicar a la filosofía los procedimientos racionales e inductivos de las matemáticas. En este ensayo, cogió como punto de partida, la duda universal, es decir, dudar de lo que anteriormente le había planteado algún fallo o indecisión, para encontrar una verdad absoluta. Dudó del conocimiento, de los sentidos, hasta de la razón. Finalmente, encontró esa verdad absoluta que nadie le podía negar: "Cogito, ergo sum" (*Pienso luego soy)*. Antes de este ensayo, la filosofía se basaba en el contraste y la comparación de opiniones de grandes intelectuales. No obstante, él rechazó esta idea y estableció: "En nuestra búsqueda del camino directo a la verdad, no deberíamos ocuparnos de objetos de los que no podamos lograr una certidumbre similar a las de las demostraciones de la aritmética y la geometría". De esta manera, Descartes concluyó que él no creería en ninguna verdad hasta haber establecido las razones para poder confiar en ella. Como matemático, Descartes sistematizó la geometría analítica, elaboró la teoría de ecuaciones, fue el responsable de la utilización de las últimas letras del abecedario para representar las incógnitas en las ecuaciones, inventó el método de exponentes para indicar las potencias de los números (ejemplo, x^3, x78), y finalmente dedujo el signo positivo y negativo de una raíz.

Avanzando cronológicamente en el tiempo, llegaremos a la época renacentista. Esta etapa se caracterizará por el uso de los infinitesimales en primer lugar, y posteriormente su eliminación y puesta en escena de los límites. Por otra parte, en este período se implantaran nuevos modelos científicos, como son el de la lógica simbólica, la lógica booleana, la inducción matemática y el cálculo preposicional. Uno de los personajes más importante será George Boole.

George Boole (1816-1864) fue un lógico y un matemático. Este intelectual británico aplicó el cálculo matemático a la lógica, fundando de esta manera, el álgebra de la lógica, o la lógica matemática. Si anteriormente en la lógica verbalizada, los conceptos eran las palabras, y los razonamientos oraciones, en la lógica matemática, los conceptos se traducirán en variables ponderadas (en el caso de la lógica binaria de 0 y 1), y los razonamientos en operaciones. De esta manera, para representar conceptos e ideas, lo podremos hacer mediante símbolos y reglas matemáticas. Así pues, Boole contribuyó a aclarar la estructura de los objetos lógicos, y a hacer ver que dos aspectos que aparentemente son dos polos opuestos, la lógica y las matemáticas, tienen muchas cosas en común. Más adelante, analizaremos detalladamente aspectos de sus análisis y teorías.

Una de las partes del trabajo es analizar las teorías de **Auguste De Morgan** (1806-1871). Este científico era, como Boole, un matemático. Estos dos autores se aprecian mucho mutuamente, hecho que nos deja ver que se pusieran de acuerdo para publicar sus investigaciones en 1847. Boole publicará *Mathematical analysis of logic*, mientras que el segundo, Auguste De Morgan, publicará *Formal Logic*. No obstante, el conjunto de sus obras será muy distante. Mientras que Boole construirá una teoría unificada, muy sistemática, la obra de Morgan es mucho más dispersa y variada. Después del estudio que consagró a Morgan, Liard afirmará que: "Sus teorías abundan detalles, muchas veces exactos, siempre ingeniosos; su sistema, lleno de notaciones variadas, cargado de distinciones verbales, dividido y subdividido en el infinito, no deja al espíritu la impresión de unidad y de simplicidad, marcada por obras definitivas". Así pues, vemos que este gran intelectual, será una de las personas más respetadas dentro del mundo científico.

Finalmente, se ha producido una última revolución que ha sido la **revolución digital**, de la cual podemos decir, que formamos parte de ella, aunque sea de una manera más o menos tangencial. Este cambio se produjo con la invención y la posterior progresión de la computadora digital, y ha concluido con el acceso de una gran parte de la sociedad a redes de alta velocidad y a procesadores muy potentes. Los intelectuales clave en esta revolución fueron Turing y Weiner. En el caso del primero, unió la lógica y la computación que ayudó que posteriormente, Weiner fundara la llamada ciencia cibernética a mediados del siglo XX.

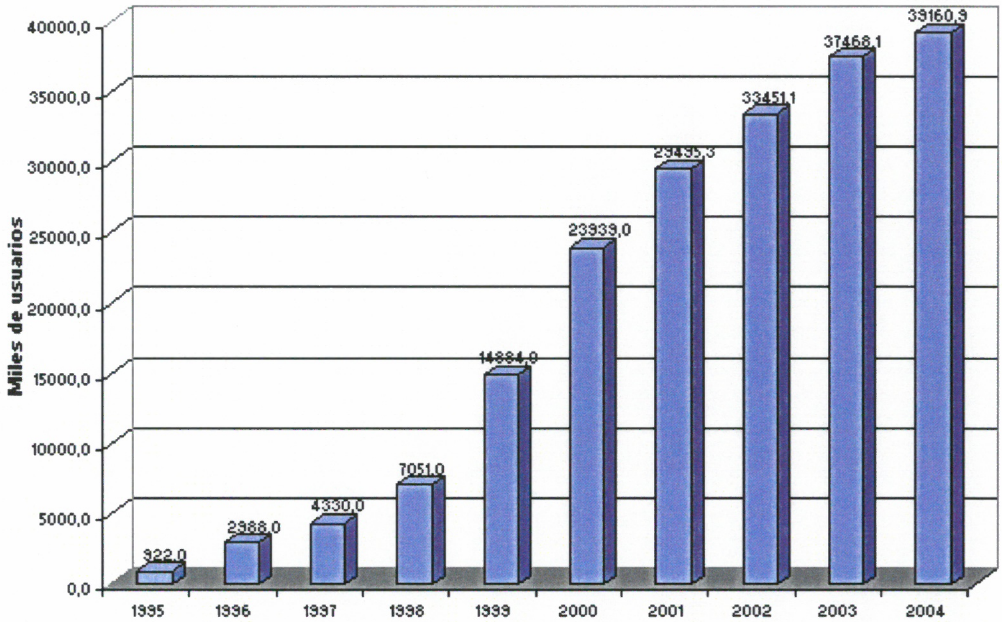
2.4 Valoración personal

¿Podemos atrevernos a augurar una próxima evolución drástica de la lógica? Teniendo en cuenta las necesidades del actual modelo social, en el cual la tecnología y la informática son los puntos clave, podemos prever que la siguiente revolución lógica será la asimilación práctica de las matemáticas y la computación dentro de la lógica, es decir, lo más importante en las próximas computadoras, será que éstas exploten la información inteligentemente, lo que significa intentar hacer máquinas que piensen inteligentemente[1]. Podríamos decir que ya estamos en esta revolución, dado que si miramos estadísticas de las cantidades invertidas por los gobiernos en la investigación para hacer robots inteligentes, que desarrollen funciones lógicas, nos daríamos cuenta de que esta evolución ya ha empezado, dado que en las principales potencias mundiales, los miembros de la llamada triada, los Estados Unidos, Japón y la Unión Europea, se destinan grandes cantidades de su capital en la financiación de investigaciones en el campo de la robótica. No obstante, esta revolución no ha hecho nada más que comenzar, dado que cada vez salen nuevas tecnologías que anteriormente creíamos imposibles de fabricar, y al cabo de un tiempo alguien ha conseguido patentarlo.

Un claro ejemplo de lo que estamos hablando es el uso del teléfono móvil. Si hacemos una mirada para atrás, hace unos diez años el hecho de tener un móvil era un hecho casual, poco frecuente en el global de la sociedad. Evidentemente, muchas personas hacían uso de su teléfono móvil, pero eran muy pocas en comparación con la actual. En el siguiente gráfico podemos ver de manera clarividente esta evolución en términos de miles de personas que poseen un teléfono móvil, desde el año mil novecientos noventa y cuatro, hasta el dos mil cuatro, es decir, que sólo en diez años, el números de teléfonos ha aumentado en un 4247%. Evidentemente, este es un resultado espectacular, y que seguramente pocos se hubieran podido imaginar. Así pues, tener una compañía de móviles ha resultado ser uno de los negocios más rentables de la última década. Si añadimos además que este gráfico está actualizado hasta el año dos mil cuatro, aún tenemos dos años de crecimiento, con lo que podríamos haber obtenido unos

[1] Ideas extraídas de la entrevista que mantuvimos con el Dr. Chicón. La entrevista completa se encuentra en el anexo 1

resultados mucho mayores si hubiéramos tenido un gráfico actualizado hasta el año dos mil siete. El gráfico[2] es el siguiente:



Además, y un hecho comparativo muy representativo, es la diferencia entre los primeros móviles y los actuales. Si hace cuarenta años los móviles podían llegar a pesar un kilogramo de peso, en la actualidad se fabrican con pesos insignificantes. Este es sólo un ejemplo de esta gran revolución, aunque podríamos destacar otros ejemplos como los ordenadores o la seguridad en los coches.

Así pues, y teniendo en cuenta los argumentos analizados en este primer apartado del trabajo, podemos concluir que la lógica nos deja ver una gran evolución en la historia de la humanidad, "naciendo" en las matemáticas de los griegos, pasando por el pensamiento de la Edad Media, donde a pesar de la presión de la religión, siguió habiendo pensadores que defendían el razonamiento lógico y experimental para alcanzar el conocimiento verdadero (por ejemplo Francis Bacon), y nadie podía osar a contradecir sus ideas. Sin embargo, la lucha de los intelectuales renacentistas ayudó a retomar el desarrollo no solamente de la lógica, sino también de todas las ciencias. Finalmente, y gracias a George Boole, la lógica se unió a las matemáticas, y pudieron progresar unidas, dando como resultado a la lógica computacional, y posteriormente a

[2] http://www.tecnociencia.es/especiales/ue_politica_informacion/img/graficos/grafico5.gif

los ordenadores, y a todas las tecnologías que actualmente tenemos. De esta manera, podemos observar la gran importancia de la lógica matemática en una sociedad tan informatizada como la nuestra.

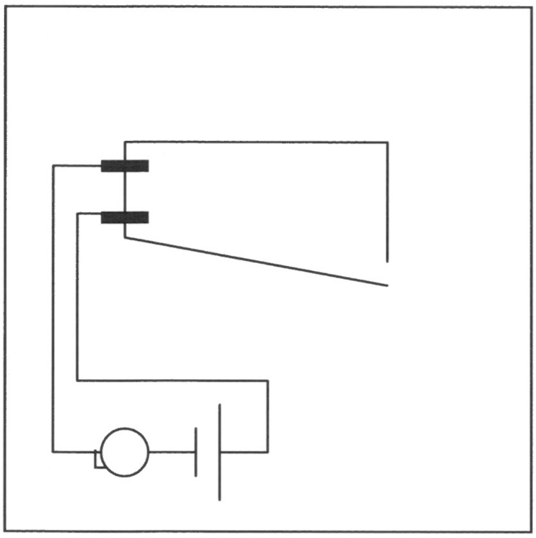
3. La lógica matemática

En este apartado presentaremos de forma breve los principios en que se basa este sistema de razonamiento. Como se verá una de sus principales ventajas es reducir los razonamientos a operaciones matemáticas simples, que pueden ser ejecutadas por una máquina (un procesador matemático).

3.1 ¿Qué es la lógica matemática?

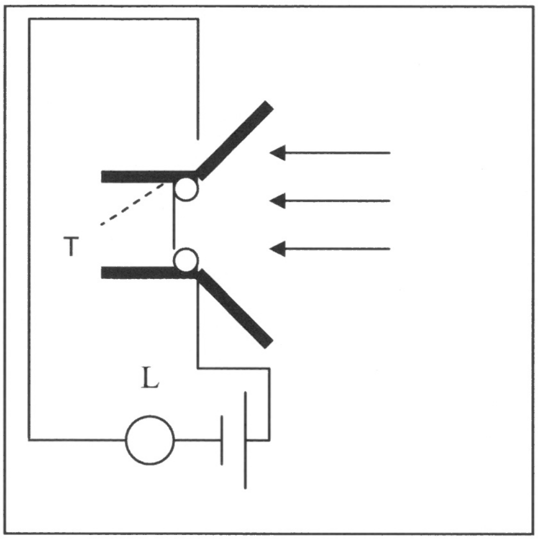
Si en el primer apartado habíamos anunciado que la lógica verbalizada estaba compuesta por conceptos (palabras) y razonamientos (frases, oraciones), la **lógica matemática** substituye todos estos conceptos y razonamientos, por variables numéricas y operaciones matemáticas. En primer lugar, se ha de decir que para construir un razonamiento partimos de conceptos previos. En lógica matemática estos conceptos se denominan variables y deben tener la cualidad de ser medibles. Por ejemplo, el concepto "llueve" puede medirse en base a los litros de precipitación caídos por metro cuadrado y hora. En tal caso este concepto podría ser tomado como una variable, que podría tomar cualquier valor numérico. Hablamos entonces de una variable analógica. La lógica matemática puede trabajar con este tipo de variables, pero son más sencillas de tratar las denominadas binarias. En este caso la variable sólo puede tomar dos valores: 0 y 1. Así, si llueve L=1 y si no lo hace L=0. Así pues, el razonamiento se transforma en una operación matemática.

El responsable de decidir qué valor toma la variable es el **sensor** o detector. Por ejemplo, un sencillo sensor para determinar el valor que toma la variable binaria "llueve" se puede observar en la siguiente imagen:



Cuando llueve la cubeta se llena de agua. El agua que llena la cubeta, cierra el circuito eléctrico y al circular la corriente se enciende la lámpara L.
La señal luminosa de la lámpara L puede ser leída, como explicaremos más adelante, por un sistema automático de control.

No obstante, se pueden analizar otro tipo de sensores, como por ejemplo, uno que nos permita medir la velocidad del viento. Este instrumento se llama anemómetro, y, en tal caso, tendríamos el valor de la variable analógica "viento".



El dispositivo en forma de embudo recoge el viento, que si tiene una velocidad suficiente hará que se levante la trampilla "T". Esta trampilla es metálica y puede girar en torno a un gozne situado en la parte superior. Cuando no hace viento, la trampilla cierra el circuito y la lámpara L estará encendida. Cuando hace viento la trampilla se levanta abriendo el circuito y la lámpara L se apaga.

3.2 Tabla de verdad

El primer concepto fundamental de este trabajo son las **tablas de verdad**. Sin embargo, este concepto no se pude explicar sin los nexos lógicos, ni viceversa. Así pues, se decidió explicar primero las tablas de verdad, dado que se creía que sería más sencillo de comprender si se empezaba a explicar primero.

El primer aspecto que se ha de explicar en este importante apartado, son los conceptos con los que trabajaremos de ahora en adelante. En primer lugar, daremos unos símbolos de veracidad para ver si una cosa es cierta o es falsa. Si una operación da como resultado que la proposición es cierta, le daremos el valor de un 1. Si por lo contrario es falso, le daremos un 0. Como en el resto del lenguaje proposicional, hay diferentes símbolos y maneras de escribir las cosas.

En este trabajo se ha escogido la variante más informática, dado que la aplicación física de la lógica matemática, y como se ha visto en la introducción, son las nuevas tecnologías. Si por lo contrario se hubiera decidido escoger la variante más matemática, se hubieran elegido los símbolos que estan explicados en cada apartado donde aparecen. De esta manera, se ha querido escoger las variables binarias para operar, aunque se hubiera podido escoger V para valores ciertos (se ha escogido 1), y F para los valores falsos (se ha escogido 0). Se ha creído oportuna esta elección, dado que resulta más sencillo operar con ceros y unos, que con letras.

Estas tablas son cuadros que representan los posibles valores lógicos de una proposición lógica. Además, es el método más conveniente para analizar los valores de certeza de una proposición lógica .Como estamos tratando con números binarios, sólo podremos encontrar en ella 0 y 1, y nunca los dos a la vez. De la misma manera que una cosa no puede ser cierta y falsa a la vez, una tabla de valores, que representa los valores lógicos de una proposición lógica, no puede tener en el mismo cuadro, un 1 y un 0. Además de contener una o varias columnas con los posibles valores lógicos, hay una última con los valores de verdad de esta proposición, resultantes de los valores de verdad de las proposiciones atómicas. Si la proposición molecular estudiada se puede descomponer en otras proposiciones moleculares más sencillas, en la tabla se obtienen columnas intermedias con los valores de verdad de estas proposiciones. En la tabla de verdad hay que poner todas las posibilidades de certeza o falsedad en forma de una tabla. Así pues, todas las reglas de certeza funcional que se utilizan para proposiciones moleculares pueden resumirse en forma de tabla. Estas tablas básicas de certeza indican rápidamente si una proposición molecular es cierta o falsa si se conoce la certeza o falsedad de las proposiciones que la forman. A continuación pondremos algunos ejemplos de tabla de verdad con los nexos lógicos previamente estudiados:

A	B	$A+B$
0	0	**0**
1	0	**1**
0	1	**1**
1	1	**1**

Por otra parte, necesitamos saber que para construir una tabla de certeza que dé todas las combinaciones posibles de asignaciones de certeza a n letras atómicas distintas, son necesarias 2^n líneas. Para cada letra atómica distinta se necesita una columna, y también se necesita una columna por cada término de enlace que se presente.

En cualquier tabla de verdad, necesitamos poner todas las posibilidades lógicas y luego analizar una por una si el resultado obtenido en cada caso, es cierto, o por si lo contrario nos da un resultado de falsedad. Aunque el resultado obtenido sea falso, no siempre lo tendremos que tener en cuenta como tal, dado que hay casos en que el resultado obtenido es empíricamente imposible. Un caso en el que lo podemos observar es el siguiente:

A	**B**	Q= $((1-A)+Q)\cdot(1\text{-}B)$
0	0	**1**
1	0	**1**
1	1	**0**
0	1	**0***

Teniendo en cuenta de que se trata de el funcionamiento de una bomba de agua, en la que A representa un sensor situado en la parte más baja de un cubo, y en el que B representa la capacidad máxima del cubo, es empíricamente imposible que haya la capacidad máxima del cubo, y sin embargo, el sensor situado en la parte más baja del mismo, no haya agua. Así pues, aunque la operación matemática sea posible, físicamente es imposible.

3.3 Nexos lógicos principales

Una reseña muy importante para entender bien este trabajo, es saber "jugar" con los números binarios en modulo 2. No estamos operando con los números de siempre, estamos trabajando con variables binarias. De esta manera, es diferente sumar uno más uno en binario, que en el sistema en el que estamos acostumbrados a operar. En el siguiente recuadro podremos observar las principales **operaciones** en números **binarios**:

0+0=0	**0·0=0**	**0-0=0**
0+1=1	**0·1=0**	**0-1=1**
1+0=1	**1·0=0**	**1-0=1**
1+1=1	**1·1=1**	**1-1=0**

En adelante, trabajaremos con variables binarias. Las relaciones entre estas variables se establecen mediante tres operaciones básicas. Estos conectivos son palabras que cambian el valor de verdad de una proposición, o sirven de enlace entre varias proposiciones. Se llaman también **conectores o nexos lógicos**. Los tres nexos lógicos más importantes son los que trabajaremos en los siguientes tres apartados, aunque, como ya se ha dicho en la introducción a la lógica, el elemento lógico "no", no es puramente un nexo lógico, dado que no enlaza dos proposiciones. Sin embargo, se considera, en lenguaje matemático, como uno de los tres nexos lógicos principales.

3.3.1 Nexo lógico "Y"

El primer conector que analizaremos será el que realiza la función de multiplicar. De esta manera, podemos deducir que el resultado sólo tendrá valor cierto, cuando todas las variables de entrada tengan valor cierto. Por ejemplo, si llueve y hace frío, cogeré el coche. Para que coja el coche, se han de cumplir las dos condiciones, que llueva y que haga frío. No obstante, no cogeré el coche si llueve pero no hace frío o si no llueve pero hace frío, o si ni llueve ni hace frío.

Esta función lógica, se puede expresar de la siguiente manera:

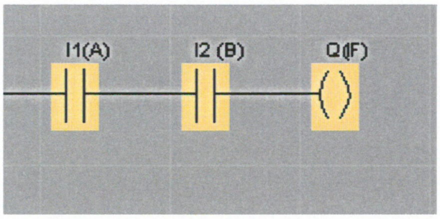
$$F= A \cdot B$$

Si queremos representar los posibles valores de salida que puede tener una función lógica deberíamos de hacerlo mediante una tabla de verdad, como se ha visto en el apartado 3.2 del trabajo. A continuación se expone la tabla de verdad correspondiente a esta operación lógico matemática.

A	B	$A \cdot B$
0	0	**0**
1	0	**0**
0	1	**0**
1	1	**1**

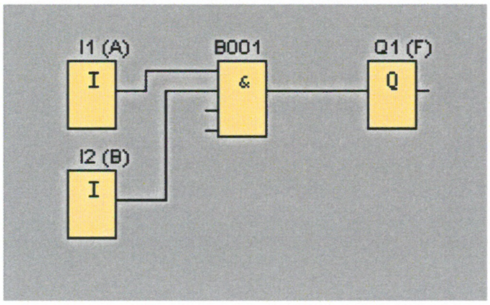
Podemos observar que F tendrá valor cierto, sólo cuando A y B tengan valor cierto (1). Por otra parte, existen otros conceptos que caracterizan esta función lógica, como son los circuitos eléctricos, y el circuito de nexos lógicos.

En primer lugar, y aunque posteriormente ampliaremos este concepto en el apartado 3.4, un circuito eléctrico nos ayuda a experimentar con interruptores electromagnéticos, para simular nuestra función. En el caso del nexo lógico "I", el interruptor electromagnético que lo caracteriza es el contacto abierto y ha de estar siempre en serie.



Así pues, vemos que si A y B tienen valor cierto, los contacto que en principio estaban abiertos se cierran, y la corriente puede pasar hasta Q, donde hará que F tenga valor 1. Pero si sólo se cierra A, la corriente no podrá llegar hasta Q, con lo que F valdrá 0. De la misma manera, si sólo B tiene valor 1, la corriente no llegará a B, y F tendrá valor 0.

Por otra parte, el circuito con nexos lógicos de esta función sería el siguiente:



En este caso, A y B son las entradas que podemos observar que están conectadas con un conector & o "Y". Podemos leer el mismo concepto anteriormente planteado, Q tendrá valor 1, cuando A y B den la señal al conector lógico, que están conectados. Si uno no lo está, la corriente no pasará por este circuito.

3.3.2 Nexo lógico "O"

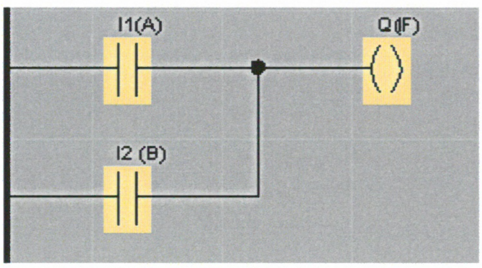
En referencia a este nexo lógico, deberíamos decir que si el anterior representaba la función de multiplicar, este realiza la función de sumar. De esta manera, habrán muchos más valores ciertos en la tabla de verdad, dado que no está tan condicionado por la otra entrada (una entrada es una de las partes de la operación lógica que influye en ella). Por ejemplo, si llueve o hace frío, cogeré el coche. Para que coja el coche sólo tendrá que cumplirse una de las dos funciones, pero nunca las dos. Así pues, cogeré el coche si llueve pero no hace frío, o si hace frío pero no llueve. No obstante, no lo cogeré si ni hace frío ni llueve, o si hace frío y llueve. Para entender mejor este segundo operador lógico, deberíamos de representar su operación matemática, y posteriormente, su tabla de verdad.

$$F = A + B$$

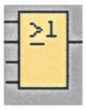
Su tabla de verdad sería la siguiente:

A	B	$A+B$
0	0	**0**
1	0	**1**
0	1	**1**
1	1	**1**

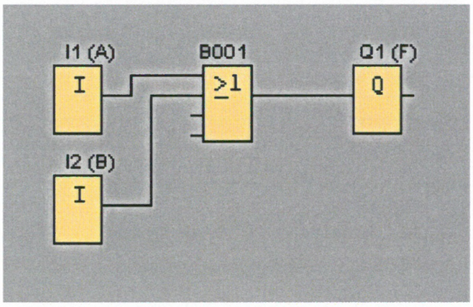
Así pues, y como ya se ha anunciado anteriormente F valdrá 1 (tendrá valor cierto) sólo cuando como mínimo una de las dos entradas tenga valor cierto. En referencia a su circuito eléctrico y su circuito con nexos lógicos, el vínculo lógico “O” se representa de la siguiente manera. En primer lugar, su circuito eléctrico se representa mediante dos contactos abiertos en paralelo unidos a una bobina. De esta manera, sólo se necesita que uno de los dos contactos se cierre para que pase la corriente a la bobina y ésta funcione.



Por otra parte, su circuito de enlaces lógicos se representa mediante dos entradas, A y B unidas a una función "O", que se representa mediante un símbolo que en tautología, que analizaremos más adelante, es el siguiente:



De la misma manera, este bloque está unido a una salida Q, que representa la bobina.



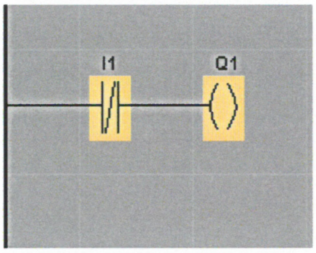
3.3.3 Operación lógica "NO"

Este último término de enlace que analizaremos en este apartado, representa la función de negación. Es importante decir, que esta operación actúa sobre una sola variable. Si anteriormente podíamos unir mediante un nexo lógico dos variables, en este caso no es posible, dado que esta función resulta sobre una sola. De esta manera el resultado en una operación lógica, sólo tendrá valor cierto cuando la proposición atómica que esta negada, tenga valor falso. Este ejemplo lo veremos mejor en una tabla de verdad. Antes de hacer la tabla de verdad de este nexo lógico, necesitamos saber que función matemática representa. Si anteriormente teníamos que el término de enlace "Y" tenía la operación matemática "*" y el nexo lógico "O" la suma, en este caso la negación tendrá el valor del complementario, es decir, (1-...). Esta operación se puede representar también, poniendo una especie de apóstrofe al lado de la letra, es decir, A'.Al tener esta operación, vemos que sólo tendrá valor cierto cuando la variable que esta negada tenga valor falso, es decir 0, dado que de esta manera la resta no dará como resultado 0. No obstante, esto lo veremos mejor en la siguiente tabla de certeza:

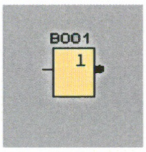
B	$1-B$
0	**1**
1	**0**

Podemos observar en la tabla de verdad, que el resultado obtenido después de realizar la operación lógica, es el contrario al que habíamos dado. Es decir, en el primer caso, de tener una falso tenemos cierto, y en el segundo caso viceversa.

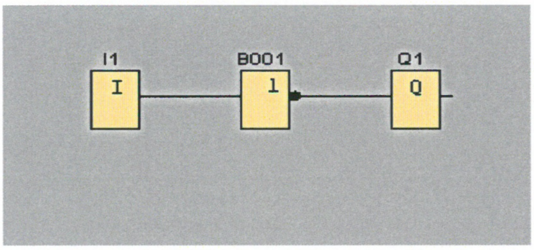
Por otra parte, y en lo referente a circuitos eléctricos y circuitos con nexos lógicos, este enlace de negación, representa en primer lugar, y en circuito eléctrico un contacto cerrado, donde la corriente pasará hasta que la entrada I1 tenga valor cierto. Como ya se ha analizado anteriormente, cuando esto suceda el contacto I1 se abrirá, y no dejará pasar la corriente por donde antes si que dejaba. El circuito eléctrico de esta conexión lógica es el siguiente:



En segundo lugar, su circuito con nexos lógicos se representa mediante una entrada I1, conectada a un nexo lógico negado, es decir, y como ya veremos más adelante en el trabajo cuando analicemos los símbolos de cada conector lógico, que se representa de la siguiente manera:



Además, este conector lógico unido a una entrada I1, están unidos obviamente a una salida Q1, que sólo tendrá valor cierto en el caso en que el nexo lógico "NO" se mantenga cerrado, y la entrada I1 tenga valor falso. Su circuito con nexos lógicos es el siguiente:

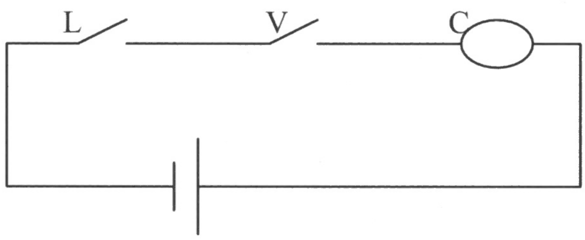


A partir de estas operaciones básicas, la suma, la resta y el complementario se pueden definir otros nexos más complejos, como el conector condicional (si...entonces...) o las puertas lógicas XOR, XAND entre otras, aunque las explicaremos más adelante.

Un hecho a tener en cuenta, es que la decisión de qué debemos hacer podrá ser tomada de forma automática por una máquina de calcular. Cada una de estas ecuaciones constituye un razonamiento completo, que denominamos programa. El sistema capaz de ejecutar un programa es un procesador matemático.

3.4 Circuitos eléctricos

Un **circuito eléctrico** elemental formado por interruptores puede actuar como un primitivo procesador matemático. Por ejemplo, el programa $L \cdot V = C$ puede ejecutarse por el siguiente circuito:

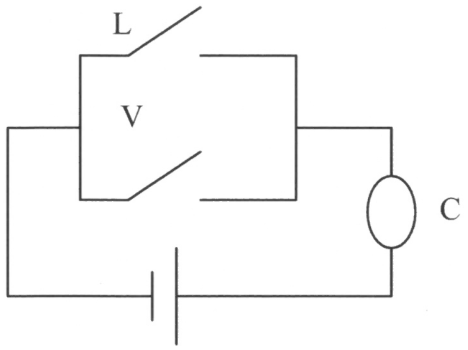


Circuito 1

Si se cierra el interruptor L y también el V, entonces se enciende la lámpara C, que nos indica que debemos coger el coche. Todos los resultados posibles que podemos obtener en la ejecución de este programa se pueden mostrar en forma de una tabla, que denominamos tabla de verdad o tautología.

L	V	$L \cdot V$
0	0	**0**
1	0	**0**
0	1	**0**
1	1	**1**

Veamos qué ocurriría en el caso del segundo programa: L + V = C



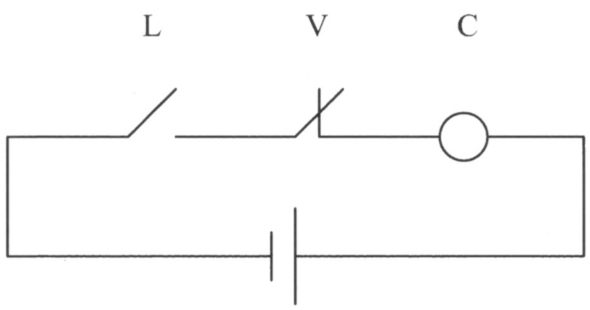
Circuito 2

Es necesario tan solo que ocurra una de las dos variables para que la lámpara C se encienda, es decir, para que tomemos la decisión de ir en coche. La tabla de verdad correspondiente a este segundo programa sería:

L	V	**C = L + V**
0	0	**0**
1	0	**1**
0	1	**1**
1	1	**1**

Fijémonos en la última fila de la tabla anterior, la suma lógica 1+1 es igual a 1. El número 2 no existe, pues trabajamos en binario.

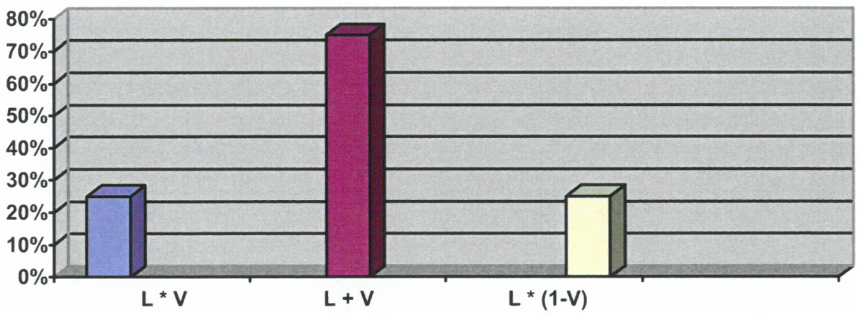
Finalmente, y cogiendo un simple ejemplo de una operación lógica, observamos el último nexo lógico indispensable: el "no".



Circuito 3

L	V	$C = L \cdot (1 - V)$
0	0	**0**
1	0	**1**
0	1	**0**
1	1	**0**

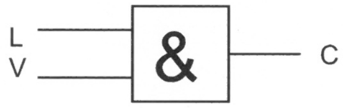
Así pues, y analizando detalladamente las tres tablas de verdad, podemos ver que en el segundo ejemplo, es decir en el… o…, es donde hay más posibilidades de que se encienda la bombilla, hay un 75% de posibilidades de que se encienda. No obstante, es un resultado lógico, dado que es el que menos condicionado está por las otras variables. En el primer caso, tenían que ser las dos variables verdaderas para que se encendiera la bombilla, y este caso, solo pasaba una vez, cuando ambas variables tenían valor 1. En la segunda tabla, observamos el resultado inverso al primero, es decir, que hay un 75% de posibilidades de que se encienda la bombilla. Solo no se enciende cuando las dos variables tienen valor falso. Finalmente, y en la tercera tabla, observamos el mismo resultado que en la primera, solo hay un 25% de posibilidades de encenderse. No obstante, y aunque haya el mismo número de posibilidades en el primer caso y en el últimos, los resultados son completamente distintos, dado que en un caso se encendería solo cuando las dos variables tendrían valor verdadero, mientras que en el último caso sólo se encendería cuando la variable L tuviera valor positivo y la variable V negativo. Podemos observar los resultados obtenidos en el siguiente gráfico, donde en el eje de las X tenemos representado los dos lógicos estudiados anteriormente ("Y" y "O") y la operación lógica "No". Por otra parte, en el eje de las Y se representan las posibilidades de que la bombilla se encienda.



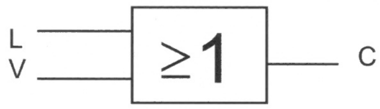
3.5 Simbología en puertas lógicas

Las operaciones básicas que pueden realizarse mediante el circuito eléctrico, se pueden representar también mediante unos símbolos, denominados puertas lógicas.

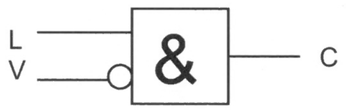
El circuito serie se asocia a la función lógica "Y" (AND=&). Para que la lámpara L del circuito 1 se encienda, es necesario que estén cerrados a la vez los interruptores L **y** V. En **simbología DIN** se representa de la siguiente manera:



El circuito paralelo se asocia a la función lógica "O". Para que la lámpara L del segundo circuito se encienda, basta con que uno de los dos interruptores esté cerrado, el L **o** el V:



Finalmente, para indicar que un contacto se encuentra cerrado, como es el caso del contacto V en el tercer circuito eléctrico, usamos la siguiente simbología:



Con ello indicamos que, si no actuamos sobre el interruptor V, éste está cerrado. Cuando pulsamos el interruptor V, abre. Este interruptor también se denomina "negado", porque realiza la acción inversa de los demás interruptores.

3.6 Autómatas

Un **autómata** es un complejo sistema de computación digital. Como tal, todo autómata tiene unas características esenciales. Tiene un mecanismo para leer entradas. Asumiremos que la entrada es una palabra dentro de un alfabeto cualquiera escrito en un archivo de entrada, que el autómata puede leer pero en ningún caso cambiar. El archivo de entrada se divide en dos celdas, donde cada una pueda sostener un símbolo. El mecanismo de entrada puede leer el archivo de izquierda a derecha, cada vez un símbolo. Además, puede detectar el final de la palabra de entrada (mediante un procedimiento de final de línea). El autómata puede producir salidas de muchas maneras: archivos, órdenes, ejecuciones. El mismo, ha de tener un sistema temporal de almacenamiento, que consista en un ilimitado número de celdas, como ya se ha dicho anteriormente capaz de guardar un simple símbolo de un alfabeto (que no debe necesariamente ser el mismo que el del archivo de entrada). El autómata puede leer y cambiar los contenidos de las celdas. No obstante, debemos recordar que el autómata no puede cambiar ni modificar el contenido de la entrada, pero si que puede en cambio, modificar el contenido de las celdas. Sería un error pensar que la modificación que puede realizar el autómata es la misma en la entrada que en las celdas. Aunque, se ha de recordar también, que pueden tener el mismo valor. Finalmente, el autómata tiene un sistema de control, que puede estar en cualquiera de los finitos números en los estados internos, y que puede cambiar el estado de alguna manera.

3.7 Práctica de los conceptos explicados anteriormente

En este apartado del trabajo, intentaremos explicar los tres nexos lógicos anteriormente explicados, el "Y", el "NO" y el "O", a partir de dos ejemplos. En primer lugar, explicaremos el "Y" y el "NO" con el ejemplo de una persiana. Posteriormente, haremos una reproducción de una bomba de agua que llena un depósito para caracterizar el nexo lógico "O".

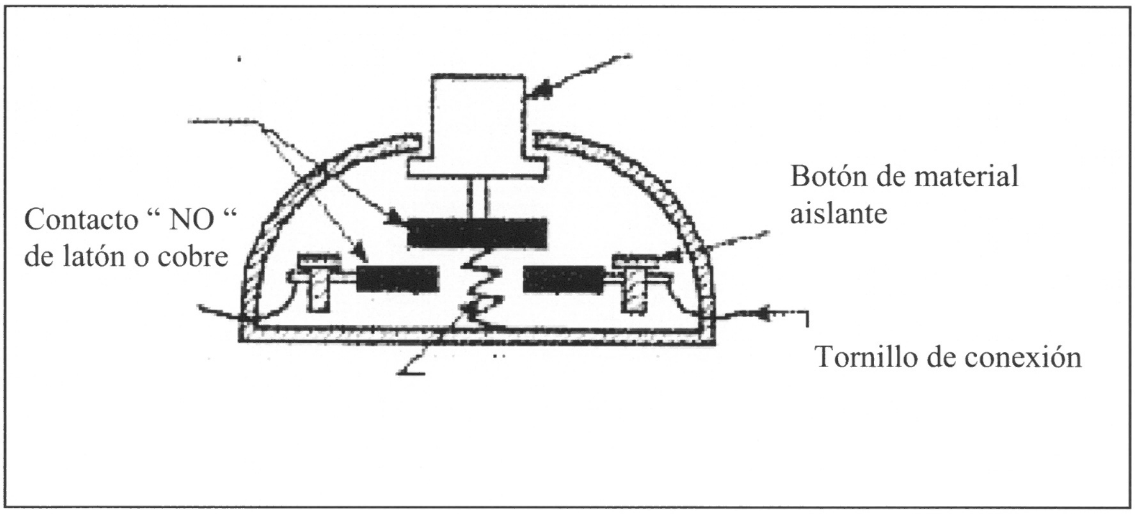
3.7.1 Persiana

En este primer apartado práctico deberíamos tener en cuenta cuatro funciones principales. En primer lugar, que una persiana enrollable se acciona mediante un motor, que hará, o bien subir, o bien bajar la **persiana**[3]. En segundo lugar, que una persiana es controlada por dos pulsadores, uno que representará que la persiana suba, y el otro que ésta baje. Por otra parte, cuando la persiana llega tanto al extremo superior máximo, como al extremo inferior máximo, unos finales de carrera desactivan el motor de ésta. Finalmente, y en lo que se refiere a un tema de la seguridad del usuario, se deberían tener en cuenta que si el usuario apretara erróneamente los dos pulsadores, tanto el de

[3] Fotografías y explicación practica de la persiana en el anexo número 2

subida como el de bajada, no pasase nada. Si no existiera este enclavamiento de seguridad, se correría el riesgo de que se produjera un corto-circuito. A continuación, explicaremos dos términos que han aparecido anteriormente, y que son importantes para el buen entendimiento de esta práctica: el final de carrera y el pulsador.

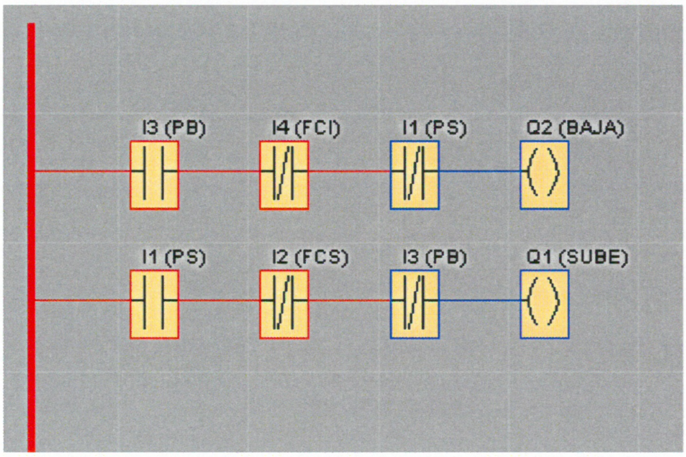
Un **final de carrera** es un pulsador que es accionado por la persiana y que sirve para detectar las posiciones extremas de ésta. Así pues, una persiana necesita de un final de carrera, para que cuando ésta llegue arriba o a abajo, el motor se pare. Si no fuera así, y por lo tanto no hubiera finales de carrera, el motor seguiría funcionando, pero la persiana se descarrilaría dado que ya habría llegado a su tope máximo o mínimo. Por lo tanto, el final de carrera, tanto el inferior, como el superior, son básicos para el buen funcionamiento de una persiana. Por otra parte, un **pulsador** es un elemento de entrada que recoge una orden del usuario. Una tecla del ordenador, o el timbre de casa son ejemplos de pulsadores. Por ejemplo, se representa en la siguiente imagen, un pulsador de un timbre para una casa:



Se puede observar que un timbre de una casa es un objeto relativamente sencillo, que “contiene” una operación lógica anteriormente utilizada, “no”. Además, consta de un botón de material aislante y de un tornillo de conexión, entre otras herramientas. Sin embargo, se quería resaltar que los timbres contienen la operación lógica “no”.

Anteriormente, habíamos estudiado que era la tabla de verdad, y habíamos puesto algunos ejemplos. En este caso, esta tabla de verdad se debería de leer de la siguiente manera: si pulsamos PS (es decir, el pulsador que da la orden de que la persiana suba) y la persiana no es arriba de todo, y no estamos pulsando PB (pulsador para que la persiana baje), entonces esta subirá. Por otra parte, si estamos apretando PB y la persiana no está debajo de todo, y no estamos pulsando PS, entonces ésta bajará[4].

Así pues, la persiana subirá si el pulsador PS está apretado sin ninguna otra condición, o en el caso de que tanto el pulsador de subida como el final de carrera inferior estén apretados. Por otra parte, el motor de la persiana accionará la función bajar, si como en el caso anterior, el pulsador de bajar esté apretado, o que tanto el pulsador PS como el final de carrera superior estén apretados sin ninguna otra condición. Por lo tanto, hay un 25% de probabilidades para que la persiana suba, hecho lógico debido a la gran cantidad de variables de las que depende. Si anteriormente habíamos explicado la práctica mediante la tabla de la verdad, ahora deberíamos explicarla mediante un circuito eléctrico, que se puede ver en la siguiente imagen:



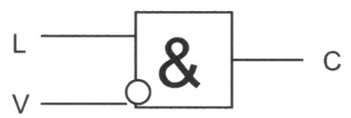
Para entender esta imagen, deberíamos explicar qué significa cada concepto. En primer lugar, y a lo que se refiere al circuito eléctrico, I1 e I3 representan contactos abiertos, es decir que no están negados, y que se cerraran si los activamos. Por ejemplo, si apretamos el pulsador de subida, PS se cerrará y la corriente podrá llegar al motor de la persiana. De esta manera, si ni PS ni PB se aprietan, la corriente no pasará, y como se ha visto en la tabla de verdad, el motor no se pondrá en marcha. Por otra parte, I2, I4, I5 e I6 representan contactos cerrados, es decir, negados. Estos contactos tienen la característica de que, al contrario de los contactos abiertos, si se detecta que están apretados, entonces el contacto se abrirá, y no dejará pasar la corriente. I5 e I6 son

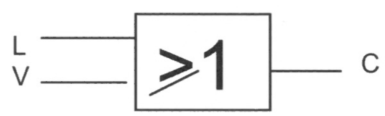
[4] Podemos ver su tabla de verdad en el anexo 3

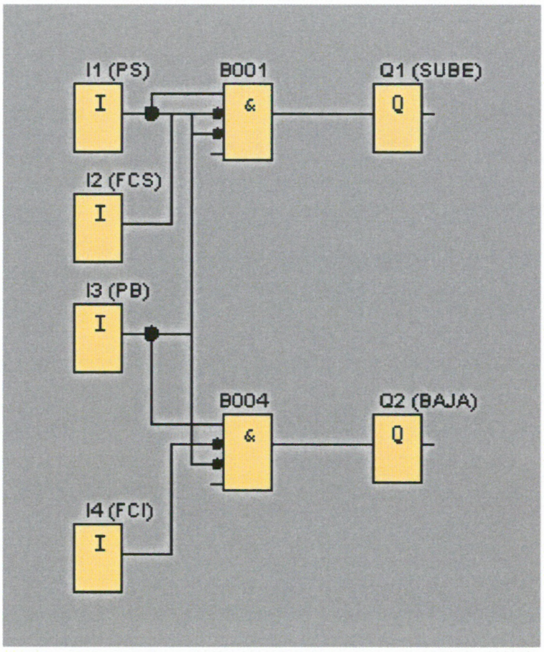
enclavamientos de seguridad. Esto permite que si debido a un error del usuario, se aprieten PS y PB a la vez, que el programa no responda. De lo contrario se podría producir un cortocircuito y dañar la salud del usuario.

Por otra parte, explicaremos los conceptos que aparecen en el siguiente circuito con nexos lógicos. En primer lugar, se ha de decir que este circuito representa la misma función que el anterior circuito. Representa el motor de la persiana (Q1). En este circuito podemos observar dos nexos lógicos que anteriormente habíamos explicado, el "O" y el "No".

Como anteriormente habíamos explicado, el primer nexo (primera imagen situada debajo de este texto) representa un contacto cerrado ("O"), mientras que el segundo representa el "No". De esta manera, si PS (elemento de entrada para que la persiana suba) y no FCS(final de carrera superior) ni PB(elemento de entrada que tiene la función de bajar la persiana si está activado), o PB y no FCI(final de carrera inferior) ni PS, entonces el motor de la persiana funcionará.







Circuito con nexos lógicos

Para esta práctica, se ha tenido que aprender a utilizar un simulador de autómatas, el *Step 5*, aunque estas imágenes correspondan al otro programa utilizado para realizar otras prácticas, visualmente mejor, más bonito, llamado *LogoSoft Confort*. Este programa, nos ha ayudado a hacer la simulación de la persiana como se ha podido observar en las imágenes anteriores.

Para implementar experimentalmente este ejemplo, se ha construido una maqueta de una persiana. En la representación siguiente puede verse el cableado de la maqueta, desde las entradas (pulsadores y finales de carrera) al autómata (explicaremos el funcionamiento del autómata en el siguiente apartado); y desde éste a la salida (motor conectado con polaridad +/- para subir y motor conectado con polaridad -/+ para bajar).

3.7.2 Puerta de garaje

Aprovechando la práctica de la persiana decidimos realizar una **puerta de garaje**[5], donde la única diferencia es el programa que se ha desarrollado para cada uno. Esta práctica es la persiana aunque implementada con un contador de tiempo. Como todos sabemos, una puerta de garaje consiste en una estructura metálica que se abre o se cierra para que un automóvil pueda entrar o salir. Pues bien, la única diferencia

[5] Ver anexo 4 para ver imágenes de la práctica

existente con una persiana, es un contador que evita que el eje de la puerta se rompa. De esta manera, y debido al gran peso de la puerta metálica, se instalan unos contadores de tiempo, que transcurrido ese tiempo, lo que hace es hacer la función contraria a la que se estaba realizando. De esta manera, si estaba subiendo y apretamos para que la puerta se cierre rápidamente, lo que estamos diciendo a la puerta es que después de unos segundos, esta inicie su sistema de cierre.

Además, y otra característica de la puerta del garaje, y es ésta, una de las razones que se ha tenido en cuenta a la hora de realizarla, es que necesitábamos un sensor fotoeléctrico que lo que hace es que si pasamos la mano delante suyo, lo que hará será pararse y realizar la función contraria a la que estaba realizando. Este es un sistema de seguridad que se encuentra no sólo en las puertas de garaje, sino que también en ascensores.

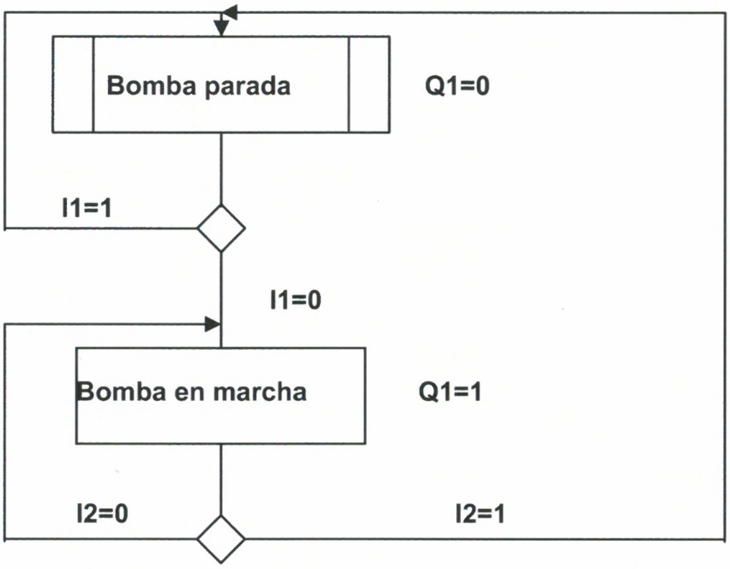
3.7.3 Regulación por intermitencia

En este segundo apartado de pequeños ejemplos experimentales, reproduciremos la función de una bomba de agua que ha de llenar un depósito. A este proceso se lo denomina **regulación por intermitencia**. La bomba[6] se debe poner en marcha cuando el nivel del agua en el depósito está justo por debajo del mínimo y no debe detenerse hasta que esté justo por encima del máximo[7].

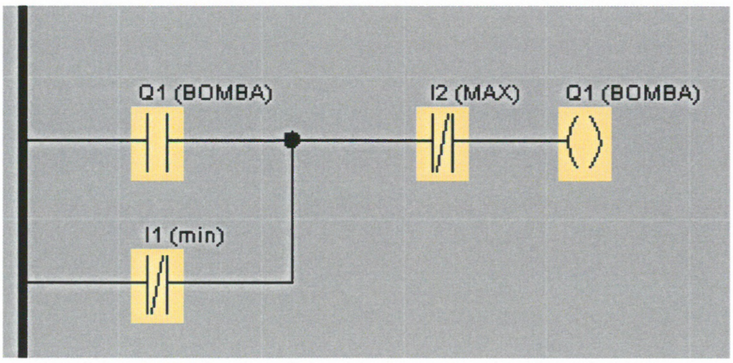
Con este tipo de regulación garantizamos que las reservas del depósito siempre se encontraran por encima del mínimo y que el funcionamiento de la bomba se producirá en intervalos suficientemente largos (sin arranques y paradas muy continuadas, que resultarían perjudiciales). Vamos a introducir una representación esquemática, denominada diagrama de flujo, para representar esta maniobra. El diagrama de flujo muestra unos “estados”, que se caracterizan por el valor que en cada uno de ellos toma la salida que estamos controlando. En nuestro caso existen dos estados: bomba parada (Q1=0) y bomba en marcha. El paso de un estado a otro se denomina “transición”. Las transiciones tienen lugar como consecuencia del cambio en el valor de una entrada concreta. Por ejemplo la transición “bomba parada” a “bomba en marcha” tiene lugar cuando la entrada I1 (nivel mínimo en el depósito) pasa del valor 1

[6] Ver anexo 5 para ver las imágenes de la práctica
[7] Ver annexo 6 para ver el esquema de la practica

(que indica que el agua se encuentra por encima de este nivel), al valor 0 (cuando el agua está por debajo del mismo). Denominamos "bucle" a un ciclo de programa en que se repite de forma continua un mismo estado: por ejemplo, mientras I1=1 el sistema de control permanece en el bucle bomba parada. El diagrama de esta práctica es el siguiente:



Por otra parte, y para comprender mejor la práctica que estamos realizando en este apartado del trabajo, se ha de analizar el circuito eléctrico del mismo. Si el agua no llega al mínimo y no llega al máximo, la bomba se pone en marcha. Cuando el agua llegue al mínimo, la bomba continuará en marcha, puesto que se ha dispuesto un contacto en paralelo de Q1 (a ello lo denominamos "realimentación" de la maniobra). La bomba seguirá en marcha hasta que el agua llegue al máximo. Por otra parte, también deberíamos analizar el circuito con nexos lógicos.



De esta manera tenemos dos contactos en paralelo (Q1 y I1), que representan un contacto en serie con una segunda entrada (I2). Todo este diagrama de contactos, se junta a la salida, la bomba de agua. Este programa debería ser analizado de la siguiente manera: si la bomba de agua está encendida o el agua no está en el mínimo, y, tampoco está en el máximo, entonces la bomba deberá seguir encendida. Esta operación de llamar a la misma se llama recursividad.

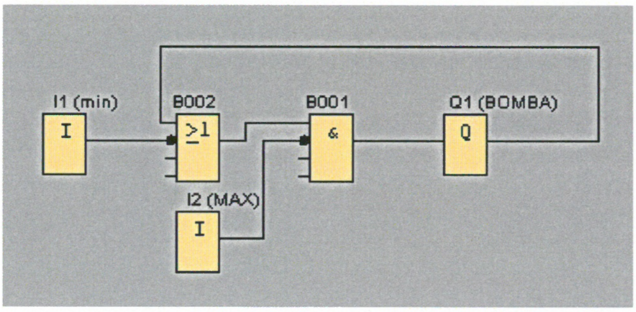
Por otra parte, la operación matemática equivalente es la siguiente:

$$((1\text{-}I1) + Q1) \cdot (1\text{-}I2) = Q1$$

, y su tabla de verdad será la siguiente. Podemos observar una peculiaridad en esta tabla, dado que en el último caso tenemos una situación matemáticamente posible, pero físicamente imposible, dado que no puede ser que haya agua en el nivel máximo y no haya agua en el nivel mínimo. De esta manera, la tabla de verdad que corresponde a esta práctica es la siguiente, aunque se ha de tener en cuenta que el último caso no es relevante, por las razones previamente analizadas.

I1	**I2**	$Q = ((1 - I1) + Q) \cdot (1 - I2)$
0	0	**1**
1	0	**1**
1	1	**0**
0	1	**0**

Finalmente, se ha de analizar el circuito de nexos lógicos correspondiente a esta práctica:



En este caso se puede observar dos elementos de entrada, Q1 y I1, unidos por un nexo lógico "O" (≥ 1). Además, vemos que I1 es un elemento de entrada negado, dado que se puede ver una bola al engancharse con el nexo lógico "O". Por otra parte, vemos que estos dos elementos de entrada están situados en paralelo con otra entrada I2, que a la vez es un contacto negado (bola cuando se une con el nexo lógico "Y"). Finalmente, todo esto está unido a la salida Q, que a la vez representa un elemento de entrada. Este concepto se llama recursividad.

En referencia a este apartado del trabajo de investigación, podríamos detallar muchos más conceptos interesantes. Sin embargo, creemos que es mejor anexar esta información, dado que no está directamente relacionada con la lógica matemática[8].

3.7.4 "Scalextric calculadora"

Esta es quizás la práctica más compleja, aunque a la vez más interesante que se ha realizado a lo largo del trabajo. Dentro de las tres prácticas realizadas con anterioridad, la persiana, la puerta de garaje y la bomba de agua, esta es quizás la más compleja. Se dice que es la más compleja debido a que el programa en ordenador que se ha tenido que programar, contiene muchos conceptos que no conocíamos y se ha tenido que aprender como funcionaban.

Este programa consiste en hacer un **scalextric** que pueda realizar la función de **calculadora**. De esta manera, podemos hacer que cualquier aparato que creamos oportuno pueda realizar funciones para las que teóricamente no estaba preparado. En este caso, el programa consiste en apretar un botón y realizar las vueltas que se crean oportunas. Nuestro programa sólo contará vueltas completas, aunque si hubiéramos querido complicar aún más el programa, hubiéramos podido poner más barreras fotoeléctricas en más puntos de nuestro circuito.

En segundo lugar, apretaremos un segundo botón, con el que podremos realizar más vueltas. Finalmente, apretaremos el botón de sumar y después de apretar el botón para que empiece a hacer vueltas, el coche hará el total de vueltas que corresponde a la suma de los dos primeros pasos.

Hubiéramos podido realizar más funciones como la de restar o multiplicar, pero lo único que cambiaba era el código del programa. Es por esta razón que no se creyó portuno, dado que no realizaba ninguna otra función especial interesante[9].

[8] Ver anexo 7
[9] Ver anexo 8 para ver imágenes de la practica

3.8 Puertas lógicas derivadas

Como ya se ha citado anteriormente, existen tres nexos lógicos fundamentales: el "Y" (AND en inglés), el "NO" (NOT en inglés) y el "O" (OR en inglés). No obstante, hay otros nexos importantes como el "XOR", el "NAND" y el "XNOR". Todos ellos derivan de la combinación de los tres nexos lógicos principales.

3.8.1 Puerta lógica derivada "XOR"

En primer lugar explicaremos el "XOR". Esta puerta viene a decir que el resultado de la operación será cierto si A o B son ciertas, pero que nunca las dos a la vez. Esta puerta, también se la llama "O" exclusivo por la razón que acabamos de explicar. Un ejemplo para entender mejor la función "XOR", sería dos interruptores conmutados. Su operación matemática característica es la siguiente:

$$F = A \oplus B$$

No obstante, también la podríamos representar de la siguiente manera:

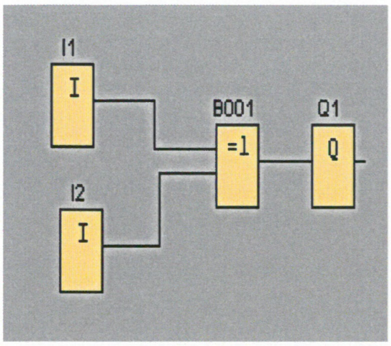
F=A'B + AB'= $((1-A)\cdot B)+(A\cdot(1-B))$

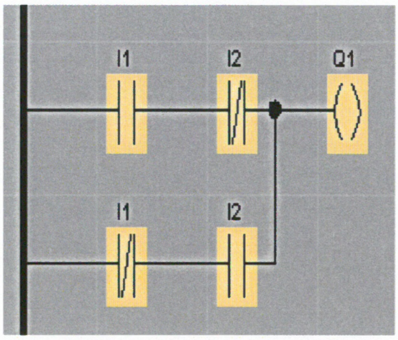
Por otra parte, su tabla de verdad sería:

A	B	Q= $((1-A)\cdot B)+(A\cdot(1-B))$
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

Así pues vemos lo que se había dicho anteriormente, de que Q solo valdrá 1, es decir, tendría valor cierto, cuando A o B tuvieran uno de ellos valor cierto o 1, pero no valdría cuando tengan tanto A como B valor cierto.

Por otra parte, se analizará los esquemas eléctricos y el circuito con nexos lógicos de esta puerta lógica derivada. En el circuito con nexos lógicos, observamos una nueva puerta lógica, la del XOR o "o" exclusivo, que conecta dos entradas a la salida. Por otra parte, se puede observar en el circuito eléctrico, dos contactos en paralelo. Los dos circuitos están representados en las siguientes imágenes:





Al pulsar una de las dos entradas, cualquiera de las dos, la salida cambia de estado, sea cual fuere aquél en que se encontraba.

3.8.2 Puerta lógica derivada "NOR"

En segundo lugar, el "NOR" quiere decir que si NO A y NO B, entonces el resultado de la operación matemática tendrá valor cierto. Esta función resulta de negar A y B. Así pues, su función matemática será la siguiente:

$$F = \overline{A + B} = \overline{A} \times \overline{B}$$

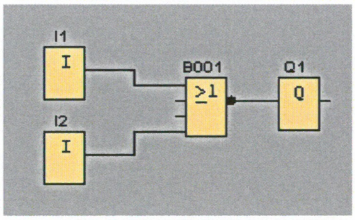
De la misma manera que anteriormente se ha escrito la representación oficial matemática, y luego la se ha transformando en nexo lógicos básicos, ahora haremos lo mismo:

$$F = 1 - (A + B)$$

Así pues, su tabla de verdad será la siguiente:

A	B	$F = 1 - (A + B)$
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**0**

En este caso, no podremos representar todavía el circuito eléctrico de la función, dado que aún no se ha explicado que es una memoria interna. Sin embargo, si que podemos representar el circuito con nexos lógicos:



3.8.3 Puerta lógica derivada "NAND"

En tercer lugar, encontramos la puerta lógica "NAND". Las características principales de ésta, son que realiza la función de un producto lógico negado. Por lo tanto, si no A o no B, entonces la operación matemática tendrá valor cierto. De esta manera, y antes de hacer la tabla de verdad, ya podemos ver que la única condición para que esta operación tenga como resultado negativo, falso (0), será que tanto A como B sean 1. Su operación matemática es la siguiente:

$$F = \overline{AB} = \overline{A} + \overline{B}$$

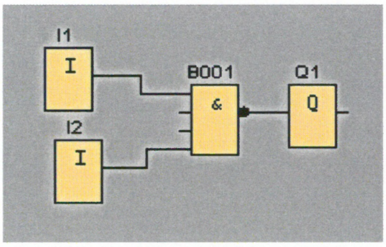
$$\downarrow$$

$$F = (1 - A) + (1 - B)$$

Como podemos observar en cada caso, tanto en el "XOR", "NOR", "NAND", y podremos observar en el "XNOR", todas las operaciones provienen de las tres principales: A+B, A·B y (1-A). De esta manera, en este caso encontraremos la última, que es la función complementaria, la que niega una variable. Todas estas cosas anteriormente explicadas, las podemos ver reflejadas en la siguiente tabla de verdad:

A	**B**	$F = (1 - A) + (1 - B)$
0	0	**1**
0	1	**1**
1	0	**1**
1	1	**0**

En lo referente a los circuitos lógicos, y electrónico, en este caso tampoco expondremos el segundo, dado que comportaría un problema a la hora de entender este apartado. Posteriormente, explicaremos que es una memoria interna, y volveremos a plantear la cuestión para que quede claro.



3.8.4 Puerta lógica derivada "XNOR"

Finalmente, encontramos la puerta lógica de equivalencia o "XNOR". Esta función representa la siguiente operación matemática:

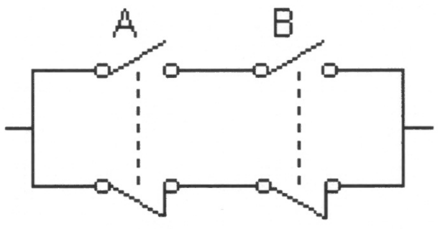
$$F = \overline{A \oplus B}$$

$$Q = (A \cdot B) + ((1 - A) \cdot (1 - B))$$

Por lo tanto, si A y B tienen valor verdadero, es decir, 1, o no A y no B, entonces, la operación dará como resultado 1. Teniendo en cuenta esta última operación matemática, podemos representar su tabla de verdad:

A	**B**	$Q = (A \cdot B) + ((1 - A) \cdot (1 - B))$
0	0	**1**
0	1	**0**
1	0	**0**
1	1	**1**

Así pues, podemos concluir que sólo Q tendrá valor cierto, cuando las dos variables tengan el mismo valor, ya sea 1 o 0. Por otra parte, el programa que se ha utilizado como simulador no tiene esta función, así que el circuito con nexos lógicos no lo podemos representar. No obstante, podemos crear un circuito electrónico que lo simule, aunque como es lógico, no tendrá las mismas características visuales que los anteriores. Es circuito es el siguiente:



3.9 Valoraciones sobre este apartado

Este ha sido quizás uno de los apartados más diversos y más completos de todo el trabajo de investigación. En este apartado se ha encontrado desde la parte más teórica de la lógica matemática, es decir, el entender todos los nexos lógicos, simples y derivados, hasta entender como y qué hace que una persiana suba y baje, entre otras cosas.

En primer lugar, se ha empezado analizando la teoría que posteriormente nos ayudaran a entender las prácticas. Ha sido en este apartado donde se ha podido comprobar nuestras "sospechas" sobre las aplicaciones prácticas de la lógica matemática. Si en la anterior valoración sobre la introducción a la lógica dábamos un porcentaje espectacular sobre el incremento de vendas de los teléfonos móviles, como ejemplo de la (r)evolución en las nuevas tecnologías, ahora podemos ver la relación directa entre éstas y la parte más práctica de la lógica matemática. Es decir, que ahora podemos ver que mediante un previo desarrollo y una posterior compilación de un programa, podemos realizar todo tipo de funciones: algunas que nos pueden parecer más sencillas, hasta otras como realizar que un "scalextric" se pueda convertir en una calculadora.

Personalmente, ha sido la parte del trabajo en la que más he aprendido, dado que mi trabajo se basaba en analizar qué era esta parte de las matemáticas, y posteriormente analizar su parte más práctica. Así pues, y aunque aún nos quedan por analizar muchos otros términos y aspectos muy interesantes, este ha sido el apartado más general sobre la lógica.

Parte teórica

Como ya se ha señalado anteriormente, la parte principal de este trabajo es la parte práctica, dado que en primer lugar, creo que es la más importante, y en segundo lugar, creo que es la más interesante. No obstante, no se ha de olvidar que hay aspectos de la lógica matemática, como son el álgebra de Boole u otras leyes fundamentales que son indispensables analizar. Así pues, este segundo gran apartado del trabajo consistirá en analizar detalladamente los aspectos más teóricos de este campo de la matemática, sin olvidar la idea de un trabajo práctico. Para ello, se ha diseñado un algoritmo que nos resuelve un buscaminas.

4. Álgebra de Boole

4. Álgebra de Boole

En este primer apartado explicaremos las diferentes leyes que existen dentro del álgebra de Boole. Como en matemáticas, existen la propiedad conmutativa o la distributiva. Además, todas ellas son deducibles a partir de los conocimientos previamente adquiridos en el trabajo.

4.1 Introducción

El **Álgebra de Boole** es una rama de las matemáticas con ciertas propiedades, aunque todas ellas distintas al álgebra más técnica. Entre otras cosas, cabe decir que es útil en el campo de la lógica. Esta álgebra es un sistema matemático compuesto por un conjunto de elementos, llamado habitualmente B, junto a dos operaciones binarias fundamentales, que se pueden escribir con los símbolos de sumar y multiplicar. Estas operaciones están definidas en el conjunto B y satisfacen los cuatro postulados de Huntington, que son los siguientes:

- Ley conmutativa
- Ley distributiva
- Elementos cero y unidad
- Complementación

4.2 Propiedad conmutativa

Esta primera propiedad que analizaremos, es igual que en matemáticas. La **propiedad conmutativa** nos ayuda a entender que es lo mismo sumar A + B, que sumar B + A. En lo referente a nuestro trabajo, es lo mismo decir que el resultado tendrá valor cierto si A y B se cumplen, que si B y A se cumplen.

$$A \cdot B = B \cdot A$$

$$A + B = B + A$$

$$(1-A) \cdot (1-B) = (1-B) \cdot (1-A)$$

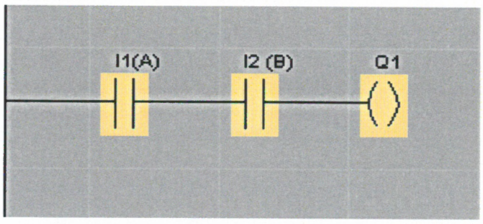
En esta propiedad, podríamos incluir todas las operaciones lógicas que sean simétricas. Sin embargo, lo que no podemos hacer es cambiar el orden de los nexos lógicos. Por ejemplo, no es lo mismo decir que si llueve y no nieva cogeré el coche, que si nieva y no llueve cogeré el coche. Si hacemos esta relación, estaríamos haciendo una relación equivocada.

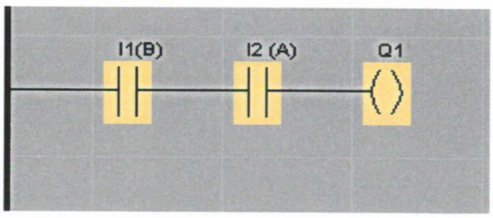
Para comprobar la primera propiedad, la propiedad conmutativa, haremos los pasos que se ha aprendido hasta ahora: la tabla de verdad, circuito con nexos lógicos y circuito eléctrico:

A	B	Q=A·B
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

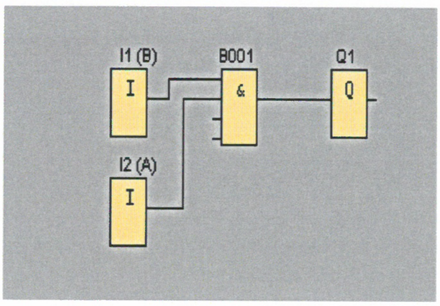
B	A	Q=B·A
0	0	**0**
0	1	**0**
1	0	**0**
1	1	**1**

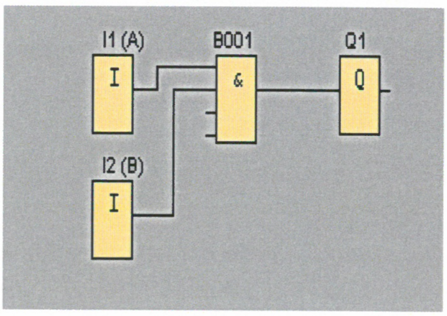
Como podemos observar, tanto en el primer caso como en el segundo, la operación matemática solo tendrá valor cierto, si A y B son ciertas. Por lo tanto, es lógico que sea lo mismo decir A·B que B·A. Por otra parte, en lo referente a los circuitos lógicos de las dos operaciones, podemos observar que son exactamente idénticos:





Vemos que no cambia en nada que A y B cambien el orden en el circuito. Podemos observar la misma característica en el esquema eléctrico con nexos lógicos.





4.3 Propiedad distributiva

Esta segundo propiedad, también tiene mucha relación a las matemáticas ordinarias, a pesar de que recordemos, el "+" y el "*" en este campo de la matemáticas, representan un símbolo, no representan una función. No obstante, ya podemos ver una diferencia entre el álgebra común, y la de Boole, dado que como ya sabemos, en el álgebra ordinaria no se cumple la propiedad distributiva de la suma respecto al producto, mientras en la de Boole, como ya veremos, sí. Este segundo postulado, dice que cada una de las operaciones "+" y "*" es distributiva con respecto a la otra. Esto es, para tres elementos cualesquiera x, y, z del conjunto B, se cumple que:

$$x \cdot (y+z) = (x \cdot y) + (x \cdot z)$$

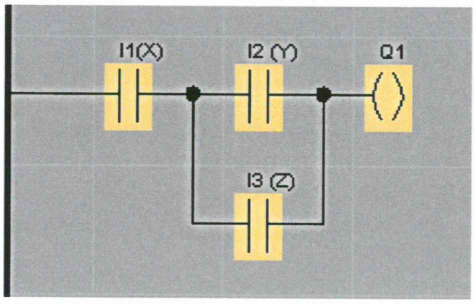
$$x + (y \cdot z) = (x+y) \cdot (x+z)$$

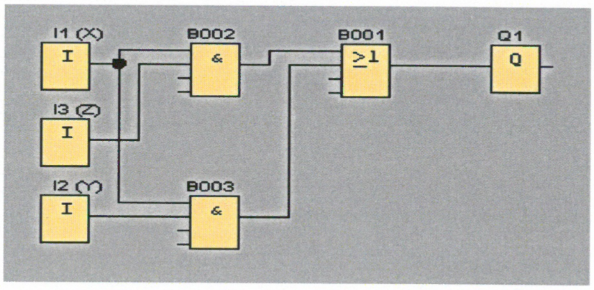
Para analizar más detalladamente esta propiedad, y verificarla, utilizaremos una tabla de verdad que nos ayudará a entender el porque este postulado de Huntington es cierto.

X	Y	Z	$(x \cdot y) + (x \cdot z)$
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**0**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

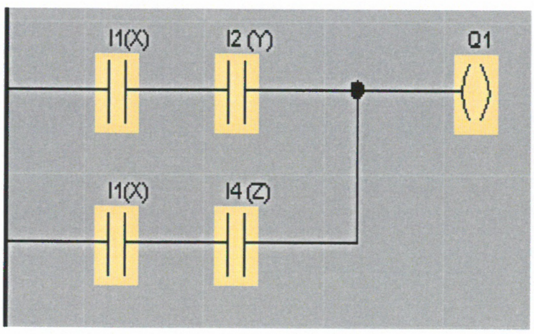
X	Y	Z	$x\cdot(y+z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

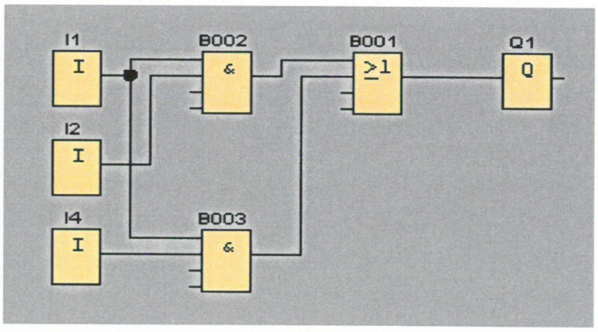
Como podemos observar si analizamos los resultados obtenidos en las dos tablas de certeza, vemos que efectivamente los resultados coinciden. No obstante, aun nos quedan dos pruebas para ver si estos dos postulados son ciertos, o, por si lo contrario fallan en algún punto. Estas dos pruebas son los circuitos eléctricos y lógicos, que, a continuación están expuestos:





En este primer circuito eléctrico, podemos observar un contacto abierto, conectado en serie a dos contactos situados en paralelo, que están unidos a una bobina. Así pues, sabemos que es la función $x\cdot(y+z)$.





En segundo lugar, vemos a la segunda función que estamos intentando demostrar, y si observamos bien, vemos que los dos circuitos de nexos lógicos son idénticos, a pesar de que sus circuitos eléctricos, sean, aparentemente, distintos. Así pues, queda demostrado este segundo postulado de Huntington.

4.4 Elementos cero y unidad

En el conjunto B existe un elemento neutro bien definido para cada una de las operaciones sumar y multiplicar. Estos elementos se representan habitualmente con los símbolos 0 y 1, y habitualmente, se les llama al 0 elemento ínfimo y al 1, elemento universal. Estos dos elementos, tienen la propiedad que:

$$0 + x = x$$

$$1 \cdot x = x$$

Así pues, estas dos operaciones lógicas, tienen una propiedad fundamental, que es que tienen un elemento neutro cada uno. De la misma manera que en una suma, en álgebra ordinaria, el elemento 0 hace de número neutro, dado que no modifica el resultado de la suma, y en la multiplicación el 1, en el sistema lógico también. Como en álgebra ordinaria, el 0 representa el elemento nulo en la suma, y el 1 en la multiplicación.

Para entender correctamente este tercer postulado, pondremos un ejemplo más. En el caso de la multiplicación, si tuviéramos la siguiente operación,

$$0 \cdot x = 0$$

Su resultado sería 0, debido a la propiedad nula del 0.

4.5 Complementación

Para todo elemento a de C existe otro elemento a’, llamado complementario de a, que cumple las siguientes propiedades con respecto a las operaciones de sumar y multiplicar:

$$A + A' = 1 \qquad\qquad A \cdot A' = 0$$

Esta caracterización de las álgebras de Boole a partir de estos cuatro postulados fue formulada por E.V. Huntington en el año 1904. No obstante, se ha de hacer reseña en decir que hay muchas otras formas de caracterizar a las álgebras de Boole a partir de otros grupos de axiomas. Podemos observar también, una gran diferencia entre el álgebra de Boole, y el álgebra ordinaria, dado que en esta última no se cumple la propiedad distributiva de la suma respecto al producto, mientras que en el álgebra de Boole, sí.

La gran ventaja de los postulados de Huntington es la simetría entre las dos operaciones sumar y multiplicar. Esta simetría permite formular el siguiente principio de dualidad, que junto con las leyes de Morgan, serán las dos leyes que analizaremos pero que no estan incluidas dentro de los cuatro postulados de Huntington respecta al álgebra de George Boole.

4.6 Principio de dualidad

En toda álgebra de Boole, si en un teorema válido intercambiamos las dos operaciones, sumar y multiplicar, e ínfimo por universal, es decir 0 por 1, obtenemos otro teorema igualmente válido.

4.7 Leyes de Morgan

Aunque esta ley no está incluida en las principales propiedades del álgebra de Boole, cabe decir que estas leyes son fundamentales en el campo de la lógica matemática. No obstante, esta ley, y otras que no analizaremos debido a que no tienen tanta importancia, se pueden deducir a partir de los cuatro postulados, como veremos más adelante. Así pues, los cuatro **postulados de Huntington** son más que suficientes y necesarios para caracterizar un álgebra de Boole.

A continuación analizaremos las importantísimas leyes de Morgan. En el lenguaje corriente ocurre a veces que hay proposiciones enunciadas de manera distinta que tienen el mismo significado. Por ejemplo,

1. No llueve y no hace sol

se puede también expresar, aunque de manera un poco forzada, que:

2. No ocurre que llueva o que haga sol.

Si (1) y (2) significan lo mismo aunque expresado de forma distinta, entonces en lógica debería de ser válido concluir que (1) de (2) o (2) de (1), lo que expresado en forma matemática vendría a representar lo siguiente:

a) de $(1-P)\cdot(1-Q)$ se puede concluir que $1-(P+Q)$

b) de $1-(P+Q)$ se puede concluir que $(1-P)\cdot(1-Q)$

De esta manera, (b) si no tiene o P o Q, entonces no se tiene P y no se tiene Q. La regla que permite esta conclusión es una de las denominadas leyes de Morgan.

Las premisas de (a) y (b) son dos de las formas proposicionales moleculares a las que se les puede aplicar las leyes de Morgan. Estas leyes también se aplican a otras formas proposicionales, como se puede ver al considerar las dos proposiciones equivalentes:

1) O no hace caloro no nieva
2) No ocurre que a la vez haga calor y nieve.

Puesto que (3) y (4) tienen el mismo significado, una puede deducirse de la otra. Por lo tanto, en forma algebraica se puede escribir:

c) de $(1-P)+(1-Q)$ se puede concluir $1-(P\cdot Q)$

d) de $1-(P\cdot Q)$ se puede concluir $(1-P)+(1-Q)$

(c) y (d), son pues, otros dos ejemplos de la aplicación de las leyes de Morgan.

Para comprobar que esta ley, deberíamos de hacer una tabla de verdad:

$$\overline{(A\cdot B)} = \overline{A} + \overline{B}$$

$1-(A\cdot B) \rightarrow (1-A)+(1-B)$						
A	B	$A\cdot B$	$1-(A\cdot B)$	$(1-A)$	$(1-B)$	$(1-A)+(1-B)$
0	0	0	**1**	1	1	**1**
0	1	0	**1**	1	0	**1**
1	0	0	**1**	0	1	**1**
1	1	1	**0**	0	0	**0**

De esta manera demostramos que $1-(A\cdot B))\leftrightarrow$ $(1-A)+(1-B)$ es cierto. No obstante, aun nos queda por comprobar la segunda premisa de estas leyes de Morgan, que es la siguiente

$$\overline{(A+B)} = \overline{A} \cdot \overline{B}$$

$1-(A+B) \rightarrow (1-A)\cdot(1-B)$						
A	B	$A+B$	$1-(A+B)$	$(1-A)$	$(1-B)$	$(1-A)\cdot(1-B)$
0	0	0	**1**	1	1	**1**
0	1	1	**0**	1	0	**0**
1	0	1	**0**	0	1	**0**
1	1	1	**0**	0	0	**0**

Como podemos observar en la tabla de verdad, obtenemos los mismos resultados en ambos casos. Así pues, se ha demostrado lo que Augustus de Morgan demostró en el siglo XIX. Con frecuencia hay más de una forma posible para la conclusión. Afortunadamente, estudiando las diferentes formas proposicionales a las que se aplican las leyes de Morgan se obtiene un modelo que se puede seguir siempre. Será posible entonces enunciar las leyes de Morgan como una regla que se aplicará a cada una e las formas de premisas en las que puede utilizarse, y en todo caso se obtiene la forma de la conclusión deseada. En definitiva, lo que se ha de hacer para aplicar las leyes de Morgan, como una regla de operación, es verificar los siguientes pasos:

- Cambiar "Y" por "O" u "O" por "Y"
- Negar cada término de la disyunción o conjunción
- Negar la fórmula completa

4.7.1 Demostración de la primera ley de Morgan

$A'+B'=(A\cdot B)'$

$$\text{X}\ \varepsilon\ A'+B' \rightarrow \begin{cases} \text{X} \notin \text{A} \\ \text{ó} \\ \text{X} \notin \text{B} \end{cases}$$

$\text{X} \notin A\cdot B \rightarrow \text{X}\ \varepsilon\ (A\cdot B)'$

De esta manera, podemos observar que es lo mismo afirmar que no A o no B, es lo mismo que no A y B. Seguidamente pondremos un ejemplo donde podremos ver que, efectivamente, esta primera ley de Morgan es cierta, pero antes, veremos, mediante una tabla de certeza que esta primera ley se cumple:

A	B	$A'+B'$	A	B	$(A\cdot B)'$
0	0	**1**	0	0	**1**
0	1	**1**	0	1	**1**
1	0	**1**	1	0	**1**
1	1	**0**	1	1	**0**

Podemos observar que los resultados de ambas operaciones matemáticas, son iguales, con la cual cosa, podemos afirmar que son ciertas. No obstante, y para asegurarnos que no es debido a un hecho casual, que el resultado coincida en ambas operaciones lógicas, se ha decidido hacer un diagrama[10], donde podemos observar claramente la veracidad de esta primera ley de Morgan.

De esta manera, queda claro que lo que no es A (A', o lo que está de color amarillo), o lo que no es B (B', o todo lo que está rayado), es la misma parte que lo que no pertenece a A y B, es decir, la parte pintada de color amarillo y rayada.

[10] Ver anexo 9 para ver este diagrama

4.7.2 Demostración de la segunda ley de Morgan

$A' \cdot B' = (A + B)'$

$$X \,\varepsilon\, A' \text{ y } B' \left\{ \begin{array}{l} X \,\varepsilon\, A' \rightarrow X \notin A \\ \text{y} \qquad\qquad\quad \text{y} \\ X \,\varepsilon\, B' \rightarrow X \notin B \end{array} \right\} X \notin A + B \rightarrow X \,\varepsilon\, (A + B)'$$

Como se ha hecho en la primera ley de Morgan, veremos, mediante una tabla de verdad si obtenemos los mismos resultados con estas dos operaciones lógicas.

A	B	$A' \cdot B'$	A	B	$(A+B)'$
0	0	**1**	0	0	**1**
0	1	**0**	0	1	**0**
1	0	**0**	1	0	**0**
1	1	**0**	1	1	**0**

Así pues, los resultados coinciden, hecho que nos deja entrever, que la segunda ley de Morgan es cierta. Sin embargo, si observamos el diagrama[11], veremos la completa veracidad de esta segunda ley de Morgan.

A continuación exponemos las principales propiedades y leyes del álgebra de Boole. Como se puede observar, están separadas las dos operaciones básicas, la suma y la resta. Esta t**abla**[12], es una gran **ayuda** para poder entender todos los postulados y las demostraciones.

---

[11] Ver anexo 10 para ver este diagrama
[12] http://es.wikipedia.org/wiki/%C3%81lgebra_de_Boole

	Adición	Producto
1	$A + A' = 1$	$A \bullet A' = 0$
2	$A + 0 = A$	$A \bullet 1 = A$
3	$A + 1 = 1$	$A \bullet 0 = 0$
4	$A + A = A$	$A \bullet A = A$
5	$A + B = B + A$	$A \bullet B = B \bullet A$
6	$A + (B + C) = (A + B) + C$	$A \bullet (B \bullet C) = (A \bullet B) \bullet C$
7	$A + B \bullet C = (A + B) \bullet (A + C)$	$A \bullet (B + C) = A \bullet B + A \bullet C$
8	$A + A \bullet B = A$	$A \bullet (A + B) = A$
9	$(A + B)' = A' \bullet B'$	$(A \bullet B)' = A' + B'$

4.8 Teoría de conjuntos

En esta parte del trabajo utilizaremos dos tipos de notaciones: la notación que se ha usado durante el conjunto del trabajo, y la que se usa para analizar estas teorías, la notación lógico matemática. El objetivo de este apartado, es ver la relación entre esta importante teoría y la lógica matemática, así como entender sus conceptos fundamentales.

Esta teoría fue expuesta por **Georg Cantor** en el siglo XIX. Antes de analizar esta teoría, deberíamos definir el concepto de ***conjunto***. Según el diccionario de la real academia española, en la sexta acepción, un conjunto es "la totalidad de los elementos o cosas poseedores de una propiedad común, que los distingue de otros; p. ej., los números pares". De esta manera, podemos añadir que un conjunto está bien definido cuando se sabe que un determinado elemento forma parte o no de un conjunto. Así pues, un lápiz rojo está dentro del conjunto de lápices rojos, porque podemos definir cuando un lápiz es rojo o no. En este caso, cualquier persona formaría los mismos conjuntos, aunque fuera daltónica, dado que quizás llamaría a este conjunto de otra manera, pero agruparía a todos los lápices rojos en el mismo conjunto. No obstante, hay cosas que no pueden ser definidas dentro de un conjunto, como las personas altas o bajas. No las podemos agrupar en un mismo conjunto, dado que el ser alto o el ser bajo es muy subjetivo. De esta manera, una persona podría considerar baja a una persona de 1.65 metros, mientras, que quizás, otras pondrían el límite en 1.6 metros. De esta manera, la única manera de poder agruparlos en un conjunto, es si pusiéramos un límite fijo.

Un conjunto se representa siempre en mayúsculas, y se pude utilizar cualquier letra para definirlos. Sin embargo, se suele utilizar la letra U, como el **conjunto universal** de las cosas que estamos tratando. De esta manera, si estamos intentando analizar el conjunto de países que forman la Unión europea, el conjunto U serán todos los países de la Unión. Otros ejemplos de conjuntos son:

- 0 : el *conjunto vacío*, que carece de elementos.
- **N**: el conjunto de los *números naturales*.
- **Z**: el conjunto de los *números enteros*.
- **Q** : el conjunto de los *números racionales*.
- **R**: el conjunto de los *números reales*.
- **C**: el conjunto de los *números complejos*

La relación con la lógica matemática es total, dado que establece ciertas reglas de la lógica. De esta manera, y como ya se había dicho anteriormente, si dentro de U hay un subconjunto A, el conjunto U es A y no A, es decir, la operación lógico matemática $U = A' + A$. Además, se puede argumentar que si $x \in A$, entonces $x \notin A'$

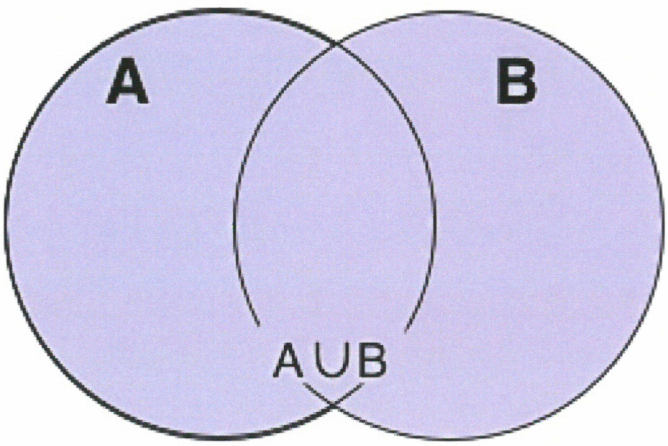
Por otra parte, existen los **elementos**, es decir, cada uno de los elementos que forman parte de un conjunto. Cada elemento es único, es decir, que no hay dos elementos iguales dentro de un mismo conjunto. Se representan con minúsculas. De esta manera, si A es un conjunto, y b y c son los elementos, se representaría $A = \{a, b \ \}$.

Esta notación se llama ***notación por extensión.*** Además, podemos decir del conjunto A que se ha creado, que $a,b \in A$. Si por lo contrario a y b no perteneciesen a A, se tendría que decir que $a,b \notin A$. Finalmente, si los elementos dentro del conjunto cumplen alguna propiedad, se ha de definir que $A = \{x : p(x)\}$. Esta notación, llamada ***notación por compresión*** viene a decir que, A es el conjunto de elementos x, que cumplen la función p(x). De esta manera, el conjunto de los números naturales se define $A = \{n : 1 \leq n \leq 3, n \in \mathrm{N}\}$.

Otra de las teorías expuestas por Cantor fue la **igualdad de conjuntos** y los **subconjuntos**. Dos conjuntos son iguales si para todos los valores de x, se cumple que $x \in A$ y que $x \in B$. Por otra parte, y en el caso en que para un cierto valor de x se cumpla que $x \in A$ y que $x \in B$, entonces, A es un subconjunto de B, o viceversa. En el caso en que A fuera un subconjunto de B, entonces B sería un **superconjunto** de A. Lo que se escribiría de manera matemática $A \subseteq B$, y $B \supseteq A$. Además, los conjuntos A y B son subconjuntos y superconjuntos de si mismos.

Como ya se ha visto anteriormente en otros apartados del trabajo, se pueden realizar operaciones con nexos lógicos. En este caso, analizaremos tres operaciones más, que son la de unión, intersección y diferencia. De esta manera, podemos operar con conjuntos, de la misma manera que con operadores lógicos.

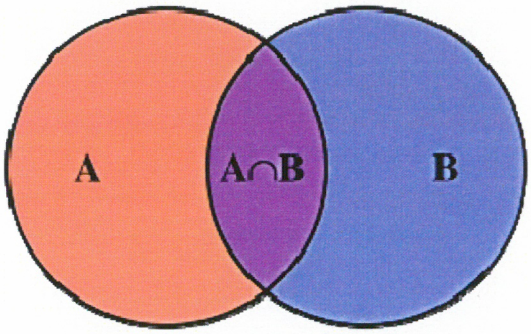
En primer lugar, la operación de **unión** corresponde a la siguiente imagen[13], donde tenemos dos conjuntos, A y B que están unidos. De esta manera, tenemos que:



[13] http://es.wikipedia.org/wiki/Teor%C3%ADa_de_conjuntos

, es decir, que $A \cup B = \{x : x \in A \vee x \in B\}$, o escrito en el lenguaje que se ha estado utilizando durante todo el trabajo $A + B = \{x : x \in A + x \in B\}$. Así pues, si $A = \{1,2,3\}$ y $B = \{2,3,4\}$, entonces $A \cup B = \{1,2,3,4\}$.

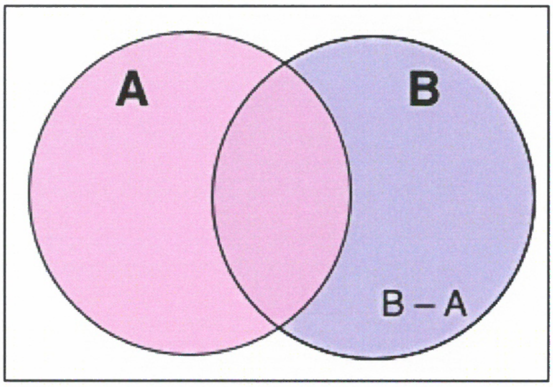
En segundo lugar, se analizará la **intersección** entre dos conjuntos. En este caso, la operación lógica correspondiente es la multiplicación, dado que se debe leer como el conjunto formado por la intersección entre A y B. De esta manera, la operación será $A \cap B = \{x : x \in A \wedge x \in B\}$, o $A \cdot B = \{x : x \in A \cdot x \in B\}$. La imagen[14] gráfica es la siguiente:



Así pues, en este caso se puede poner el siguiente ejemplo: si $A = \{1,2,3\}$ y $B = \{2,3,4\}$, entonces $A \cap B = \{2,3\}$.

Finalmente, se analizará la **diferencia** entre diferentes conjuntos. Si tenemos dos conjuntos que interseccionan entre si, podemos afirmar que los elementos del conjunto B que no son elementos de la intersección, son la diferencia, como se puede observar en la siguiente imagen[15].

[14] http://es.wikipedia.org/wiki/Teor%C3%ADa_de_conjuntos
[15] http://es.wikipedia.org/wiki/Teor%C3%ADa_de_conjuntos



De esta manera, podemos decir que $B - A = \{x : x \in B \wedge x \notin A\}$, o $B - A = \{x : x \in B \cdot x \notin A\}$. Además, podemos ir más allá y deducir una igualdad.

Si x es la diferencia entre los conjuntos A y B, entonces, x contiene B y no A.

$x \in (B - A) \Leftrightarrow x \in B \wedge x \notin A \Leftrightarrow x \in B \wedge x \in A'$
$x \in (B - A) \Leftrightarrow x \in (B \cap A')$

De esta manera, tenemos que la diferencia corresponde a la expresión de que x corresponde al subconjunto formado por B y no A.

$B - A = B \cap A'$
$x \in (B \cap A') \Leftrightarrow x \in B \wedge x \in A' \Leftrightarrow x \in A' \wedge x \in B \Leftrightarrow x \in A' \wedge x \notin B' \Leftrightarrow x \in (A' - B')$

Así pues, se ha deducido que $B - A = A' - B'$.

Si anteriormente habíamos expuesto una tabla con el conjunto de operaciones con operadores lógicos, tenemos que, siendo A, B y C tres conjuntos cualquiera, y U el superconjunto de éstos[16]:

[16] http://wmatem.eis.uva.es/~matpag/CONTENIDOS/Conjuntos/marco_conjuntos.htm

PROPIEDADES	UNION	INTERSECCION
1.- Idempotencia	$A \cup A = A$	$A \cap A = A$
2.- Conmutativa	$A \cup B = B \cup A$	$A \cap B = B \cap A$
3.- Asociativa	$A \cup (B \cup C) = (A \cup B) \cup C$	$A \cap (B \cap C) = (A \cap B) \cap C$
4.- Absorción	$A \cup (A \cap B) = A$	$A \cap (A \cup B) = A$
5.- Distributiva	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
6.- Complementariedad	$A \cup A' = U$	$A \cap A' = \varnothing$

Otra de las operaciones que se pueden realizar con diferentes conjuntos, es la multiplicación, o el ***producto cartesiano***. De esta manera, el producto de dos conjuntos se define por $A \cdot B = \{(x, y) | x \in A \wedge y \in B\}$. De esta manera, y si $A = \{1,2,3\}$ y $B = \{4,5,6\}$, el *producto cartesiano* es $S = A \cdot B$, es decir, $S = \{4,10,18\}$.

Una vez se ha definido el *producto cartesiano*, hay dos conceptos en relación a éste que hay que definir: los ***grafos*** y las ***proyecciones***. En primer lugar, llamamos *grafo relativo* a $A \cdot B$ a todo subconjunto del producto. Así pues, si C es el superconjunto del producto entre dos conjuntos cualesquiera A y B, se define $C \subseteq A \times B$. Utilizamos el signo de multiplicación de una cruz, para no confundir con el nexo lógico "y", que en este caso no tiene nada que ver. En segundo lugar, y dado un *grafo relativo* del producto entre dos conjuntos, se llama *proyección* de G sobre A al conjunto W_aG= $\{a \in A | (a,b) \in G, \exists b \in B\}$. El símbolo $\exists$ es un cuantificador, llamado el cuantificador existencial, que hablaremos más adelante en este apartado, aunque viene a decir (la proposición entera), que existen una a y una b, tal que su par pertenece al conjunto G.

Todos estos conceptos pueden generalizarse en ***familias de conjuntos***. Si para todos los elementos i dentro de un conjunto A, entonces se define el conjunto $\{Ai : i \in I\}$, que viene a denominar la *familia de conjuntos* indicada por I.

Dos pares (a,b) y (c,d) de $A \cdot B$ son iguales, si y solo si a = c y b=d. Así pues, y de la misma manera tenemos que si, $A \cdot B = C \cdot D \Leftrightarrow (A = C \wedge B = D)$.

Otra de las características de esta teoría de conjuntos, es que hay unos cuantificadores que nos permiten definir nuevas reglas. Así pues, tenemos dos cuantificadores: el cuantificador universal y el existencial. Ambos tienen la función de indicar si ciertos elementos cumplen con alguna propiedad.

El **cuantificador *universal*** se representa por $\forall$, y se usa para decir que *todos* los elementos de un conjunto cumplen con una cierta propiedad. Anteriormente habíamos definido en la *notación por compresión* esta misma propiedad. De esta manera, son equivalentes estas dos notaciones. Si anteriormente decíamos que $A = \{x : p(x)\}$, para definir que todos los elementos x dentro del conjunto A, cumplían con la función p(x), propiedad que hace referencia a un elemento x, ahora escribiremos $\forall x \in A | \; p(x)$.

Por otra parte, el **cuantificador *existencial*** se usa para decir que al menos un elemento dentro del conjunto que se quiere analizar, cumple con una cierta propiedad p(x). Así pues, decimos que $\exists x \in A | p(x)$. De esta manera, esta proposición se lee diciendo que existe una x que pertenece al conjunto A, tal que cumpla la propiedad p(x). Muchas veces se representa esta función mediante otra expresión equivalente $\{x \in A | p(x)\} \neq 0$.

En el caso que se quisiera negar cualquiera de las dos proposiciones anteriormente analizadas, se debería negar la proposición p(x) y modificando el cuantificador. De esta manera, si se tuviera la proposición $\forall x \in A | \; p(x)$, su negación sería $\exists x \in A | p(x)'$.

Una vez analizados todos los aspectos de esta teoría matemática, vemos la relación que hay entre la lógica matemática y la teoría de conjuntos. Para acabar de esta relación, se ha diseñado una tabla comparativa entre la lógica de proposiciones y la teoría de conjuntos.

Teoría de con juntos	Lógica matemática
$A \cup B$	$A + B$
$A \cap B$	$A \cdot B$
A'	$1 - A$
$A - B$	$A \cdot (1 - B)$
$A \cup (A \cap B) = A$	$A + (A \cdot B) = A$
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$A + (B \cdot C) = (A + B) \cdot (A + C)$
$(A \cup B)' = A' \cap B'$	$(1 - (A + B)) = (1 - A) \cdot (1 - B)$

4.9 Posibilidades de resultado con nexos lógicos

Una vez se ha analizado la teoría de la lógica matemática, se intentó ver si se pueden formular todo tipo de resultados mediante las operaciones lógicas. De esta manera, queríamos ver si era posible obtener todos los resultados, combinando los nexos lógicos existentes. Cabe decir, que este no es un trabajo de buscar información en Internet o en libros, sino que este fue un trabajo propio.

A	B	$A \cdot (1 - A)$	$A \cdot B$	$A \cdot (1 - B)$	A	$B \cdot (1 - A)$	B
0	0	**0**	**0**	**0**	**0**	**0**	**0**
0	1	**0**	**0**	**0**	**0**	**1**	**1**
1	0	**0**	**0**	**1**	**1**	**0**	**0**
1	1	**0**	**1**	**0**	**1**	**0**	**1**

A	B	$((1 - A) \cdot B) + (A \cdot (1 - A))$	$A + B$	$1 - (A \cdot B)$	$1 - (((1 - A) \cdot B) + (A \cdot (1 - B))$	$1 - B$
0	0	**0**	**0**	**1**	**1**	**1**
0	1	**1**	**1**	**0**	**0**	**0**
1	0	**1**	**1**	**0**	**0**	**1**
1	1	**0**	**1**	**0**	**1**	**0**

A	B	$A+(1-B)$	$1-A$	$(1-A)+B$	$(1-A)+(1-B)$	$(1-A)+A$
0	0	**1**	**1**	**1**	**1**	**1**
0	1	**0**	**1**	**1**	**1**	**1**
1	0	**1**	**0**	**0**	**1**	**1**
1	1	**1**	**0**	**1**	**0**	**1**

El primer aspecto que deberíamos analizar es la **existencia del 0**. Si analizamos el primer caso que se ha puesto, es decir en el caso en que quisiéramos obtener como resultado todo ceros, es decir, todos los resultados falsos, deberíamos negar la variable que teníamos anteriormente. De esta manera, y si lo pensamos más detenidamente, este hecho es lógico, dado que si no puede ser que tengamos A y no A a la vez.

En segundo lugar, cabe relacionar uno de los ejemplos que se ha citado anteriormente, con un nexo lógico derivado. En la segunda tabla de operaciones, la primera de ellas corresponde a la operación **XOR**.

Finalmente, y la tercera conclusión que se ha sacado del análisis de esta tabla, es la **existencia del uno**. Como conocemos en álgebra, todo número o variable que la multiplicamos por uno, obtenemos el mismo resultado. Pues bien, en este caso también tenemos esta operación, que corresponde a la última de todas. Si nos miramos esta tabla de una manera racional, veremos que es lógico lo que se ha escrito, dado que si tenemos que o no A o A, obtenemos siempre un resultado positivo, dado que para todos los valores de A obtenemos valor cierto, y para todo lo que no es A también. De esta manera, es imposible que obtuviéramos otro resultado que no fuera todo positivo.

Como en un principio pensábamos, se pueden obtener todos los resultados mediante las operaciones lógicas. De esta manera, podemos observar también, que obteniendo la mitad de los resultados, podríamos haber obtenido los demás, dado que simplemente, tendríamos que haber negado estos. Aunque hayamos puesto sólo una operación por ejemplo, podríamos haber puesto otras muchas, aunque se ha preferido estas, dado que son las más simplificadas que se ha podido obtener.

4.10 Valoración personal

Este es con toda seguridad la parte del trabajo que ha sido de mayor labor matemática, dado que todas estas demostraciones, las se ha tenido que demostrar a partir de otras que eran de una gran complejidad. No obstante, ha sido una labor entretenida, dado que el hecho de aprender nuevas matemáticas, nuevos símbolos matemáticos, e intentar entender las matemáticas de uno de los mejores y más reconocidos matemáticos de todos los tiempos, ha sido un gran placer personal.

Cabe resaltar dentro de este último apartado, que además de toda la labor de racionalizar unas expresiones que en un primer principio nos parecían irracionales, se ha realizado una labor de análisis nuevo. Decimos que se ha realizado una labor de análisis nuevo, dado que en ningún libro ni en ninguna fuente de Internet, se ha podido encontrar la última parte de este trabajo. Aunque pueda parecer sencilla, resulta compleja y entretenida, dado que se ha podido ver que con la excepción de algunos casos, en concreto dos, se ha encontrado una fórmula como mínimo para cada posibilidad. Dentro de las fórmulas que se ha encontrado, podríamos haber puesto más, pero se ha creído que con una fórmula era suficiente para entender lo que queríamos expresar. No obstante, resultó un poco frustrante, el estar pensando conseguir una posibilidad que es creíamos imposible. Sin embargo, todos los resultados que se ha obtenido son lógicos, y si en vez de intentar resolver este problema de manera matemática hubiéramos intentado resolverlo de una manera más racional, los hubiéramos resuelto mucho más rápido.

Así pues, y a pesar de la complejidad de estas matemáticas, ha resultado interesante el buscar y analizar esta álgebra de Boole, y posteriormente intentar encontrar fórmulas para cada posibilidad. En el siguiente punto seguiremos analizando teoría sobre la lógica matemática, y veremos más aplicaciones prácticas sobre este campo de las matemáticas.

5. Decidibilidad

Si se ha investigado tan profundamente en un concepto tan teórico, que quizás, se aleja un poco de la idea principal del trabajo es por su suma importancia. No obstante, realizaremos también, una parte práctica para acabar de entender este importante concepto dentro del campo de la lógica matemática.

5.1. Concepto de decidibilidad

Las cuestiones de **decibilidad** son importantes en la parte algorítmica de las matemáticas y especialmente, en la teoría de los autómatas. Decimos que un problema es decidible si existe un programa o algoritmo que lo resuelva. Si este programa no existe, decimos entonces que el problema no puede ser decidible.

El concepto de indecibilidad ha jugado un papel muy importante en el último siglo. De hecho, hace cien años que David Hilbert propuso que había un programa efectivo que determinaba la veracidad o la falsedad de cualquier operación matemática. No obstante, esta propuesta de Hilbert fue destrozada por el concepto de ***incompletness theorem*** del científico Gödel.

Este teorema tiene dos puntos fundamentales. En primer lugar, y "*en cualquier formalización consistente de las matemáticas que es lo bastante fuerte para definir el concepto de números naturales, se puede construir una afirmación que ni se puede demostrar ni se puede refutar dentro de ese sistema".* Este célebre teorema, es uno de los peores comprendidos en el mundo científico, y a dado paso a múltiples afirmaciones e interpretaciones. Si analizamos bien este primer teorema, veremos que viene a decir que cualquier sistema que permita definir los números naturales es necesariamente incompleto: contiene afirmaciones que ni se pueden demostrar ni contradecir. De esta manera, y la idea principal a la que Gödel quería llegar, era que en la mayoría de los casos, no podemos descubrir todo el conjunto de axiomas, dado que cada vez que añadamos un nuevo axioma, habrá otro que quedará fuera de alcance. En segundo lugar, y el segundo teorema de Gödel era que "n*ingún sistema consistente puede usarse para demostrarse a sí mismo*". Esta idea, más filosófica que matemática por otra parte, se parece mucho a las ideas defendidas por Karl Popper, cuando éste afirmaba que los científicos tenían una fe irracional en la razón. Es decir, que los científicos confiaban en sus teorías científicas dado que sus cálculos les habían dicho que funcionaban. De esta manera, Kurt Gödel rechazó la idea de un programa que determinaba la veracidad o falsedad de una operación matemática, mediante estos dos teoremas.

Una vez analizado el teorema de incompletud de Gödel, intentaremos analizar el concepto de indecibilidad. Para clasificar este concepto se necesitaba una definición clara. Ésta llegó con la maquina de Turing que era considerada como un algoritmo. Una vez esta notación fue formalizada, se demostró que existían muchos programas para los cuales no había solución posible. Para demostrar que un problema no es decidible, sólo tenemos que demostrar que no hay un algoritmo que lo solucione. Esto se suele hacer reduciendo al máximo el programa a un algoritmo P que se sepa que funciona. En esta reducción, se ha de demostrar que si nuestro programa fuera decidible, entonces P sería una contradicción. Algunos de estos simples programas de indecibilidad son el problema *Post Correspondance* o el décimo problema de Hilbert.

El ocho de agosto de mil novecientos los mejores pensadores de la época se reunieron en Paris en el II Congreso Internacional de Matemáticas. En esa convención matemática, David Hilbert propuso los veintitrés problemas abiertos en matemáticas, e instó a los matemáticos a resolverlos. Estos veintitrés problemas fueron llamados los veintitrés problemas de Hilbert. Estos problemas abarcaban todos los campos de las matemáticas: ocho eran de puramente de investigación y los quince restantes eran problemas concretos. De estos quince problemas concretos, doce han sido resueltos y aún hoy hay tres que no se han podido resolver. Estos tres son:

- La hipótesis de Riemann
- La conjetura de Goldbach
- La conjetura "existen infinitas parejas de primos consecutivos"

Posteriormente, el veinticuatro de mayo del año dos mil los más célebres matemáticos del momento se volvieron a reunir en Paris, en la llamada **reunión del milenio**. En esta convención matemática se presentaron siete problemas no resueltos hasta el momento, con la intención de que algún matemático los supiera solucionar. Estos siete problemas fueron escogidos entre muchos, de manera que abarcaran todos los campos de las matemáticas. Tal es la complejidad de estos problemas, que sólo una de las teorías, la conjetura de Poincaré ha sido demostrada, curiosamente el año pasado durante la realización de este trabajo. Posteriormente a esta convención matemática el instituto Clay de Matemáticas (Massachussets, EUA) ofreció un millón de dólares a la persona que resolviera uno de estos siete problemas. De esta manera "nacieron" los llamados siete problemas del milenio, que son los siguientes:

- P versus NP.
- La conjetura de Hodge.
- La conjetura de Poincaré.
- La Hipótesis de Riemann.

- El problema de Yang-Mills.
- El problema de Navier-Stokes.
- La conjetura de Birch y Swinnerton-Dyer.

Uno de los problemas más conocidos de lógica matemática y en ciencias computacionales es determinar si existen problemas decidibles que puedan resolverse mediante comprobaciones algorítmicas, pero en cuyo proceso de solución se tenga más tiempo del usual. Este es el caso del **problema del P vs NP**. Para explicarlos utilizaremos el siguiente ejemplo:

Un día un amigo nos invita a una fiesta. Al llegar nuestro amigo se va y nos quedamos solos, sin teóricamente conocer a nadie. Sintiéndose tímido se pregunta si conocerá a alguien en la fiesta. Nuestro amigo vuelve y nos dice que cree que hay una mujer que nos conoce. Nosotros la miramos y verificamos si la afirmación de nuestro amigo es cierta o no. No obstante, si nuestro amigo no nos hubiera dicho que conocíamos a esa mujer, hubiéramos tenido que realizar una visión exhaustiva de todo el salón hasta dar con ella.

La idea que queremos resaltar con este ejemplo, es que en la mayoría de problemas de decisión, generar una solución es mucho más difícil que comprobar que una solución es correcta. Haciendo un símil con el ejemplo previo, es mucho más difícil ver que hay alguien en esa fiesta que conocemos, que venga alguien y nos diga que hay alguien a quien conocemos.

Este es sólo uno de los muchos problemas de decisión. La relación entre este problema de decisión y el problema del P versus NP, es determinar si hay problemas de decisión como este que puedan ser resueltos mediante sencillas comprobaciones algorítmicas, con ordenadores, pero que en cuyo proceso de solución se requiera un tiempo muy superior al usual. Este es el problema del P versus NP, formulado el año 1971 por Stephen Cook. Este científico inglés afirmó que P eran todos aquellos algoritmos que podían ser resueltos con tiempo polinomico, mientras que en NP estaban determinados todos aquellos algoritmos no determinísticos que emplean tiempo polinómico.

En la teoría de la computación los recursos que se suelen estudiar son el tiempo de ejecución de un programa(o el número de pasos realizados en la construcción de la solución) y el espacio. Sin embargo se podrían estudiar otros recursos como el número de procesadores necesarios para realizar el programa. En computación, cuando el tiempo de ejecución de un algoritmo es menor que un cierto valor calculado previamente mediante el conjunto de variables entrantes usando una fórmula polinómica, se puede concluir que el programa analizado se puede resolver en tiempo polinómico. Cualquier fórmula obtenida que sea un polinomio se podrá afirmar que el programa al que representa es en tiempo polinómico, y están dentro del conjunto P, mientras que si la fórmula obtenida no lo es, como es el caso de las funciones exponenciales, no tardan tiempo polinómico, y están dentro del conjunto NP. Estos últimos no tienen una solución algorítmica, es decir, que ningún procesador podría resolver el programa analizado en un tiempo razonable.

Es importante resaltar que la complejidad de un problema viene dado por el número de pasos que lleva a resolverlo, aunque lógicamente, el número exacto de pasos viene determinado por el procesador, el lenguaje entre muchos otros factores. No obstante, existen dos clases de complejidad: P y NP. En primer lugar, la clase de complejidad P es el conjunto de problemas de decisión que pueden ser resueltos en una máquina determinista en tiempo polinómico. En segundo lugar, la clase de complejidad NP son el conjunto de problemas de decisión no-deterministas que pueden ser resueltos.

Dentro del conjunto NP encontramos una gran cantidad de problemas, aunque nos centraremos en dos: el problema de satisfacibilidad booleana (SAT) y el clásico problemas del buscaminas. Debido a la gran importancia que tienen estos problemas, se han hecho numerosas investigaciones y esfuerzos para encontrar algoritmos que resuelvan alguno de los problemas NP en tiempo polinómico. Dentro del conjunto NP hay un subconjunto de problemas llamados NP-completo en el que se puede transformar polinomialmente en cada uno de los problemas de NP-completo. Este tipo de problemas son los más complicados y muy probablemente no formen parte de la clase de complejidad de P. La razón por la que afirmamos esto, es debido al hecho que de tenerse una solución polinómica para un problema de NP-completo, todos los problemas de NP tendrían también una solución en tiempo polinómico. De esta manera, Stephen Cook concluyó que de tenerse un algoritmo en P para uno de los problemas NP-completo, se tendría una solución en P para todos los problemas de NP. Este científico inglés aportó además que el problema de satisfacibilidad booleana (SAT) es un NP-completo. A partir de esta deducción, se han podido deducir muchos otros algoritmos que son NP-completos. La gran mayoría de estos están en el libro de Garey and Johnson's de 1979 *Computers and Intractability: A Guide to NP-completeness.*

Si como habíamos afirmado anteriormente P es igual a NP, entonces P abarcaría las zonas NP y NP-completo[17]. Así pues, descubriendo un NP-completo, encontraríamos una solución en P para todos los problemas NP. De esta manera, la importancia de encontrar la anterior igualdad, radica en que de encontrarse un algoritmo en P para un problema NP-completo, todos los problemas NP-completos, y consecuentemente todos los problemas de NP, tendrían soluciones en tiempo polinómico. Para la persona que pueda demostrar esta igualdad planteada por Stephen Cook, se loe dará un millón de dólares como premio.

Como se ha afirmado anteriormente, algunos de los problemas más interesantes de NP-completos están directamente con la lógica de proposiciones. De esta manera, analizaremos dos problemas: el primero será el problema de satisfacibilidad booleana (SAT), mientras que el segundo lo analizaremos de una manera mucho más completa, con la inclusión de un algoritmo, el buscaminas.

5.2 Problema de satisfacibilidad booleana (SAT)

Como ya se ha afirmado anteriormente, este problema fue el primero que se pudo demostrar que está dentro del subconjunto NP-completo. Además, a partir de este se han podido deducir que otros problemas están dentro del subconjunto NP-completo. No solamente por su carácter histórico es importante para nuestro trabajo, sino que está directamente relacionado con la lógica matemática.

El problema de satisfacibilidad booleana se pregunta si, para un circuito booleano dado, existe una selección de datos de entrada que produce una salida verdadera. Por lo tanto, este problema nos viene a decir que todo problema lógico, se puede resumir con un conjunto de nexos lógicos. Si alguno de los resultados de esta operación tiene valor cierto, entonces este sistema, para ese valor dado, satisface el problema de satisfacibilidad booleana.

Este hubiera sido un problema muy interesante de resolver mediante un algoritmo propio. Sin embargo, no se ha podido realizarlo debido al poco tiempo disponible, y la gran cantidad de información y temas que se ha tenido que tratar. Así pues, se ha preferido centrarnos en otro programa que se ha creído más interesante, el problema del buscaminas. Este problema es muy parecido al de la satisfacibilidad

[17] Ver anexo 11

booleana, dado que como veremos a continuación, también utiliza los nexos lógicos y sus respectivas operaciones lógicas.

5.3 El buscaminas

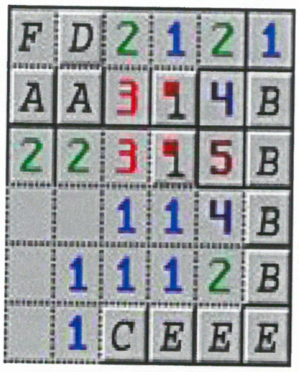
El objetivo de este apartado del trabajo, es ver que un juego tan común como el buscaminas, puede ser resuelto median te unas operaciones lógicas. Así pues, la relación entre el buscaminas y la lógica matematica es profunda, y se puede ver, que una jugada puede ser resuelta mediante un programa, en vez de usar nuestro razonamiento.

El problema que estamos analizando, no es evidentemente la simple resolución de un buscaminas. Es obvio que si tenemos en cuenta que su resolución le aportaría al que lo resuelva un millón de dólares, no es tan trivial. Este problema esta relacionado con el P vs NP. La conexión entre el juego y el problema del millón de dólares, fue explicada por el científico Richard Kaye en su obra *Minesweeper is NP-complete* publicado en el año 2000. ¿Es posible resolver un buscaminas, de un número más que considerable de casillas, mediante una operación lógica y en un tiempo analizable en tiempo polinómico? Antes de profundizar en la cuestión, se describirá primero el juego el buscaminas.

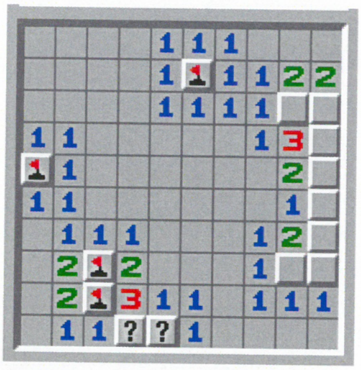
5.3.1 En qué consiste el juego

El buscaminas, o *Minesweeper* en inglés, es un juego de lógica creado en el año 1989 por Robert Donner, aunque actualmente lo conocemos debido al ser uno de los juegos que nos vienen instalados por defecto en cualquier operador Windows. A continuación se explicará en qué consiste este juego, aunque siempre relacionándolo con el programa que se ha desarrollado.

El objetivo de este juego es despejar el tablero sin apretar ninguna mina. Para ello, el jugador deberá apretar encima de ciertas casillas, donde después de apretar, podrá ver si hay una mina. En caso de que la haya, el jugador ha perdido. En caso contrario, el jugador deberá seguir con la partida hasta que o bien acabe, o acabe apretan do una mina. En este juego existen distintos símbolos, y dependiendo de la persona que ha desarrollado el algoritmo, tiene unos símbolos u otros. Podemos encontrar muchos tipos de símbolos, además de las minas y los números, como el punto, el interrogante, la bandera o variables.



1.1 Buscaminas avanzado



1.2 Buscaminas windows

1.3 Nuestro buscaminas

El programa de Windows (figura 1.2) tiene la posibilidad de elegir entre dos tipos de símbolos, la bandera y el interrogante. La bandera representa la casilla donde el jugador está convencido que hay una mina, y el interrogante donde no está muy seguro del símbolo que esconde.

Por otra parte, en el buscaminas avanzado (figura 1.1) podemos poner los mismos símbolos, más variables. Este buscaminas nos ayuda a encontrar las operaciones lógicas del sistema que estamos analizando, de tal manera que su resolución sea matemática. En el caso del sistema anterior (figura 1.1), podemos resolver el sistema de la siguiente manera:

Debido a los doses que se encuentran debajo de A, podemos deducir que A son minas. Una vez que sabemos que en la casilla A hay minas, B deben ser minas también debido al 4 y al 5 que tienen al lado. Por otro lado, podemos deducir que ni E ni D son mina, debido a que si lo fueran, este sistema sería incompatible, inconsistente. Finalmente, y dependiendo del número que se escondiera bajo D, podríamos deducir el valor de F. Aunque en este caso no se ha resuelto el sistema mediante una operación lógica, las variables nos ayudan a formularlas. Sin embargo, podríamos también haber resuelto el sistema mediante una operación lógica, basada en que las dos posiciones con bandera tengan mina[18]:

$$O = (A \cdot (1 - D)) \cdot (B \cdot (1 - E)) \cdot C \cdot F$$

Como podemos comprobar, obtenemos el mismo resultado que habíamos obtenido si resolvíamos este sistema mediante un razonamiento lógico. Así pues, podemos resolver un sistema mediante una operación lógica.

Finalmente, en el buscaminas que se ha desarrollado, podemos poner números, puntos, minas e interrogantes. En el caso de los números, los ponemos cuando nos referimos a las minas que hay en las casillas vecinas, que en el caso en que la casilla analizada no este situada ni en la primera fila ni en la primera columna, podemos llegar a encontrar ocho minas como máximo, y cero como mínimo. En el caso en que la casilla analizada esté en primera fila o columna, se ha de mirar el número de casillas vecinas. En el caso de que el sistema que estuviéramos analizando fuera el anterior, (también mostrado seguidamente)

```
1.1001..1
112111...
012.2....
011111100
000000000
001110000
002x2?111
002x201?1
```

, en la posición (1,1), tenemos tres casillas vecinas, con lo que el número máximo de minas que podemos encontrar alrededor son 3 (no es el caso de este sistema).

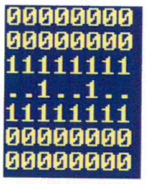
Por otra parte, encontramos el interrogante cuando sabemos que en esa posición hay un número que no sabemos cual es. Así pues, y mediante las minas que hay a su alrededor, se ha de determinar su cantidad. En este caso, no contaremos los puntos, dado que no sabemos aún si hay una mina o no.

De esta manera, también tenemos el punto, en esa casilla donde no sabemos si hay una mina o un número. De esta manera, si determinamos que a esa casilla no le corresponde ninguna mina, sabremos seguro que le corresponde un número.

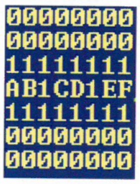
[18] Ver anexo 12 para ver la tabla de verdad correspondiente a esta operación lógica

5.3.2 Azar en el juego

A pesar de saber ya como jugar, se pueden dar casos en que es imposible determinar si en una posición hay mina o no. Es en este punto donde nos aparece un punto de azar en el juego. En el caso



, no podemos determinar en que posición hay mina, y en la posición en la que no la hay. La única condición que podemos afirmar, es que o bien uno de los dos puntos consecutivos es mina, o bien el otro punto lo es, pero en ningún caso puede ser que los dos sean minas, y que ninguno de los dos lo sea. De esta manera, deberíamos de hacer una tabla de verdad para ver los valores según los cuales este sistema puede ser compatible y los valores para los que no lo pueden ser. Para ello, deberemos modificar el archivo de entrada a la siguiente entrada, para realizar una operación lógica:



La operación lógica sería la siguiente,

$$O = (((A \cdot (1-B)) + (B \cdot (1-A))) \cdot ((C \cdot (1-D)) + (D \cdot (1-C))) \cdot ((E \cdot (1-F)) + (F \cdot (1-E))))$$

y su correspondiente tabla de verdad[19]:

Como podemos observar, existen ocho posibilidades según las cuales existe un resultado cierto, con lo que, como habíamos afirmado anteriormente, existe una posibilidad de azar en este juego.

[19] Ver anexo 13 para analizar su tabla de verdad

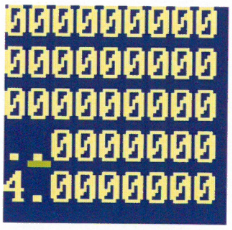
5.3.3 Relación entre el buscaminas y el problema del P vs NP

Un problema de tipo **P** es aquel que se puede resolver dentro de un tiempo de ejecución que no sea mayor que alguna potencia del número de datos iniciales. Lógicamente, un problema que no sea considerado P, es decir **no-P**, no estará dentro de este rango. Además, podemos afirmar, que un problema es de tipo P si encontramos un algoritmo que lo resuelva en tiempo polinómico. Podríamos mencionar una gran cantidad de algoritmos considerados P, como es un algoritmo que ordene una lista de números o de palabras, buscar una cadena de caracteres en un texto, entre muchos otros algoritmos. Sin embargo, y por otra parte, no hay tantos problemas que no se puedan resolver en tiempo polinómico, como parece ser el caso del problema del viajante (encontrar la ruta más corta para realizar un recorrido) o encontrar los factores primos de un número entero, o la encriptación para enviar datos, entre otros programas. No obstante, ninguno de estos programas se ha podido demostrar que es no-P. Esta dificultad en la demostración, radica en el hecho de que no nos podemos basar en ningún algoritmo preexistente correcto. Si quisiéramos demostrar que un problema no es P, deberíamos probar que no se puede resolver en tiempo polinómico.

Hasta la fecha de hoy, el mayor logro en la demostración de que un problema no es P, es demostrar que todos los problemas que no son P, tienen algo en común: si podemos resolver uno de ellos en un tiempo polinómico, entonces demostraríamos que todos los problemas que son no-P se pueden resolver en tiempo polinómico.

Como ya se ha visto en el apartado 5.1 de nuestro trabajo de investigación, el problema de que si P es igual a NP viene a decir que si son todos uno mismo. Muchos científicos creen que no, que P no es igual a Np, aunque si alguien pudiera demostrar esta igualdad, es decir, resolver algún problema nP-completo y demostrar que nP completo es igual a P, y que por fuerza, NP ha de ser igual a P, entonces todos estos científicos estarían equivocados. Por el momento, y debido a que nadie ha conseguido demostrar ninguno, no se cree en la igualdad.

Por otra parte, deberíamos recordar que uno de los problemas NP-completos más conocidos es el problema del SAT. El nexo de este problema con el buscaminas nos deja ver la relación entre el buscaminas y el problema del P vs NP. Así pues, es de suma importancia su análisis. El nexo que los conecta viene dado por la consistencia del buscaminas. Un sistema del buscaminas es consistente cuando todos los símbolos realizan correctamente sus funciones dentro del programa. De esta manera, si tuviéramos el siguiente sistema:



Vemos que este es un sistema incompatible o inconsistente, debido a que en primer lugar, es imposible que hayan cuatro minas porque esta en una esquina y donde el número de minas que puede tener como mucho son tres. Además, sólo tiene tres puntos o minas que hagan que el dígito sea mayor o menor.

Como podemos observar, es sencillo comprobar si un conjunto de minas ser corresponden con los números dados, pero es por el contrario, prácticamente imposible colocar correctamente la misma colocación de las minas que teníamos anteriormente. Este problema ya no consiste en encontrar las minas sino en determinar si un conjunto dado de números forman una partida coherente de buscaminas. En el caso en que nos encontráramos con el anterior sistema, nos daríamos cuenta que el programador habría cometido un error en su desarrollo del programa.

Fue Kaye quien afirmó que el problema de la satisfacibilidad booleana para un circuito booleano dado puede ser codificado como un problema de consistencia de buscaminas para una combinación determinada de números distribuidos en casillas utilizando un procedimiento de codificación que se ejecuta en tiempo polinómico. De esta manera, si pudiéramos resolver el problema de consistencia de buscaminas en un tiempo polinómico, se habría resuelto también el problema del SAT para este determinado circuito. De esta manera, el problema de consistencia de buscaminas es NP-completo.

5.3.4 Algoritmos relacionados con el buscaminas

Durante estos dos años de estudio sobre la lógica matemática, se ha descubierto que aquello que nos habíamos planteado en un principio no era suficiente. Este es uno de los apartados que no teníamos previsto, pero que con el desarrollo del trabajo, se ha visto que sería un punto muy interesante de cara a tener una visión global de la lógica matemática. En este caso, se ha querido ir más allá de la teoría, e intentar dar un paso hacia la resolución del problema analizado en el punto anterior de demostrar la igualdad de P y NP. Para ello, se ha decidido hacer dos algoritmos que analizaremos a continuación.

5.3.5 Consistencia del buscaminas

En la parte anterior del trabajo, se ha resuelto este problema tan complejo, aunque, evidentemente, lo se ha resuelto con un rango de filas y columnas muy pequeño. De esta manera, esta es una primera aproximación a la resolución del mismo. Seguidamente analizaremos el programa[20], aunque si desean lo pueden encontrar en el CD anexado.

Se ha estructurado nuestro primer algoritmo en dos partes: el cuerpo del programa, y un subprograma dentro del mismo. De esta manera, en la parte principal del programa, leemos el archivo de entrada y seguidamente vamos al subprograma llamado *check*. Es en esta parte del algoritmo donde podemos afirmar que es la parte más importante, aunque no esté físicamente dentro de la parte principal del algoritmo. En el subprograma *check* analizamos, mirando los "vecinos" de cada número, cuantas minas hay en su alrededor.

En los programas analizados durante la realización del trabajo de investigación, se ha podido ver que la complejidad a la que se puede llegar para la resolución del mismo es altísima, aunque se ha querido simplificar el problema, eliminando cualquier tipo de símbolo más allá de los números y de las minas. Así pues, se ha querido dejar claro este apartado de consistencia del buscaminas.

5.3.6 Resolución de un buscaminas

Este es el programa principal de este apartado de algoritmos. El algoritmo citado a continuación resuelve un sistema de buscaminas con puntos ('.'), interrogantes ('?'), minas('x') y números (1,2...8). Seguramente la resolución del propio algoritmo sería más sencilla si utilizásemos un concepto informático llamado recursividad, pero debido a nuestra inexperiencia con este código, decidimos enfocar su resolución desde otro punto de vista.

[20] Ver anexo 14 para analizar el código de esta primera aproximación

Para ello nos planteamos diferentes objetivos que teníamos que ir consiguiendo para llegar al punto final: su resolución completa sin recursividad. En primer lugar, tuvimos que determinar si un algoritmo era compatible o no no-compatible. Esta primera parte es el algoritmo anterior. En segundo lugar, tuvimos que simplificar el sistema, debido a que pensamos que una vez finalizado el programa, éste sería muy lento, debido a un factor que explicaremos a continuación. De esta manera, simplificamos el sistema mediante la colocación de las minas donde estuviéramos seguros que en esa posición había una mina. De esta manera, de tener el siguiente sistema,

llegaríamos a otro mucho más desarrollado:

De esta manera, pasamos de tener un sistema con siete incógnitas, a uno de sólo dos. Si relacionamos este último concepto con una tabla de verdad, veremos la gran simplificación que se ha realizado: sabemos que una tabla de verdad esta compuesta por el siguiente número de posibilidades, donde n representa el número de incógnitas:

$$2^n$$

De esta manera, mientras en el primer caso tenemos que el número de posibilidades a analizar son 2^7(128), en el segundo de los casos tenemos que se ha de analizar simplemente 2^2(4). De esta manera ganamos eficacia en nuestro algoritmo[21].

En este caso, la parte más destacada dentro de nuestro programa está dentro del subprograma seguro. Sin embargo, aún no se ha acabado de analizar nuestro programa, aunque paso a paso nos vamos acercando hasta nuestro objetivo. El siguiente paso fue juntar estas dos partes y realizar la parte que anteriormente no habíamos podido realizar debido a la recursividad. Por esta razón, decidimos relacionar aún más este algoritmo con nuestro trabajo, y usar los números binarios. De esta manera, lo que anteriormente no pudimos hacer con la recursividad lo haremos mediante 0 y 1. El siguiente paso fue

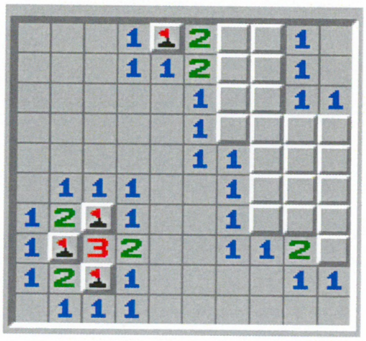
[21] Ver anexo 15 para ver su código

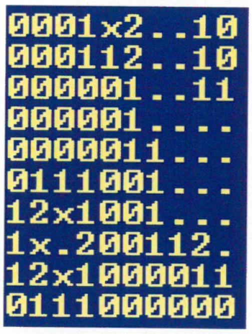
mirar dentro de las posibilidades que aun teníamos, cuales podían tener una mina y ser compatibles. Para ello, utilizamos los números binarios, que nos permitieron poner 1 donde había minas y 0 donde no había. Lo que hicimos fue poner un número a cada variable, y de esta manera llegar hasta el número de variable mayor en número binario. Una vez pasado el número de la variable a binario, substituíamos los unos por minas, y mirábamos si era compatible. Si lo era, lo guardaba en una tabla de dos dimensiones llamada "solu". De esta manera conseguimos resolver el problema del buscaminas[22].

5.3.7 Análisis comparativo entre nuestro algoritmo y el de Windows

Aunque nuestro algoritmo no tenga la función de juego, podemos comparar nuestro algoritmo con el que nos viene instalado por defecto en Windows, dado que son una complementación muy buena. Lo que haremos para ver que nuestro algoritmo funciona correctamente, será avanzar en el buscaminas de Windows, y una vez veamos que podemos resolver el problema mediante nuestro algoritmo lo haremos.

1[er] paso:



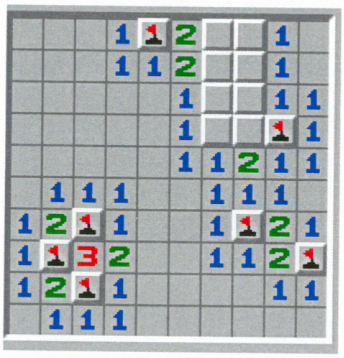


Lógicamente, si analizamos desde este punto un buscaminas, es probable que haya muchas soluciones dentro del buscaminas. Es por esta razón, que aún deberíamos profundizar más para poder llegar a un buen resultado, pero gracias a este paso que se ha dado, nos aseguramos las posiciones donde seguro que hay mina, y las que seguro que no las hay, los interrogantes, que recordemos, pueden solo esconder números.

[22] Ver anexo 16 para ver el código final de nuestro programa

```
0001x2..10
000112..10
000001..11
000001x...
0000011...
0111001??.
12x1001x??
1x?200112x
12x1000011
0111000000
```

2º paso:

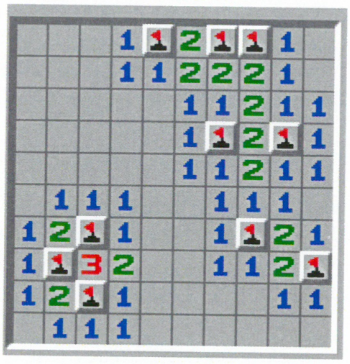


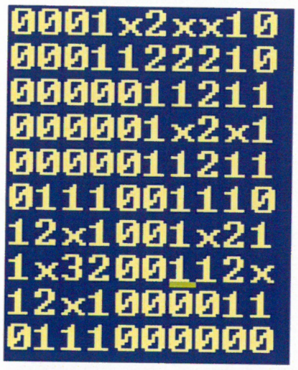
```
0001x2..10
000112..10
000001..11
000001..x1
0000011211
0111001110
12x1001x21
1x3200112x
12x1000011
0111000000
```

Aún ahora tenemos que el sistema podría tener más de una solución, con lo que deberíamos profundizar más. No obstante, podemos observar que se ha resuelto bastantes incógnitas. La salida que obtenemos es la siguiente:

```
0001x2..10
000112..10
000001??11
000001x?x1
0000011211
0111001110
12x1001x21
1x3200112x
12x1000011
0111000000
```

3er paso:





Vemos que tenemos el buscaminas resuelto. De esta manera vemos que podemos resolver un buscaminas sencillo, es decir, de un número limitado de filas y columnas, mediante este algoritmo. Sin embargo, el programa que se ha realizado, y el que muchos otros han desarrollado, descubren las casillas que parecen ser más evidentes, y no hay ningún algoritmo que haya sido desarrollado, que dado un cuantioso número de filas y columnas, destapar todas las casillas. Aunque pueda parecer imposible, así es, y la persona que pueda resolver este algoritmo de manera completa, se llevará el premio del millón de dólares.

5.4 Valoración personal

Esta ha sido la última parte analizable referente a la lógica matemática dentro de este trabajo de investigación. Sin embargo, ha resultado ser, como en el último apartado, muy interesante aunque a la vez complicado.

En primer lugar, nos ha resultado muy interesante, debido a que anteriormente conocía vagamente, y a causa de la mención por parte de la Sra. Silberstein en una clase de "Raonament matemàtic", cuales eran estos siete problemas del millón de dólares. Pues bien, ahora podemos afirmar que se ha analizado profundamente uno de ellos, y saber cuales son los demás y de qué tratan, aunque de manera muy general. De esta manera, se ha analizado el problema del millón de dólares de demostrar la igualdad entre los P y los NP. Esta ha sido, seguramente, la parte más teórica de este último apartado, pero nos ha servido para entender la relación entre este problema y el buscaminas.

Ha sido precisamente este tema, uno de los que nos han parecido más curiosos dentro del trabajo de investigación. Decimos que nos ha parecido curioso, dado a que la mayoría de los que estamos acostumbrados a jugar a este "juego" (ahora ya no lo consideramos tanto un juego, sino un problema de lógica matemática que se ha convertido en un gran juego de entretenimiento), no nos habíamos planteado nunca la posibilidad de que tuviera relación con la lógica matemática. Muchos pensábamos que era un juego simplemente de lógica, dado que lo podíamos resolver mediante el uso de la razón y del azar. Sin embargo, se ha aprendido que mediante una operación lógica, podemos resolver el problema con menos pasos de los iniciales. Además, ahora tenemos un algoritmo que nos ayuda a resolver este problema.

En relación al algoritmo que se ha desarrollado para la elaboración de este trabajo, ha sido muy bonito, dado que durante la realización de esta parte del trabajo ya no hacíamos informática, y, sinceramente, era una de las materias diferentes a las demás. No obstante, ha sido un poco frustrante, dado que al plantearme el problema, creía que lo podría resolver. Sin embargo, y con el transcurso del tiempo, me fui dando cuenta que la manera en que me lo había planteado no era la adecuada porque necesitaba utilizar la recursividad. Precisamente fue este concepto informático, uno de los que no me quedó claro el año pasado. De esta manera, y viendo que no nos salía el programa tal como habíamos previsto, tuvimos que hacer un cambio en el planteamiento. Con este cambio pude completar el algoritmo.

6. Incidencias

En esta parte del trabajo analizaremos las incidencias que se ha tenido a lo largo de estos dos años de realización del trabajo de investigación. Es seguramente, la parte más personal del mismo, y lo que le hace distinguirse de un libro científico.

Durante el transcurso de estos dos años de trabajo en el campo de la lógica matemática, se ha podido descubrir una gran cantidad de nuevos aspectos que no conocíamos. Un claro ejemplo de ello, es si miramos el índice inicial con el que presentamos en este trabajo. De esta manera, se ha podido descubrir mucha más información de la que creíamos conocer.

Estos nuevos aspectos que se ha tratado en el trabajo que anteriormente no conocíamos, han sido desarrollados después de analizar e indagar en libros e información relacionada con Internet. De esta manera, se ha encontrado una gran cantidad de información tanto en Internet, como en libros que se ha podido consultar. Aunque pueda parecer mentira, esta fue nuestra primera gran dificultad, la de seleccionar la información necesaria que posteriormente analizaríamos en el trabajo. Esta gran cantidad de información que se ha encontrado, ha sido un obstáculo en un doble sentido: en primer lugar, debido a la gran cantidad de información, que a veces nos hacía pensar que todo aquello que no explicásemos era relevante, y que deberíamos de explicarlo todo. No obstante, vimos que en cada libro, fuente que encontrábamos habían temas constantes. De esta manera, seleccionamos la información más relevante para nuestro trabajo. El segundo grado de dificultad dentro de este campo tan extenso, fue la complejidad de los artículos y los libros que encontrábamos. La lógica matemática no es un tema frecuente en los estudiantes de bachillerato, y tuvimos que indagar sobre la información que encontrábamos, debido a que muchas veces no entendíamos el significado de una fórmula. Esta ha sido seguramente la gran dificultad en que nos se ha encontrado.

Posteriormente, y medida que íbamos entendiendo los fundamentos de la lógica matemática, y debido a que queríamos dotar a nuestro trabajo de una gran parte práctica, empezamos a montar las primeras maquetas. Esta parte, apasionante pero muy difícil, resultó en muchos momentos estresante, dado que nunca antes habíamos realizado una maqueta de ese estilo. La práctica más complicada de montar fue la de la persiana, dado que no podíamos colocar los finales de carrera de manera que la persiana al tocarlos parase. No obstante, y debido al hecho de que la parte visual de la persiana no era muy bonita, al final, conseguimos colocarlos y finalizar esta parte práctica. Por otra parte, la práctica más sencilla de montar fue el scalextric, porque este era un juego al que de niños todos habíamos jugado. No fue tan sencillo, por el contrario, montar la parte de circuitos eléctricos. Es en esta parte de las prácticas, donde el Sr. Valls nos enseñó como se montaba

Respecto al programa que utilizamos para programar el autómata, también tuvimos algunas incidencias, dado que al principio utilizamos el programa *LogoSoft Comfort*, aunque tuvimos que cambiar este programa por el *step 5*, dado que con el primero no funcionaba el autómata. Además, y respecto al autómata, tuvimos que cambiar el que el Sr. Valls me proporcionó, dado que era un poco antiguo, y sólo funcionaba con un operador de Windows 95. Al no poder conseguir ningún software de Windows 95, dado que todos los ordenadores actuales funcionan con un software más avanzado, tuvimos que conseguir otro que funcionara con un Windows más avanzado. No obstante, no se acabaron aquí los problemas, dado que el autómata programable que nos proporcionaron no leía el lenguaje, el código de ninguno de los dos programas que teníamos, y que habíamos aprendido a programar. Finalmente, pedimos un autómata compatible con un Windows avanzado y que pudiera leer alguno de nuestros programas. El autómata que nos proporcionaron, funcionaba sólo con el *Step 5*, por la cual cosa, fue en el que profundizamos más. Debido a estos errores que cometimos a la hora de conseguir el autómata, perdimos una gran cantidad de tiempo respecto a la parte práctica. No obstante, mientras no conseguíamos el autómata, montamos todas las prácticas, y al recibir el definitivo, sólo tuvimos que comprobar que las habíamos montado y programado correctamente.

Mientras montábamos las prácticas, íbamos leyendo y analizando puntos interesantes para nuestro trabajo. Esta fue la parte más divertida del trabajo, dado que me estaba adentrando en un mundo fascinante, y fundamental para todos nosotros en la actualidad. Afortunadamente, no tuvimos ninguna incidencia más, hasta llegar a una de las partes realizada íntegramente por nosotros, dado que queríamos ver una relación que creíamos muy interesante, y que, sin embargo, no estaba en ningún otro trabajo. Estamos hablando de la parte de 5 de nuestro trabajo, titulada *posibilidades de resultado con nexos lógicos*. Recordemos, que en esta parte de nuestro trabajo de investigación queríamos analizar, si había alguna combinación que no se pudiera dar nunca. Aunque pudiera parecer un trabajo sencillo, no es así dado que en un principio dábamos por hecho de que era posible encontrar una operación lógica que nos diera un resultado determinado. Efectivamente, ahora podemos concluir que esto es así. En un principio, y de las diez y seis que queríamos analizar, descubrimos catorce. De esta manera, nos faltaban dos que no podíamos descubrir de ninguna de las maneras. De hecho, y una vez analizado el problema, nos faltaba sólo una, dado que la otra era su negada. Nos faltaba por encontrar, quizás, las dos más sencillas de todas. Sin embargo, y como ya se ha afirmado anteriormente, intentamos resolver este problema de manera matemática, y hubiéramos tenido que resolverlo de manera racional. Nos faltaban las operaciones de A y no A. Como ya se ha afirmado, estábamos pensando en operaciones en las que tuviéramos que combinar operaciones y variables, y no nos dimos cuenta que la operación que buscábamos estaba a nuestro alcance.

Una vez llegamos a la segunda parte de nuestro trabajo, la parte teórica, es donde estuvimos más tiempo. De hecho estuvimos desde la incorporación como tutor del Sr. Alemany hasta hace relativamente poco tiempo, hasta finales de diciembre. Si estuvimos tanto tiempo, es debido a que es la parte más compleja de entender de esta parte de las matemáticas. Tuvimos que analizar, (re)leer más de una vez partes enteras de libros debido a su dificultad. Además, y como ya se ha explicado en este trabajo, los análisis, artículos o cualquier tipo de fuente no está destinada a los alumnos de bachillerato, tuvimos que esforzarnos mucho más para una buena comprensión. Sin embargo, íbamos avanzando en el trabajo de una manera mucho más rápida del previamente previsto. Este hecho nos permitió concentrarnos durante más tiempo en la parte final de este apartado teórico, la parte del análisis del problema del ¿P=NP? Debido a nuestro desconocimiento en la vida interna de las matemáticas, es decir, el mundo de los matemáticos, desconocíamos los siete problemas de millón. Tan solo conocíamos uno, debido a que coincidió que uno de ellos, la conjetura de Poincaré fue resuelta por un extravagante científico. Debido a todo el debate que se produjo, la Sra. Silberstein nos habló sobre este problema. Fue por esta razón, que nos interesamos aun más en este problema del millón, que, aunque lógicamente no lo fuéramos a resolver, queríamos dar una aproximación.

Durante la realización del algoritmo que resolvía el buscaminas, tuvimos que volver a empezar cuando vimos que no teníamos suficiente conocimiento de recursividad para desarrollar nuestro programa de esta manera. Así pues, volvimos a empezar planteándolo desde otra perspectiva. En este caso si que pudimos resolver el problema mediante este nuevo enfoque.

Durante el desarrollo de todo este trabajo se ha tenido una serie de problemas, incidencias, se ha cometido errores, se ha tenido que esperarnos más de cuatro meses a que nos pudieran dar el autómata ideal para nuestras prácticas...Sin embargo, todos estos problemas se han podido solucionar de alguna manera o de otra, para acabar realizando este trabajo. Podemos afirmar que alguna de estas incidencias han sido positivas para la realización de este trabajo de investigación, dado que sin alguna de ellas, no habríamos podido conocer algunos aspectos de la lógica, que, aunque finalmente no hayan acabado en el trabajo, nos han servido para conocer aun más este campo fundamental. De esta manera, los errores han servido para solidificar este trabajo.

7. Posibles líneas de investigación

Como veremos en el siguiente apartado del trabajo de investigación, la lógica matemática no se acaba en simples operaciones lógicas, sino que estas operaciones, nos permiten realizar multitud de funciones y nuevas tecnologías, para suplir las limitaciones del ser humano.

Durante todo este trabajo de investigación, se ha podido observar que la lógica y la computación van estrechamente unidas. Es por esta razón, que creemos que la computación evolucionará hacia donde evolucione nuestra razón, es decir, hacia donde nos guíe nuestra conciencia. En una de las conferencias asistidas en el curso de segundo de bachillerato, el Dr. Amat (Catedrático en la Universidad Politécnica de Cataluña (UPC) responsable del grupo de investigación de robótica inteligente) argumentaba que actualmente, l única limitación que tienen los desarrolladores de robots, de programas inteligentes, es el propio ser humano.

En este sentido, las limitaciones que aporta el ser humano son dobles. En primer lugar, que no sabemos que más inventar, dado que no consumimos toda la capacidad que nos pueden proporcionar las nuevas tecnologías. De esta manera, con los microprocesadores, actualmente hay de mucha mayor capacidad, de la que realmente podemos llegar a utilizar. De esta manera, ¿para qué seguir desarrollando productos que no hacen falta? En segundo lugar, las limitaciones de los seres humanos, es que hay gente que está en contra de la (r)evolución de las nuevas tecnologías, por miedo a que estas les puedan sacar el sitio de trabajo en un futuro.

De esta manera, lo que queremos decir con esto, es que hay múltiples salidas dentro de este campo, pero estas salidas dependen de nosotros mismos, si las queremos aprovechar o no. Sin embargo, podemos afirmar también, que teniendo en cuenta el ritmo de desarrollos de las nuevas tecnologías, esta (r)evolución parece no tener límites, dado que estamos en constante crecimiento, y nuestros limites parecen no tener fin.

Una línea que se podría seguir, seria desarrollar el problema de la satisfacibilidad booleana (SAT) y analizarlo más detalladamente. Además, se podría realizar un algoritmo que nos ayudara a entender mejor este concepto.

Parece pues, que los seres humanos han visto en las nuevas tecnologías, una manera de escapar de sus límites como seres vivos, e intentar emular a ese ser superior creador de vida. Aunque pueda parecer muy exagerado, podríamos afirmar que gracias al desarrollo de las matemáticas de Boole, y el posterior análisis de las mismas, nos han llevado hasta la situación actual, donde parece que vivamos en tan poco espacio.

Posibles líneas de investigación hay múltiples[23] y muy variadas. Un claro ejemplo sería un posible desarrollo de una máquina que al entrar en un cine, en un hospital, en un museo, o en general aquellos sitios donde los teléfonos móviles deberían estar desconectados, desarrollar un aparato que nos dijera que tenemos el móvil

[23] Ver anexo 1(entrevista con el Dr.Chicón)

encendido. Esta era una de las aplicaciones prácticas que queríamos realizar, aunque finalmente no lo desarrollamos debido a la poca relación que le acabamos viendo al trabajo. Como ya se ha argumentado anteriormente, este es sólo un ejemplo de lo que podemos llegar a desarrollar en un futuro.

La verdad es que se han hecho una gran cantidad de ensayos sobre las nuevas tecnologías y los límites de la misma. Todos ellos concluyen que el ser humano se ha convertido en dependiente de éstas. Sólo hace falta el mirar el nuestro alrededor, para darse cuenta de que efectivamente, estamos completamente influenciados y limitados por las nuevas tecnologías. Sin embargo, ¿esto es un hecho por el cual deberíamos estar preocupados, o, por lo contrario, deberíamos estar tranquilos? Personalmente, deberíamos estar tranquilos, dado que las nuevas tecnologías hacen nuestra vida más sencilla, más segura, y, en general, más práctica. De esta manera, el desarrollo de nuevas tecnologías es el futuro de este planeta.

8. Conclusiones

Una vez analizados todos los puntos que se ha creído indispensables para el entendimiento y la profundización en la materia, creemos que necesitamos explicar las conclusiones que podemos sacar de nuestro estudio. De esta manera, intentaremos recapitular toda la información tratada anteriormente, y elaborar una serie de conclusiones que nos permitan acabar de entender los conceptos fundamentales de la lógica matemática.

Si analizamos los fundamentos que se ha ido argumentando en cada apartado de este trabajo de investigación, veremos que **la lógica matemática se ha convertido en uno de los campos más importantes**, sino el que más, de las ciencias. De esta manera, y aunque pueda parecer contradictorio, unas teorías que, en principio, no iban a servir de mucho a la comunidad científica, han acabado siendo el tronco que sostiene todo el árbol de las nuevas tecnologías actuales. Además, parece mentira que **de unas teorías tan poco prácticas, se haya podido llegar a unas conclusiones tan prácticas**.

Así pues, podemos concluir, que a partir de unas operaciones tan básicas y comunes como el **"Y", "O" y "No" se ha desarrollado la sociedad actual**. De esta manera, a partir de unos nexos lógicos que aprendemos a utilizar cuando somos niños, se ha sostenido nuestras vidas actuales, aunque, estos nexos lógicos estén **escritos en lenguaje matemático**.

En relación al lenguaje que se ha utilizado, y como ya habíamos citado anteriormente, se ha aprendido la **gran cantidad de lenguajes** que se utilizan para lo que parece que es una misma materia. Dependiendo en la materia que trabajemos, utilizaremos una simbología u otra. Este hecho nos hace reflexionar sobre la **cantidad de materias y campos que se han desarrollado a partir de la lógica matemática**, como la electrónica y la parte matemática de la lógica.

El **desconocimiento social** por esta parte de las matemáticas **es enorme**, y deberíamos poder extender estas ideas matemáticas a toda la sociedad, de manera comprensible.

Uno de los objetivos que nos planteamos al comienzo de este trabajo, era entender las aplicaciones prácticas de la lógica matemática. Una vez analizado todo este trabajo de investigación y de análisis, podemos concluir que, actualmente, **podemos hacer una infinidad de cosas con nexos lógicos y con sus respectivas operaciones**. En nuestro caso, se ha desarrollado una persiana y una puerta de un garaje en una misma construcción, y lo único que se ha tenido que cambiar ha sido el programa.

Aunque las **puertas lógicas más conocidas** son "Y", "O" y "No", **podemos deducir otras** a partir de estas mismas, como son las puertas lógicas derivadas, "XOR", "NOR", "NAND" y "XNOR".

En relación a las operaciones que se pueden realizar mediante operaciones lógicas, podemos afirmar que podemos encontrar todas las opciones de resultado. De esta manera, sabemos que **mediante una operación lógica matemática, podemos**

encontrar el resultado deseado. Podemos concluir también, que **existen unos números** tales que, como pasa con las operaciones matemáticas algebraicas, nos den como resultado **0 y 1**. Las operaciones que nos dan estos resultados son las siguientes:

$$A \cdot (1 - A)$$

$$(1 - A) + A$$

De esta manera, si analizamos el primer caso, vemos que el resultado de multiplicar una variable por su conjugado, es decir, por su valor negado, es siempre 0. Por otra parte, podemos encontrar una operación, la segunda, que nos de cómo resultado 1. Si tenemos que no A o A, siempre será valor cierto.

Podemos concluir también, que de la misma manera que en álgebra existen **propiedades para las operaciones**, en lógica también, dado que tenemos la propiedad conmutativa, la distributiva, la complementación el principio de dualidad, y las leyes más importantes dentro de esta parte de las matemáticas: las leyes de Morgan. Se ha analizado todas estas leyes, y podemos concluir que son necesarias para cohesionar la lógica matemática.

Finalmente, podemos concluir que **todo problema NP-completo, que pueda ser resuelto en tiempo polinómico**, es decir, en tiempo de una función polinómica, no exponencial en relación a sus variables, **resolverá todos los demás problemas en tiempo no polinómico**. De aquí radica su importancia, y el hecho de que el instituto Clay haya ofrecido un millón de dólares a la persona que lo pueda resolver. Se ha podido apreciar su dificultad con el análisis del problema, y el desarrollo de un algoritmo que resuelva un buscaminas.

Así pues, y teniendo en cuenta todos estos aspectos que se ha podido concluir, y todos los aspectos anteriormente se ha visto, podemos entender su importancia, y el desconocimiento general por la materia. Sin embargo, podemos concluir que su conocimiento dentro de la sociedad es muy bajo, aunque todos la utilizamos cotidianamente: **todas las nuevas tecnologías funcionan mediante operaciones lógicas que resuelven en ceros y unos**.

9. Glosario

Álgebra de Boole..55
Aristóteles..19
Auguste De Morgan..21
Autómatas..38
Buscaminas..80
Buscaminas y PvsNP..84
Circuitos eléctricos..35
Complementación..59
Conjunto..65
Conjunto universal..66
Cuantificador universal..71
Cuantificador existencial..71
Decibilidad..76
Diferencia..68
Elemento..66
Elemento cero y unidad..59,73
Elemento lógico "no"..18
Familias de conjuntos..70
Final de carrera..40
Georg Cantor..65
George Boole..21
Grafos..70
Igualdad de conjuntos..67
Igualdad de subconjuntos..67
Incompletness theorem..76
Intersección..68
Leyes de Morgan..60
Lógica..15
Lógica informal..15
Lógica formal..15
Lógica matemática..26
Nexos principales..30
Y..30

O..31
No..33
Notación por extensión..67
Notación por compresión..67
NP..85
Operaciones binarias..29
P..85
Persiana..39
Postulados de Huntington..60
Problema del P vs NP..78
Principio de dualidad..60
Producto cartesiano..70
Propiedad conmutativa..55
Propiedad distributiva..56
Proposiciones..16
Atómicas..17
Moleculares..17
Proyecciones..70
Puerta garaje..43
Puertas lógicas derivadas..48
XOR..48,73
NOR..49
NAND..50
XNOR..51
Pulsador..40
PvsNP..78
Razonamiento Lógico..15
Regulación por intermitencia..44
René Descartes..20
Reunión del milenio..77
Revolución científica..20
Revolución digital..21
SAT..80
Scalextric calculadora..47

Sensor........26
Silogismo........15
Simbología DIN........38
Y........38
O........38
No........38
Superconjunto........67
Tablas de ayuda........64, 70,72
Tabla de verdad........27
Teoría de conjuntos........65
Términos de enlace........18
Y........18
O........18
Si..entonces........19
Unión........67

10. Bibliografía

ALMANSA, J (2006). *Fundamentos de lógica matemática y computación*. Editorial Sanz y torres.

BINEFA, X (1998) *Lógica computacional*

BLANCHE, R, DUBUCS, J (1996). *La logique et son histoire*. Collection "U", Série philosophique.

COUTURAT, L (1914) *L'algèbre de la logique*. Gauthier-Villars et Cie

CROSSLEY, J.N (1983). *¿Qué es la lógica matemática?* Cuadernos de filosofía y ensayos. Editorial Tecnos[24]

CUENA, J. (1985) *Lógica informàtica*. Editorial alianza

HILL, S, SUPPES, P (2004). *Introducción a la lógica matemática*. Editorial Reverté

KAC, M (1968).*Mathematics and Logic: retrospects and prospects*. Colección: A britanica perspective

LIARD, L. *Logique*. Paris : Masson et Cie

PABION, J.F (1976)*Logique mathematique*. Colección Hermann

SAINZ, M.A (1987) *Matemáticas. Lógica, conjuntos, geometría, estadística*. Editorial crítica.

SÁNCHEZ, P (1986). *Álgebra de Boole, lógica elemental y métodos de demostración matemática*. PPU[25]

http://www.uv.es/ivorra/Libros/Logica.pdf (libro que se puede consultar por internet)

http://educ.queensu.ca/~compsci/resources/BoolLogic/titlepage.html

http://www.math.csusb.edu/notes/sets/boole/boole.html

http://crpuzzles.com/logic/

http://perplexus.info/

http://math.sfsu.edu/smith/Documents/BooleanLogic.PDF

http://www.cs.nyu.edu/courses/summer03/G22.2340-001/lectures/lecture01.ppt

[24] Disponible en la biblioteca de la escuela
[25] Disponible en la biblioteca de la escuela

http://cnx.rice.edu/content/m10514/latest/

http://www.csc.liv.ac.uk/~konev/COMP210/logic1.ex.pdf

http://www.educajob.com/xmoned/temarios_elaborados/filosofia/De%20la%20l%F3gica%20cl%E1sica%20a%20la%20l%F3gica%20simb%F3lica.htm

http://web. mat.bham.ac.uk/R.W.Kaye/minesw/minesw.htm

http://www.cecm.sfu.ca/personal/jborwein/algorithms.html

http://es.wikipedia.org/wiki/Tiempo_polin%C3%B3mico

http://es.wikipedia.org/wiki/NP

http://es.wikipedia.org/wiki/NP-completo

http://es.wikipedia.org/wiki/Buscaminas

http://es.wikipedia.org/wiki/Problema_de_satisfacibilidad_booleana

http://es.wikipedia.org/wiki/Teor%C3%ADa_de_conjuntos

http://www.claymath.org/Popular_Lectures/Minesweeper/

http://www.claymath.org/millennium/P_vs_NP/

http://www.computer.org/cise/articles/Top_Algorithms.htm

La entrada en las esencias, mediante
Operaciones en los razonamientos,
Grupos de nexos,
Indiscutibles verdades,

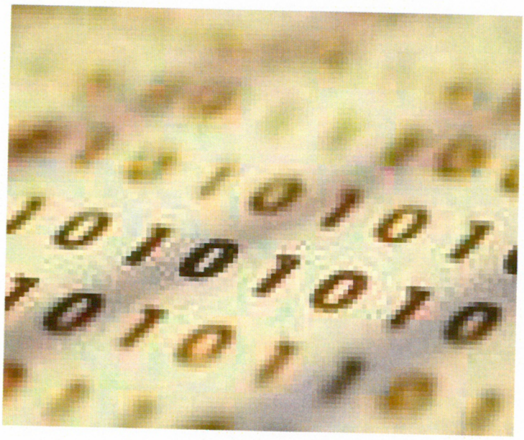
Con algo razón y lógica, para

Acabar operando

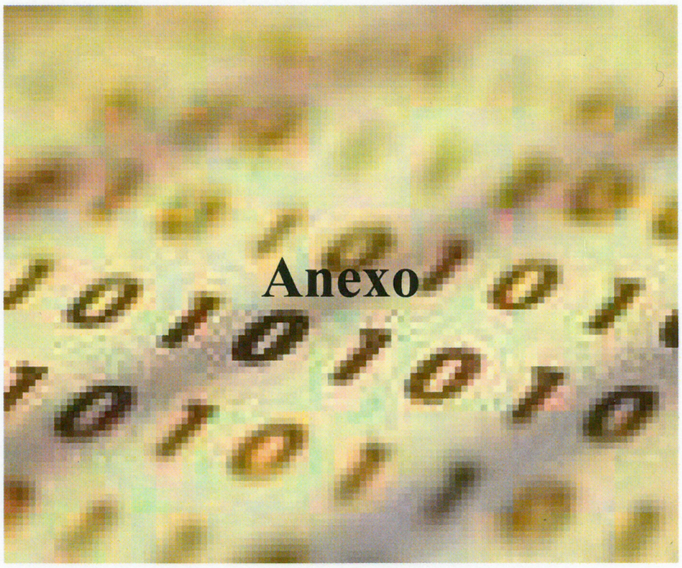
-

Mirar el brillo de tus razonamientos
Al ver dos operaciones unidas,
También al mirar tus tablas,
Espacios que llenan en mi alma;
Mar de infinitas curiosidades
Acotadas por blancas
Trazas de mi mente, que dejan
Imposibles movimientos
Capaces de abrir y cerrar las operaciones
Analizadas por expertos, sabios de nuestros tiempos

Poema propio



Lógica matemática: Álgebra de Boole y análisis práctico



Trabajo de investigación

4a Edició Premi Poincaré 2007

Segon premi

Lógica matemàtica. Álgebra de Boole y análisis práctico

Per les diferents aplicacions pràctiques que fa de les quals destaquem la presentació acurada en diferents formats, incloent una pàgina web.

Autor: Román Sandoval Villamora

Centre: Aula Escola Europea (Barcelona)

Tutor: Sr. Joan Alemany

La lógica desempeña, por su carácter de lenguaje básico para la representación del conocimiento, un papel fundamental en esa tendencia. En sus orígenes, la lógica se inició como una ciencia dedicada a la identificación de las formas humanas de razonamiento, con el objetivo de crear criterio para discernir las discusiones filosóficas de los antiguos griegos.

Lógica informática, *José Cuena*

La misma limpieza de las matemáticas le privad e inocencia. Hay algo turbio en tanta claridad, por arbitrarios que sean sus postulados.[...]La lógica y la matemática son en esencia juego, un bello juego cuyo truco consiste justamente en su misma ausencia de truco. Ese e su engaño y por él se justifican.

Eugenio Trías

Índice

1) Entrevista con el Dr. Chicón, gerente del centro de visión computerizada..............7

2) Imágenes de la persiana..12

3) Tabla de verdad de la práctica de la persiana...16

4) Imágenes de la puerta del garaje..18

5) Imágenes del depósito de agua...19

6) Cronograma del depósito de agua...21

7) ¿Cómo funciona un detector de nivel?..23

8) Imágenes del scalextric...26

9) Diagrama de la primera ley de Morgan...28

10) Diagrama de la segunda ley de Morgan..29

11) Diagrama para ver los tipos de problemas...30

12) Tabla de verdad del buscaminas...31

13) Tabla de verdad correspondiente al problema del azar...................................34

14) Código de esta primera aproximación...37

15) Código del programa "seguro"..39

16) Código de la resolución final del buscaminas..41

1) Entrevista con el Dr. Chicón, gerente del centro de visión computerizada

Uno de los objetivos en el trabajo de investigación, es ver las aplicaciones prácticas que tiene la lógica matemática, o más concretamente, la lógica binaria. Es a causa de esta última razón, que se decidió visitar el centro de inteligencia artificial del campus de la Universidad Autónoma de Barcelona, en Bellaterra. Más concretamente, el centro de visión por computador. Para concertar esta entrevista, se tuvo que entrar en la página web del centro de visión computerizada, http://www.cvc.uab.es, y buscar una persona que trabajase en este campus. Por esta razón, se contactó con el Sr. Agustín Chicón, gerente del centro de visión computerizada y que, muy amablemente, concedió la entrevista. La entrevista tuvo lugar en el mes de Septiembre de 2006, una vez se creía que teníamos el material suficiente como para poder realizar una entrevista.

El centro de visión computerizada (CVC) es un consorcio fundado en el año 1995 por el departamento de industria, la DURSI i la UAB. Es un edificio de 2000m^2 dedicados a la investigación y el desarrollo en el campo de la visión por computador. Así pues, se dedican a investigar, pero a la misma vez, ofrecen sus conocimientos a las empresas que requieran de sus servicios. Además, está compuesto por los medios de tecnología más punteros en el campo de la visión artificial. Teniendo en cuenta todas estas características, se decidió hacer una visita al CVC, y pedirle al Sr. Agustín Chicón, que muy amablemente accedió a ser entrevistado, que respondiese a unas preguntas antes de hacer la visita al centro.

Cabe resaltar, que se tuvo la fortuna de hacer la entrevista el mismo día en qué los licenciados en este centro, presentaban sus proyectos finales. Así pues, se pudo ver lo último en nuevas tecnologías. Es por esta razón, que la mayor parte de la entrevista fue la explicación de los trabajos, donde, se pudo observar, que efectivamente, tal y como nos había anunciado el Sr. Chicón, no hay tanta diferencia entre Japón y Estados Unidos, con España, en el campo de la visión por computador. Un claro ejemplo de ello, es que muchos de los alumnos en este centro, hacen conferencias y cursos en todo el mundo, desde Tokio hasta Nueva York. Además, el Sr. Chicón insistió repetidamente, que todas las clases que se dan en el CVC, son en inglés, hecho que produce que muchos de sus alumnos sean extranjeros. Así pues, la primera idea que el Sr. Chicón quiso dejar clara, fue que no hay diferencia alguna con países como Japón o Estados Unidos. Otro claro ejemplo, es el trato con grandes multinacionales como Ausonia, Balay, Boston Scientific, Citroën, Sony, entre muchas otras grandes empresas mundialmente conocidas.

El primer tema tratado en la entrevista, fue las actividades realizadas en este centro. El Sr. Chicón afirmó que en el CVC hay dos líneas de trabajo: la investigación y el desarrollo. Mientras algunos están trabajando en descubrir nuevas tecnologías, hay otros que desarrollan productos para empresas. Por otra parte, hay un sistema académico en el centro, donde los alumnos, ya titulados, pueden hacer un postgrado en visión por computador. En referencia a la investigación y el desarrollo, hay diferente vías: el análisis de documentos, los sistemas de ayuda a la conducción, el reconocimiento y análisis de movimiento, textura y color. En referencia al tema de desarrollo de productos para las empresas, el Sr. Chicón explicó las fases de este desarrollo. En primer lugar, se produce un primer contacto entre la empresa y el CVC, donde se plantea el problema de la empresa, y se trata de buscar un diagnóstico. En el caso de que este proyecto sea factible, se inicia el proyecto. No obstante, y si el proyecto resulta imposible en términos logísticos, o de poca viabilidad, se realiza un estudio de viabilidad, donde se concluye si este proyecto es factible o no. Si es factible, se realiza un prototipo.

Por otra parte, otro de los temas tratados fue el de la visión por computador. Según el Sr. Chicón, es una disciplina científica que pretende desarrollar modelos y sistemas computacionales que sean capaces de explicar y de duplicar las capacidades del sentido de la visión natural. Además, se puede aplicar a prácticamente todos los ámbitos de la actividad humana y nos abre una infinidad de posibilidades de aplicaciones sólo limitadas por la imaginación. El Sr. Chicón, nos propuso el ejemplo de que la retina ocular podría ser considerada un sensor. El ojo completo, una cámara. Finalmente, el ser humano completo, podría ser considerado un ordenador. En relación a este aspecto, se le propuso al Sr. Chicón que si teniendo en cuenta la espectacular evolución de la tecnología en los últimos años, si el creía que algún día los robots podrían substituir por completo a un ser vivo. El Sr. Chicón afirmó que a pesar de que actualmente muchas empresas hayan substituido algunos puestos de trabajo por autómatas, él no cree posible que algún día los robots hayan substituido por completo a los seres vivos. No obstante, afirmó que es posible que aquellos trabajos más repetitivos, como el de trabajar en una fábrica de material para los coches, por ejemplo, es posible que para aumentar la producción y el capital de la empresa, éstos decidan instalar robots. No obstante, este hecho salva el trabajo a muchos otros trabajadores de la empresa, dado que si no fuera porque se instalan robots, estas empresas tendrían que deslocalizar a otros países más atractivos en nivel económico, con mano de obra más barata. Además, se pueden llegar a mantener algunos puestos de trabajo, dado que alguien ha de vigilar el trabajo de las máquinas. Además, un robot nunca podrá tener la imaginación que tiene un ser humana. De esta manera, los trabajos como arquitecto o director de una empresa, no podrían ser realizados por un robot.

Otra de las cuestiones surgidas en esta entrevista, fue las aplicaciones del CVC. El Sr. Chicón, afirmó, muy orgullosamente, que trabajan con muchos hospitales para ayudar a descubrir nuevas formas de tratamientos, o maneras para analizar una enfermedad. Una de las investigaciones más recientes, era una especie de pastilla, aunque en realidad era una cámara. De esta manera, el paciente se tragaba esta "pastilla", y los médicos podían observar como estaba su intestino, su hígado, etc. Por otra parte, también trabajan con muchas empresas de construcción, y pudimos ver las aplicaciones posibles de las nuevas tecnologías en este campo. Por ejemplo, nos impresionó particularmente, un proyecto donde podíamos dibujar a mano alzada la vista superior de una habitación, y el programa nos representaba en 3-D la habitación. De esta manera, el arquitecto podría ver antes de dibujar la casa al detalle, como quedaría esa habitación. Este programa lo llamaron HDICAD (Hand drawn input for CAD systems).

En el campo de la seguridad, se han desarrollado grandes aplicaciones, como es el caso de la biometría, de la detección i reconocimiento de movimientos, o el reconocimiento de objetos. En este caso, nos pusieron el ejemplo de la previsión de un movimiento por parte de una persona. Teniendo en cuenta el recorrido realizado previamente, y las imágenes faciales, hay un algoritmo que intuye el siguiente movimiento de una persona. Este caso es un gran paso para la seguridad en bancos o establecimientos, dado que si podemos intuir el siguiente paso de una persona, nos podemos anticipar a ella.

Durante la entrevista nos fueron surgiendo muchas preguntas que el Sr. Chicón nos respondía muy amablemente, como cuando le preguntamos que cual era la función de la computación y la automatización en los sistemas de visión. Él nos respondió que juegan un papel fundamental. Además, es gracias al desarrollo de las nuevas tecnologías que se pueden hacer todos estos descubrimientos y desarrollos para empresas. Por otra parte, afirmó que cuando Boole estudió la lógica, y propuso su teoría sobre la lógica matemática, no lo hizo con la intención de que actualmente pudiéramos estar desarrollando productos para casi todas las cosas. No obstante, las personas que le siguieron en el tiempo, y decidieron tener en cuenta sus teorías, y fueron desarrollándolas, gracias a este desarrollo que hoy tenemos todos estos electrodomésticos de última generación. Si uno de ellos hubiera ido por otro campo, en vez de intentar desarrollar las teorías de Boole, no sabemos cual habría sido el curso de la historia.

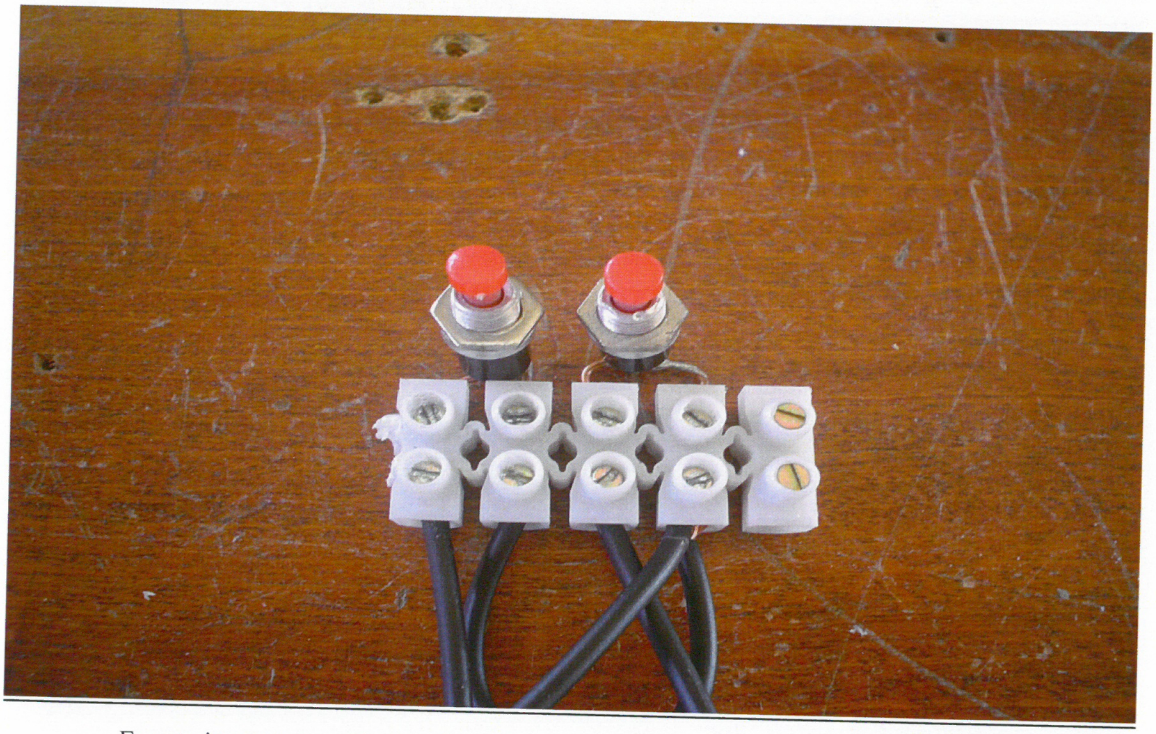
Como ya hemos afirmado anteriormente, coincidió nuestra visita con la exposición de los trabajos realizados por los alumnos del CVC. El Sr. Chicón, nos propuso de ir a ver los trabajos de estos alumnos. A pesar de que muchos de ellos no los podíamos entender, debido a su gran complejidad, intuimos la mayoría. Muchos de los trabajos, eran los que durante la entrevista el Sr. Chicón me había ido comentando. Con el que más nos impresionamos, fue con uno que teniendo en cuenta la posición de un vehículo, podía intuir cual era el horizonte. De esta manera, esto podría ser un gran avance en el campo de la seguridad en los coches. Otros trabajos no iban tanto en el campo de la investigación de nuevas tecnologías, y prefirieron desarrollar otro tipo de programas como es el caso de un programa que analiza los rasgos faciales de cada persona, e intenta establecer una relación, a partir de todas las caras que ha visto hasta el momento, entre ellas. Todos estos trabajos, nos los iba explicando con una gran satisfacción. En todos y cada uno de ellos, había un detalle que para él era algo espectacular. Por otra parte, había todos los alumnos mirándonos dado que estábamos viendo sus trabajos poco antes de que los presentaran a los profesores, y evidentemente, estaban muy nerviosos. Al detenernos en cada trabajo se oía un silencio para escuchar la opinión que tenía sobre el trabajo, el Sr. Chicón.

Al cabo de una hora y media después de haber entrado a entrevistar al Sr. Chicón, éste tuvo que marcharse debido a que tenía una reunión con los demás profesores para analizar cada trabajo. Así pues, tuvimos que marchar. Durante el camino de vuelta de la UAB, íbamos reflexionado sobre lo que acabábamos de ver. Lo que más nos sorprendió de esta visita fue el hecho de que según el Sr. Chicón, España no está tan lejos en el campo de la visión por computador que otros países como es el caso de Estados Unidos o Japón. Si hubiéramos tenido que situar a España en una posición en el ranking de países, lo hubiéramos situado por debajo de otros países, como Francia o Alemania. Después de esta visita, le dimos mucha más importancia de la que anteriormente le habíamos dado a Boole, dado que él es el precursor de que actualmente hoy en día, haya autómatas que puedan intuir el horizonte, cosa impensable en la época de Boole. No obstante, hemos de recordar que Boole no fue el único, pero fue el primero.

2) Fotografías y explicación de la persiana

Los instrumentos que se ha necesitado para realizar esta práctica son:

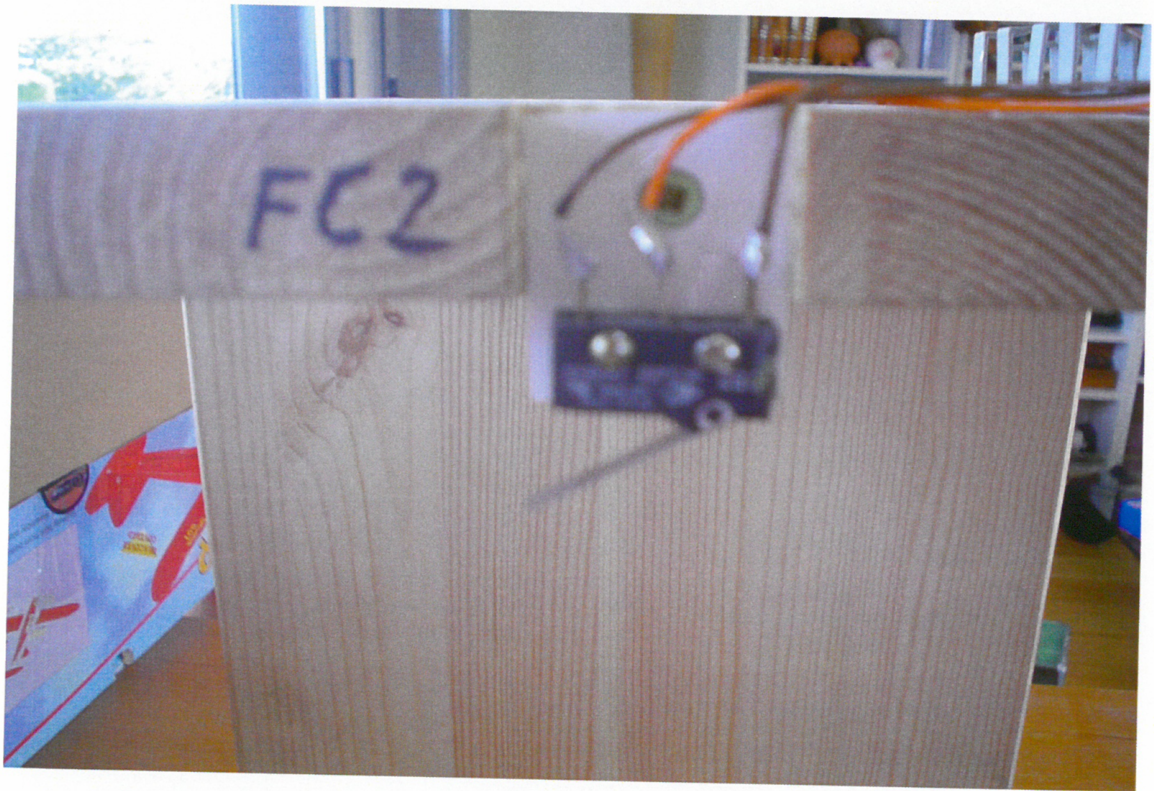
- madera (para construir la base de la persiana)
- cable de cobre (para realizar el cableado)
- pulsadores
- un trozo de tela (como persiana)
- Una barra de aluminio y dos de plástico
- Finales de carrera, superior e inferior



En esta imagen se pueden ver los dos pulsadores que utilizaremos para esta práctica.

En esta imagen se puede ver la persiana físicamente, con los dos finales de carrera, uno arriba y el otro abajo, dos carriles para que baje la persiana, y el cableado que sale del motor de la persiana (arriba a la derecha de la imagen)

En esta fotografía se puede ver la cortina que representa la persiana. Vemos también, el peso que ha de haber en la parte inferior de la cortina, para que cuando toque con el final de carrera, produzca sufciente fuerza como para apretarlo.



En esta imagen se puede ver el final de carrera superior de la persiana y su cableado.

En esta imagen podemos ver el final de carrera inferior de la persiana.

3) Tabla de verdad de la práctica de la persiana

Esta tabla de verdad corresponde a una persiana. Contiene cuatro elementos de entrada, y dos de salida. Estos elementos de entrada corresponden a la función PS (para que la persiana suba), FCS (final de carrera superior), PB (para que la persiana baje), y, finalmente, FCI (final de carrera inferior). De esta manera, se pueden observar todas las posibilidades que puede tener una persiana para subir, bajar, o que esté tocando cualquiera de los dos finales de carrera. Además, el programa de la persiana ha de contener un sistema de seguridad, para el cual, el usuario no pueda resultar dañado después de un cortocircuito provocado por la persiana. Este cortocircuito pude llegar a ocurrir, cuando el usuario apreta tanto el botón de subida como de bajada a la vez. Si no existiera este sistema de seguridad, se produciría un cortocircuito. Por otra parte, esta tabla de verdad tiene dos elementos de salida: S(si tiene valor 1 y B valor 0, la persiana subirá), y B(si tiene valor cierto la persiana bajará, siempre y cuando S no tenga valor cierto). Finalmente, se puede ver en la tabla de verdad una situación imposible (físicamente hablando), dado que no es viable que tanto el final de carrera superior, como el inferior tengan valor cierto. En otras palabras, no es posible que la persiana esté a la vez totalmente bajada y totalmente subida.

PS	**FCS**	**PB**	**FCI**	$S = PS \cdot (1 - FCS) \cdot (1 - PB)$	$B = PB \cdot (1 - FCI) \cdot (1 - PP)$
0	0	0	0	**0**	**0**
0	0	0	1	**0**	**0**
0	0	1	0	**0**	**1**
0	0	1	1	**0**	**0**
0	1	0	0	**0**	**0**
0	1	0	1	**0**	**0**
0	1	1	0	**0**	**1**
0	1	1	1	**0**	**0**
1	0	0	0	**1**	**0**
1	0	0	1	**1**	**0**
1	0	1	0	**0**	**0**

PS	**FCS**	**PB**	**FCI**	$S = PS \cdot (1 - FCS) \cdot (1 - PB)$	$B = PB \cdot (1 - FCI) \cdot (1 - PP)$
1	0	1	1	**0**	**0**
1	1	0	0	**0**	**0**
1	1	0	1	**0**	**0**
1	1	1	0	**0**	**0**
1	1	1	1	**0**	**0**

4) Imágenes de la práctica

Para realizar esta práctica, se ha utilizado la persiana como soporte, y la única cosa que se ha tenido que cambiar, además del programa, ha sido, que hemos tenido que colocar una barrera fotoeléctrica, para emular cuando pasan coches. Si no hubiera esta barrera, los coches pasarían cuando estuviera cerrando y podrían tener un accidente, dado que la puerta de garaje seguiría bajando mientras el coche estuviera pasando. Los instrumentos son:

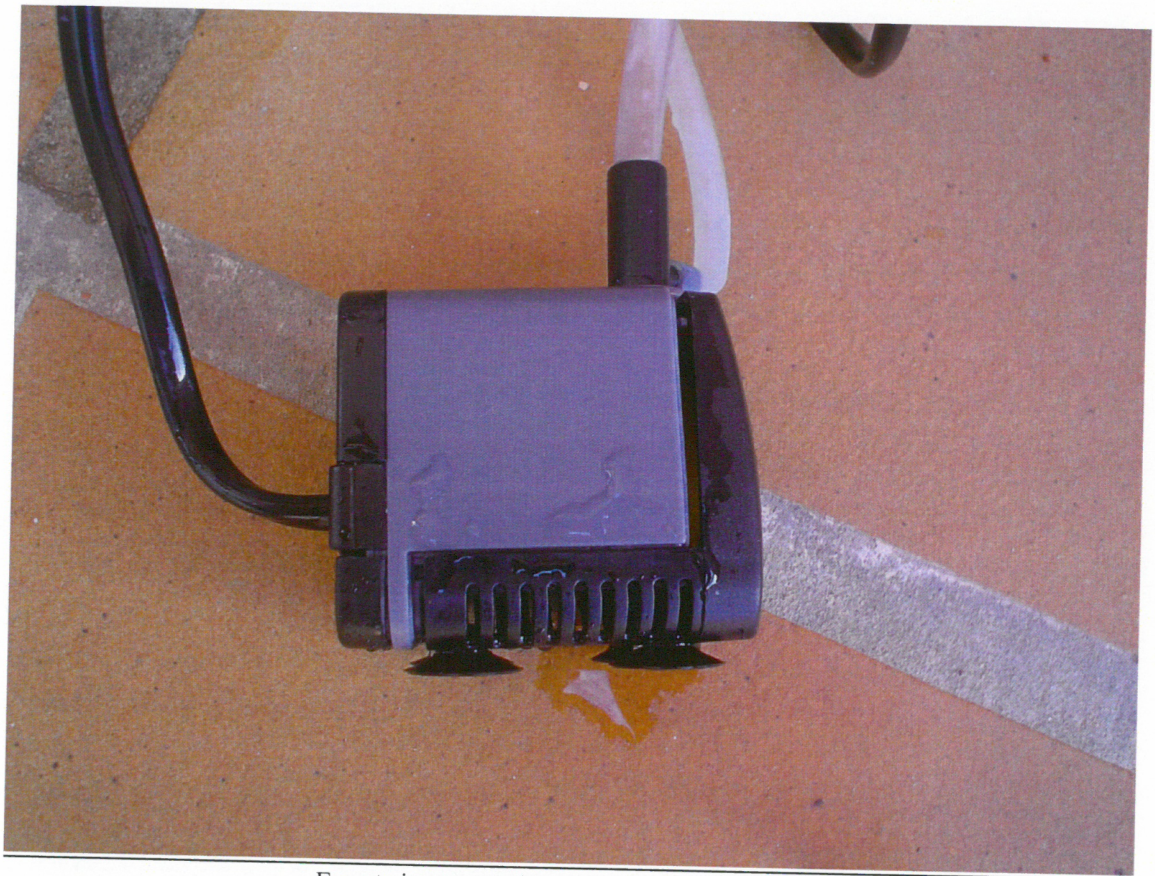
- madera (para construir la base de la puerta)
- cable de cobre (para realizar el cableado)
- pulsadores
- un trozo de tela (como puerta de garaje)
- Una barra de aluminio y dos de plástico
- Finales de carrera, superior e inferior
- Barrera fotovoltaica

En esta imagen se puede ver la barrera fotovoltaica, que consta de dos instrumentos: un espejo y la propia barrera. La barrera necesita el espejo, para saber hasta donde llega el espacio donde puede pasar la puerta de garaje.

5) <u>Imágenes de la práctica de la bomba de agua</u>

El material necesario para realizar esta práctica ha sido:

a. Bomba de agua
b. Dos recipientes de plástico
c. Cable de cobre
d. Bomba de agua



En esta imagen podemos ver la bomba de agua.

En esta imagen podemos ver una visión general de la práctica.

6) <u>Cronograma del depósito de agua</u>

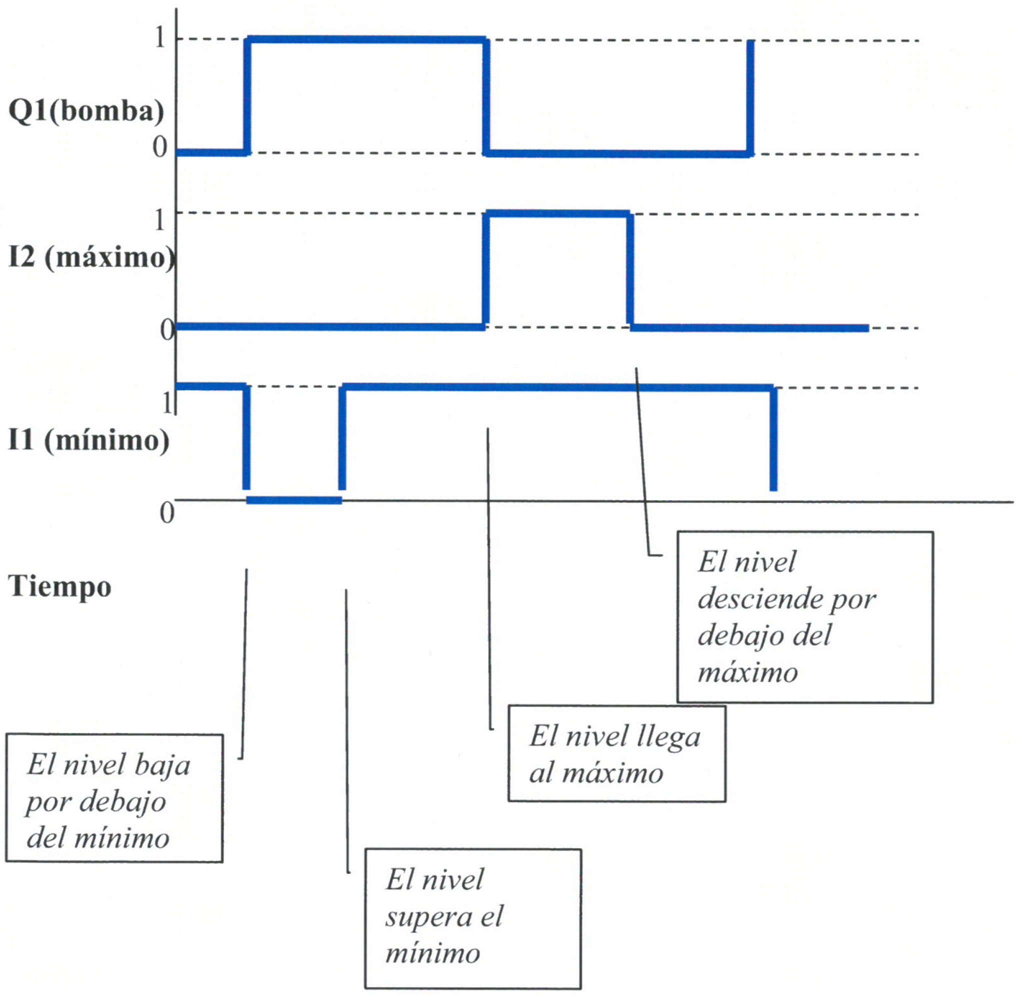
En un cronograma podemos ver diferentes situaciones representadas bajo un mismo grafico. En este caso, se puede observar la situación de la bomba de agua (encendida o apagada), según diferentes situaciones, es decir, el nivel del agua. Así pues, podemos ver que cuando el agua está en el nivel mínimo, la bomba de agua se enciende, y no se apagará hasta que llegue al máximo. Entonces la bomba volverá a agapagarse, es decir, volverá a tener valor 0.

I1: detector de nivel mínimo (I1=1 si el agua está por encima del nivel mínimo)

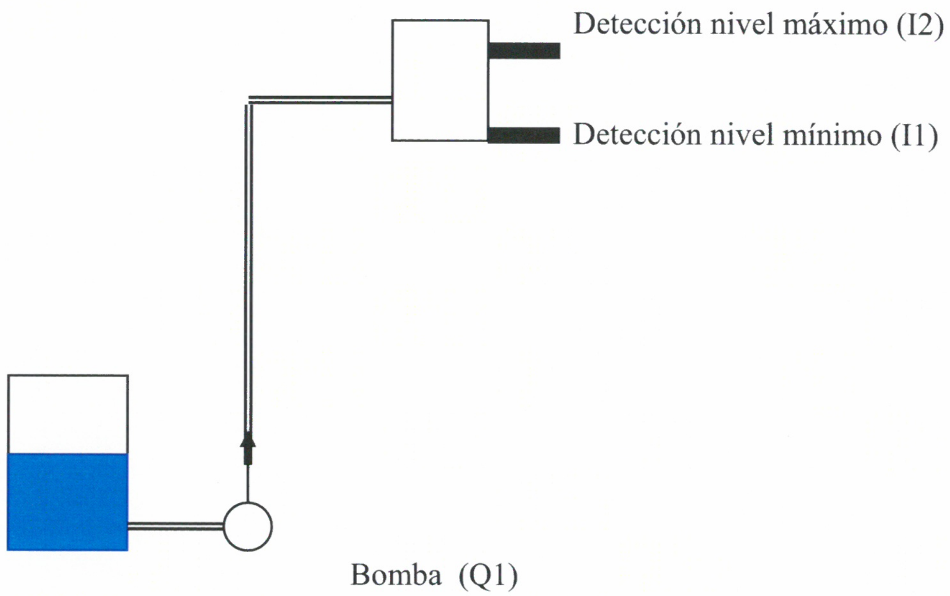
I2: detector de nivel máximo (I2=1 si el agua está por encima de este nivel)

Q1: motor bomba (Q1=1 si la bomba está en marcha)

(Cronograma).



En el siguiente esquema podemos observar el montaje práctico de la maqueta:



7) ¿Cómo funciona un detector de nivel?

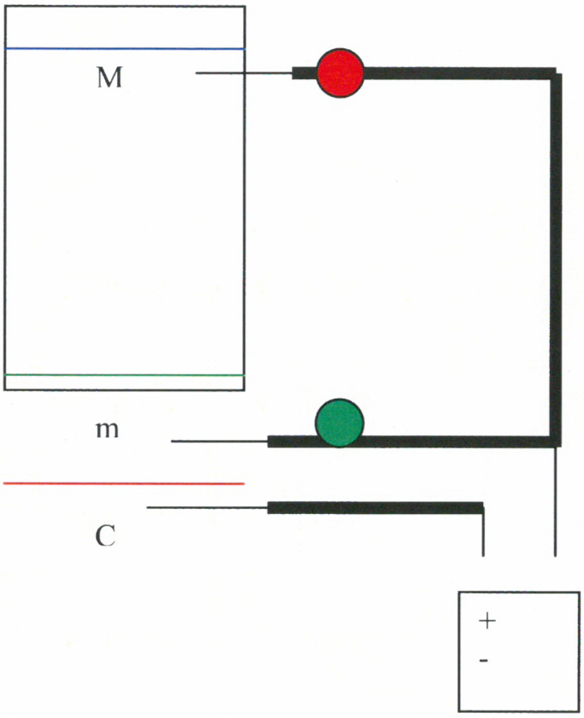
Un detector de nivel es un aparato que sirve para detectar la capacidad de algún elemento. De esta manera, hay detectores de nivel del agua, de gas, de petróleo, incluso existen detectores de nivel de malaria. En nuestro caso, analizaremos tres tipos de detectores de nivel: un detector que funciona mediante la continuidad eléctrica del agua, por flotación y por ultrasonidos.

En primer lugar, se analizará el caso de un detector de nivel por continuidad eléctrica a través del agua, mediante unas sondas situada en el interior del depósito. Este detector de nivel, es el que se usará en la práctica, dado que es el mejor para realizar esta práctica. En la siguiente imagen podemos observar tres situaciones distintas en relación a los detectores de nivel. En primer lugar, cuando el agua está en el nivel rojo. En este caso, y como no hay continuidad eléctrica, dado que no hay agua ni en el máximo ni en el mínimo, los dos detectores nos indicaran valor 0. De esta manera, el detector de nivel nos dirá que el agua está por debajo del mínimo, y sabremos que hay que empezar a llenar el depósito.

En segundo lugar, y en el caso en que el agua esté en el nivel verde, el detector verá que está por encima del mínimo, pero aún no habrá llegado al máximo, con lo que deberá seguir llenando el depósito. De esta manera, m=1, y M=0, dado que habrá continuidad eléctrica entre el mínimo y el agua. De esta manera, vemos que es mediante la continuidad del agua que el detector nos indica el nivel del depósito.

Finalmente, y el último caso a analizar es cuando haya continuidad eléctrica en todo el depósito. En ese momento, y debido a la continuidad eléctrica, el depósito sabrá que ha llegado a su máxima capacidad, y mandará apagar la bomba de agua. De esta manera, y en este último caso obtendremos que m=1 y M=1, con lo que la bomba se apagará, hasta que no vuelva a haber continuidad en todo el depósito, es decir, que m=M=0.

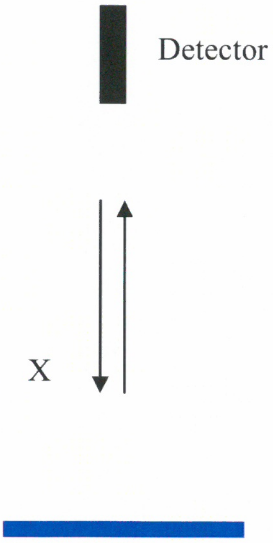
Para saber si la bomba de agua está apagada o encendida, existen unos LED (pequeñas superficies luminosas) que nos permiten observar si la bomba está o no en funcionamiento. Si los LED verde y rojo están apagados, la bomba debe ponerse en marcha. La bomba debe detenerse cuando los dos LED estén encendidos a la vez.



En las aplicaciones reales se utilizan detectores más complicados para conocer el nivel de líquido en un depósito. Existen detectores que funcionan por flotación y, en el caso de líquidos inflamables, como la gasolina, hay detectores por ultrasonidos.

Esquema de un detector de nivel por flotación

Los detectores por ultrasonidos envían una onda sonora dirigida hacia la superficie de líquido y miden el tiempo que tarda en llegar la reflexión del sonido. De esta manera pueden calcular la distancia entre el punto donde se encuentra la sonda y la superficie del líquido:



$$V = \frac{2x}{t} \rightarrow x = \frac{v \cdot t}{2}$$

8) Imágenes de la práctica de la persiana

Para realizar esta práctica del scalextric, se necesita:

- Un circuito y dos coches de scalextric
- Barrera fotovoltaica para contar las vueltas
- Cable de cobre para realizar el cableado

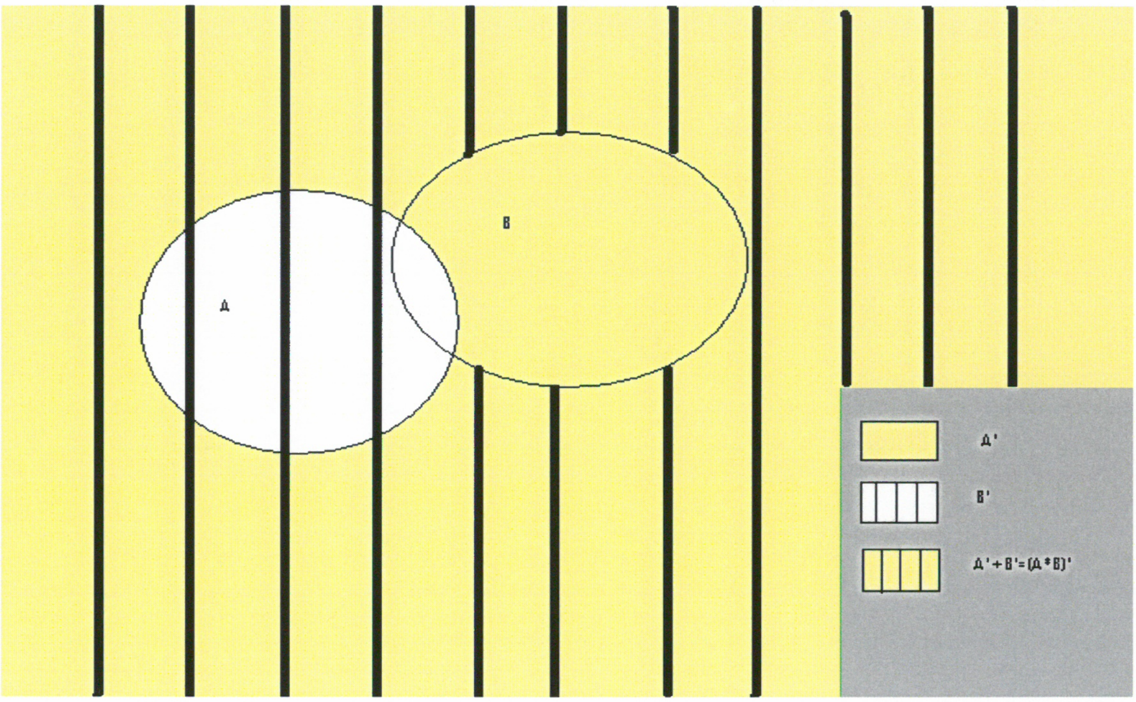
Fotografía tomada durante el montaje teórico del cableado

Fotografía tomada durante el montaje práctico de la práctica

Fotografía tomada durante la conexión del scalextric con el autómata

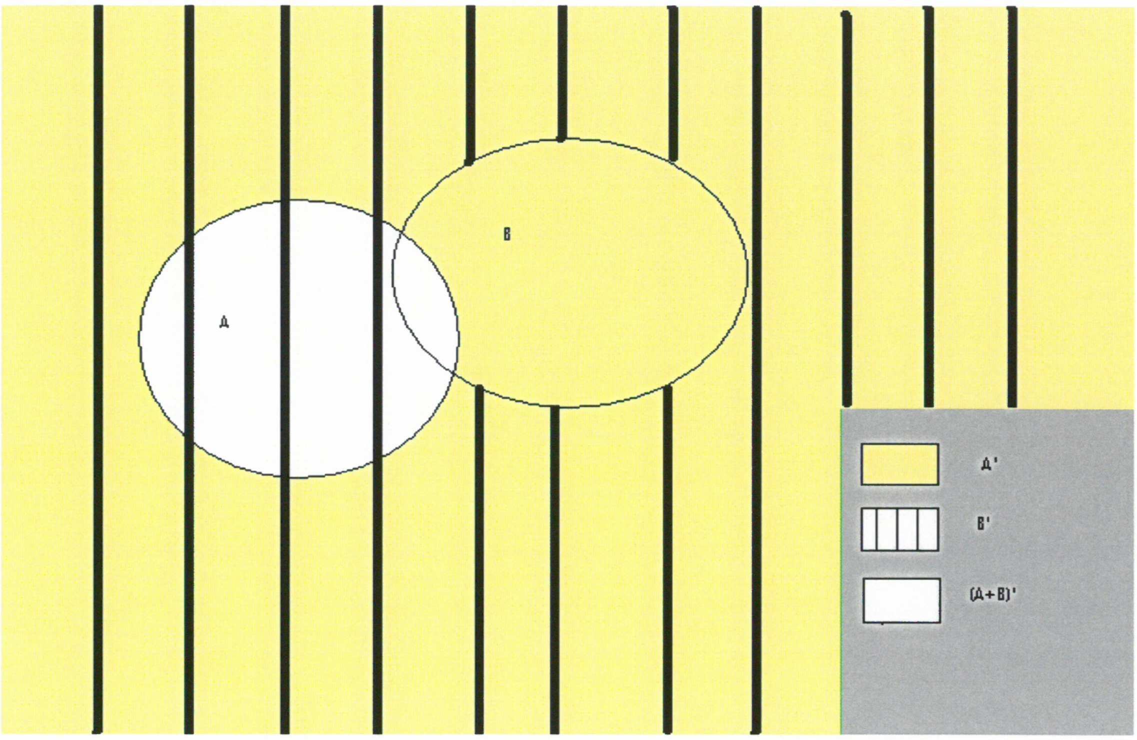
9) Diagrama de la primera ley de Morgan

En el siguiente diagrama se puede ver una demostración alternativa de la primera ley de Morgan. En la leyenda se puede ver lo que representa cada espacio.



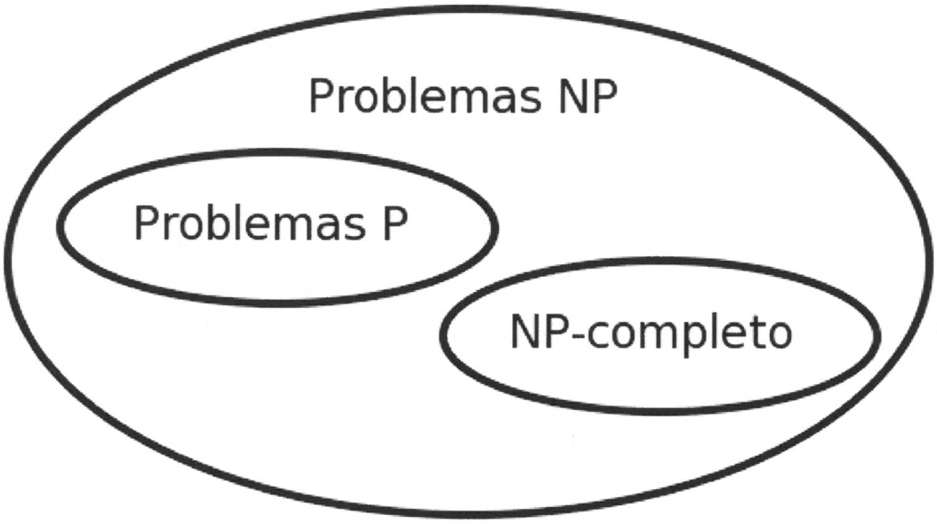
10) Diagrama de la segunda ley de Morgan

En este diagrama se puede ver una demostración alternativa de la segunda ley de Morgan.



11) Diagrama para ver los tipos de problemas

Como se ha dicho anteriormente en el trabajo, existen tres tipos de problemas: los problemas que se pueden resolver en tiempo polinómico, los que no, y los NP-completos. Comos se puede ver en la siguiente imagen, los problemas P y NP-completo son subconjuntos de los problemas NP. Así pues, resolviendo cualquier problema NP-completo se podrán resolver cualquier tipo de problema NP.



12) Tabla de verdad del buscaminas

En la siguiente tabla de verdad se pueden ver todas las posibilidades de resultado del problema del buscaminas que se está analizando en el trabajo. Cada una de las variables corresponde con una posición del buscaminas, mientras que O, representa el resultado de la operación. Se recuerda que $O = (A \cdot (1-D)) \cdot (B \cdot (1-E)) \cdot C \cdot F$.

A	B	C	D	E	F	O
0	0	0	0	0	0	**0**
0	0	0	0	0	1	**0**
0	0	0	0	1	0	**0**
0	0	0	0	1	1	**0**
0	0	0	1	0	0	**0**
0	0	0	1	0	1	**0**
0	0	0	1	1	0	**0**
0	0	0	1	1	1	**0**
0	0	1	0	0	0	**0**
0	0	1	0	0	1	**0**
0	0	1	0	1	0	**0**
0	0	1	0	1	1	**0**
0	0	1	1	0	0	**0**
0	0	1	1	0	1	**0**
0	0	1	1	1	0	**0**
0	0	1	1	1	1	**0**
0	1	0	0	0	0	**0**
0	1	0	0	0	1	**0**
0	1	0	0	1	0	**0**
0	1	0	0	1	1	**0**
0	1	0	1	0	0	**0**
0	1	0	1	0	1	**0**
0	1	0	1	1	0	**0**
0	1	0	1	1	1	**0**

A	B	C	D	E	F	O
0	1	1	0	0	0	**0**
0	1	1	0	0	1	**0**
0	1	1	0	1	0	**0**
0	1	1	0	1	1	**0**
0	1	1	1	0	0	**0**
0	1	1	1	0	1	**0**
0	1	1	1	1	0	**0**
0	1	1	1	1	1	**0**
1	0	0	0	0	0	**0**
1	0	0	0	0	1	**0**
1	0	0	0	1	0	**0**
1	0	0	0	1	1	**0**
1	0	0	1	0	0	**0**
1	0	0	1	0	1	**0**
1	0	0	1	1	0	**0**
1	0	0	1	1	1	**0**
1	0	1	0	0	0	**0**
1	0	1	0	0	1	**0**
1	0	1	0	1	0	**0**
1	0	1	0	1	1	**0**
1	0	1	1	0	0	**0**
1	0	1	1	0	1	**0**
1	0	1	1	1	0	**0**
1	0	1	1	1	1	**0**
1	1	0	0	0	0	**0**
1	1	0	0	0	1	**0**
1	1	0	0	1	0	**0**
1	1	0	0	1	1	**0**
1	1	0	1	0	0	**0**
1	1	0	1	0	1	**0**
1	1	0	1	1	0	**0**

A	B	C	D	E	F	O
1	1	0	1	1	1	**0**
1	1	1	0	0	0	**0**
1	1	1	0	0	1	**1**
1	1	1	0	1	0	**0**
1	1	1	0	1	1	**0**
1	1	1	1	0	0	**0**
1	1	1	1	0	1	**0**
1	1	1	1	1	0	**0**
1	1	1	1	1	1	**0**

13) Tabla de verdad correspondiente al problema del azar

En este caso, como en el anterior, las variables de entrada corresponden a las distintas posibilidades que se han de analizar, mientras que O corresponde a la salida, que es el resultado de la siguiente operación lógica:

$$O = (((A\cdot(1-B))+(B\cdot(1-A)))\cdot((C\cdot(1-D))+(D\cdot(1-C)))\cdot((E\cdot(1-F))+(F\cdot(1-E))))$$

A	B	C	D	E	F	O
0	0	0	0	0	0	**0**
0	0	0	0	0	1	**0**
0	0	0	0	1	0	**0**
0	0	0	0	1	1	**0**
0	0	0	1	0	0	**0**
0	0	0	1	0	1	**0**
0	0	0	1	1	0	**0**
0	0	0	1	1	1	**0**
0	0	1	0	0	0	**0**
0	0	1	0	0	1	**0**
0	0	1	0	1	0	**0**
0	0	1	0	1	1	**0**
0	0	1	1	0	0	**0**
0	0	1	1	0	1	**0**
0	0	1	1	1	0	**0**
0	0	1	1	1	1	**0**
0	1	0	0	0	0	**0**
0	1	0	0	0	1	**0**
0	1	0	0	1	0	**0**
0	1	0	0	1	1	**0**
0	1	0	1	0	0	**0**
0	1	0	1	0	1	**1**
0	1	0	1	1	0	**1**
0	1	0	1	1	1	**0**
0	1	1	0	0	0	**0**

A	B	C	D	E	F	O
0	1	1	0	0	1	**1**
0	1	1	0	1	0	**1**
0	1	1	0	1	1	**0**
0	1	1	1	0	0	**0**
0	1	1	1	0	1	**0**
0	1	1	1	1	0	**0**
0	1	1	1	1	1	**0**
1	0	0	0	0	0	**0**
1	0	0	0	0	1	**0**
1	0	0	0	1	0	**0**
1	0	0	0	1	1	**0**
1	0	0	1	0	0	**0**
1	0	0	1	0	1	**1**
1	0	0	1	1	0	**1**
1	0	0	1	1	1	**0**
1	0	1	0	0	0	**0**
1	0	1	0	0	1	**1**
1	0	1	0	1	0	**1**
1	0	1	0	1	1	**0**
1	0	1	1	0	0	**0**
1	0	1	1	0	1	**0**
1	0	1	1	1	0	**0**
1	0	1	1	1	1	**0**
1	1	0	0	0	0	**0**
1	1	0	0	0	1	**0**
1	1	0	0	1	0	**0**
1	1	0	0	1	1	**0**
1	1	0	1	0	0	**0**
1	1	0	1	0	1	**0**
1	1	0	1	1	0	**0**
1	1	0	1	1	1	**0**

A	B	C	D	E	F	O
1	1	1	0	0	0	**0**
1	1	1	0	0	1	**0**
1	1	1	0	1	0	**0**
1	1	1	0	1	1	**0**
1	1	1	1	0	0	**0**
1	1	1	1	0	1	**0**
1	1	1	1	1	0	**0**
1	1	1	1	1	1	**0**

14) Código de esta primera aproximación

```
Uses crt;


var i, j, c: integer;
  t: array[1..10,1..10]of char;
  te, ts: text;
  q: char;
  b: boolean;


Procedure check;
 Begin
  for i:=1 to 10 do
   Begin
    for j:=1 to 10 do
     Begin
      if (t[i,j]<>'.')and(t[i,j]<>'?') then
       Begin
        if ((i>1)and(j>1))and((t[i-1,j-1]='x')) THEN inc (c);
        if (j>1)and((t[i,j-1]='x')) THEN inc (c);
        if ((i<10)and(j>1))and((t[i+1,j-1]='x')) THEN inc (c);
        if (i<10)and((t[i+1,j]='x')) THEN inc (c);
        if ((i<10)and(j<10))and((t[i+1,j+1]='x')) THEN inc (c);
        if ((j<10))and((t[i,j+1]='x')) THEN inc (c);
        if ((i>1)and(j<10))and((t[i-1,j+1]='x')) THEN inc (c);
        if ((i>1))and((t[i-1,j]='x')) THEN inc (c);
        q:=char(c+ord('0'));
        if t[i,j]<>q then b:=false;
       End;
     End;
   End;
 End;


Begin
```

```
assign(te,'buscaminas.in'); reset(te);
assign(ts,'buscaminas.out'); rewrite(ts);
b:=true;
for i:=1 to 10 do
 Begin
  for j:=1 to 10 do read(te,t[i,j]);
  readln(te);
 End;
check;
If b=true then writeln(ts,'Este sistema es compatible')
              else  writeln(ts,'Este sistema no es compatible');

close(te); close(ts);
End.
```

15) Código del programa "seguro"

```
uses crt;
var i,j,c,cc:integer;
    t:array[1..10,1..10]of char;
    te,ts:text;
    q:char;
    b:boolean;
    tt,ttt: array[1..10] of integer;


Procedure seguro;
 Begin
  for i:=1 to 10 do
   Begin
    for j:=1 to 10 do
     Begin
      if (t[i,j]<>'.')and(t[i,j]<>'?')and(t[i,j]<>'x')and(t[i,j]<>'0') then
       Begin
        c:=0;
        if ((i>1)and(j>1))and((t[i-1,j-1]='.')or(t[i-1,j-1]='x')) THEN inc (c);
        if (j>1)and((t[i,j-1]='x')or(t[i,j-1]='.')) THEN inc (c);
        if ((i<10)and(j>1))and((t[i+1,j-1]='x')or(t[i+1,j-1]='.')) THEN inc (c);
        if (i<10)and((t[i+1,j]='x')or(t[i+1,j]='.')) THEN inc (c);
        if ((i<10)and(j<10))and((t[i+1,j+1]='x')or(t[i+1,j+1]='.')) THEN inc (c);
        if ((j<10))and((t[i,j+1]='x')or(t[i,j+1]='.')) THEN inc (c);
        if ((i>1)and(j<10))and((t[i-1,j+1]='x')or(t[i-1,j+1]='.')) THEN inc (c);
        if ((i>1))and((t[i-1,j]='x')or(t[i-1,j]='.')) THEN inc (c);
        q:=char(c+ord('0'));
        if q=t[i,j] then
         Begin
          if ((i>1)and(j>1))and((t[i-1,j-1]='.')or(t[i-1,j-1]='x')) THEN t[i-1,j-1]:='x';
          if (j>1)and((t[i,j-1]='x')or(t[i,j-1]='.')) THEN t[i,j-1]:='x';
          if ((i<10)and(j>1))and((t[i+1,j-1]='x')or(t[i+1,j-1]='.')) THEN t[i+1,j-1]:='x';
```

```
        if (i<10)and((t[i+1,j]='x')or(t[i+1,j]='.')) THEN t[i+1,j]:='x';
        if ((i<10)and(j<10))and((t[i+1,j+1]='x')or(t[i+1,j+1]='.'))
         THEN t[i+1,j+1]:='x';
        if ((j<10))and((t[i,j+1]='x')or(t[i,j+1]='.')) THEN t[i,j+1]:='x';
        if ((i>1)and(j<10))and((t[i-1,j+1]='x')or(t[i-1,j+1]='.')) THEN t[i-1,j+1]:='x';
        if ((i>1))and((t[i-1,j]='x')or(t[i-1,j]='.')) THEN t[i-1,j]:='x';
       End;
     End;
    End;
  End;
 End;



Begin

assign(te,'buscaminas.in'); reset(te);
assign(ts,'buscaminas.out'); rewrite(ts);
b:=true;
for i:=1 to 10 do
 Begin
  for j:=1 to 10 do read(te,t[i,j]);
  readln(te);
 End;
seguro;

for i:=1 to 10 do
 Begin
  for j:=1 to 10 do write(ts,t[i,j]);
  writeln(ts);
 End;
close(te); close(ts);
End.
```

16) Código de la resolución final del buscaminas

```
uses crt;

var i,d,j,c,cc,k,kk,k1,h,f,ff:integer;
   t,solu:array[1..10,1..10]of char;          {Definimos todas las variables}
   te,ts:text;
   q:char;
   b,binar,solucion:boolean;
   tt,tx,ty: array[1..10] of integer;
   bi:array[1..11]of integer;

Procedure segur; {Todo este procedure es la primera parte de nuestros objetivos}
 Begin                    {sirve para reducer el número de variables}
  for i:=1 to 10 do
   Begin
    for j:=1 to 10 do
     Begin
      if (t[i,j]<>'.')and(t[i,j]<>'?')and(t[i,j]<>'x')and(t[i,j]<>'0') then
       Begin
        c:=0;
        if ((i>1)and(j>1))and((t[i-1,j-1]='.')or(t[i-1,j-1]='x')) THEN inc (c);
        if (j>1)and((t[i,j-1]='x')or(t[i,j-1]='.')) THEN inc (c);
        if ((i<10)and(j>1))and((t[i+1,j-1]='x')or(t[i+1,j-1]='.')) THEN inc (c);
        if (i<10)and((t[i+1,j]='x')or(t[i+1,j]='.')) THEN inc (c);
        if ((i<10)and(j<10))and((t[i+1,j+1]='x')or(t[i+1,j+1]='.')) THEN inc (c);
        if ((j<10))and((t[i,j+1]='x')or(t[i,j+1]='.')) THEN inc (c);
        if ((i>1)and(j<10))and((t[i-1,j+1]='x')or(t[i-1,j+1]='.')) THEN inc (c);
        if ((i>1))and((t[i-1,j]='x')or(t[i-1,j]='.')) THEN inc (c);
        q:=char(c+ord('0'));
        if q=t[i,j] then
         Begin
          if ((i>1)and(j>1))and((t[i-1,j-1]='.')or(t[i-1,j-1]='x')) THEN t[i-1,j-1]:='x';
          if (j>1)and((t[i,j-1]='x')or(t[i,j-1]='.')) THEN t[i,j-1]:='x';
```

```
          if ((i<10)and(j>1))and((t[i+1,j-1]='x')or(t[i+1,j-1]='.')) THEN t[i+1,j-1]:='x';
          if (i<10)and((t[i+1,j]='x')or(t[i+1,j]='.')) THEN t[i+1,j]:='x';
          if ((i<10)and(j<10))and((t[i+1,j+1]='x')or(t[i+1,j+1]='.')) THEN t[i+1,j+1]:='x';
          if ((j<10))and((t[i,j+1]='x')or(t[i,j+1]='.')) THEN t[i,j+1]:='x';
          if ((i>1)and(j<10))and((t[i-1,j+1]='x')or(t[i-1,j+1]='.')) THEN t[i-1,j+1]:='x';
          if ((i>1))and((t[i-1,j]='x')or(t[i-1,j]='.')) THEN t[i-1,j]:='x';
         End;
       End;
     End;
   End;
for i:=1 to 10 do
  Begin
   for j:=1 to 10 do solu[i,j]:=t[i,j];
  End;
 End;
```

Procedure identificar; {Nos pone en dos tablas, una de cada eje, tx y ty, las coordenadas de los '.' que posteriormente habremos de mirar si va una mina o no}

```
 Begin
  for i:=1 to 10 do
   Begin
    for j:=1 to 10 do
     Begin
      If t[i,j]='.' then Begin inc(cc); tx[cc]:=i; ty[cc]:=j; End;
     End;
   End;
 if cc=0 then solucion:=true;
 End;
```

Procedure consistent; {Nos dice si el sistema es consistente o no.}

```
 Begin
  binar:=true;
```

```
for i:=1 to 10 do
 Begin
  for j:=1 to 10 do
   Begin
    if (t[i,j]<>'.')and(t[i,j]<>'x') then
     Begin
      f:=0; ff:=0;
      if ((i>1)and(j>1))and((t[i-1,j-1]='x')) THEN inc (f);
      if (j>1)and((t[i,j-1]='x')) THEN inc (f);
      if ((i<10)and(j>1))and((t[i+1,j-1]='x')) THEN inc (f);
      if (i<10)and((t[i+1,j]='x')) THEN inc (f);
      if ((i<10)and(j<10))and((t[i+1,j+1]='x')) THEN inc (f);
      if ((j<10))and((t[i,j+1]='x')) THEN inc (f);
      if ((i>1)and(j<10))and((t[i-1,j+1]='x')) THEN inc (f);
      if ((i>1))and((t[i-1,j]='x')) THEN inc (f);

      if ((i>1)and(j>1))and((t[i-1,j-1]='.')) THEN inc (ff);
      if (j>1)and((t[i,j-1]='.')) THEN inc (ff);
      if ((i<10)and(j>1))and((t[i+1,j-1]='.')) THEN inc (ff);
      if (i<10)and((t[i+1,j]='.')) THEN inc (ff);
      if ((i<10)and(j<10))and((t[i+1,j+1]='.')) THEN inc (ff);
      if ((j<10))and((t[i,j+1]='.')) THEN inc (ff);
      if ((i>1)and(j<10))and((t[i-1,j+1]='.')) THEN inc (ff);
      if ((i>1))and((t[i-1,j]='.')) THEN inc (ff);

      q:=char(f+ord('0'));
      If (t[i,j]='?')and(ff<>0) then binar:=true;
      if (t[i,j]<>'?')and(t[i,j]<>q) then binar:=false;
      if binar=false then break;
     End;
    if binar=false then break;
   End;
 if binar=false then break;
```

```
  End;
 End;


Procedure binari;
 Begin
  for c:=1 to cc do
   Begin
    h:=c;
    if h>=1024 then Begin h:=h-1024; bi[1]:=1; End;
    if h>=512 then Begin h:=h-512; bi[2]:=1; End;
    if h>=256 then Begin h:=h-256; bi[3]:=1; End;
    if h>=128 then Begin h:=h-128; bi[4]:=1; End;       {Nos pasa el número de la
    if h>=64 then Begin h:=h-64; bi[5]:=1;   End;        variable a sistema binario}
    if h>=32 then Begin h:=h-32; bi[6]:=1;   End;
    if h>=16 then Begin h:=h-16; bi[7]:=1;   End;
    if h>=8 then Begin h:=h-8; bi[8]:=1;     End;
    if h>=4 then Begin h:=h-4; bi[9]:=1;     End;
    if h>=2 then Begin h:=h-2; bi[10]:=1;    End;
    if h>=1 then Begin h:=h-1; bi[11]:=1;    End;
    for k:=1 to 11 do
     Begin
      if bi[k]=1 then Begin t[tx[c],ty[c]]:='x';  consistent; End;
     End;
    for d:=1 to 11 do bi[d]:=0;
    If binar=true then                          {Mira si el sistema es compatible o no}
     Begin
      for i:=1 to 10 do
       Begin
        for j:=1 to 10 do solu[i,j]:=t[i,j];
       End;
     End
    else t[tx[c],ty[c]]:='.';
```

```
 End;
End;
```

Procedure comprobar; {Mira si la solución definitive sea correcta y no se hayan producido errores}

```
Begin
 for i:=1 to 10 do
  Begin
   for j:=1 to 10 do
    Begin
     if (t[i,j]<>'x') then
      Begin
       if t[i,j]='.' then
        Begin
         ff:=0;
         if ((i>1)and(j>1))and((t[i-1,j-1]='x')) THEN inc (ff);
         if (j>1)and((t[i,j-1]='x')) THEN inc (ff);
         if ((i<10)and(j>1))and((t[i+1,j-1]='x')) THEN inc (ff);
         if (i<10)and((t[i+1,j]='x')) THEN inc (ff);
         if ((i<10)and(j<10))and((t[i+1,j+1]='x')) THEN inc (ff);
         if ((j<10))and((t[i,j+1]='x')) THEN inc (ff);
         if ((i>1)and(j<10))and((t[i-1,j+1]='x')) THEN inc (ff);
         if ((i>1))and((t[i-1,j]='x')) THEN inc (ff);
         if ff<>0 then solu[i,j]:='?';
        End;
       if (t[i,j]<>'.')and(t[i,j]<>'?')and(t[i,j]<>'x') then
        Begin
         ff:=0;
         if ((i>1)and(j>1))and((t[i-1,j-1]='.')) THEN inc (ff);
         if (j>1)and((t[i,j-1]='.')) THEN inc (ff);
         if ((i<10)and(j>1))and((t[i+1,j-1]='.')) THEN inc (ff);
         if (i<10)and((t[i+1,j]='.')) THEN inc (ff);
         if ((i<10)and(j<10))and((t[i+1,j+1]='.')) THEN inc (ff);
```

```
        if ((j<10))and((t[i,j+1]='.')) THEN inc (ff);
        if ((i>1)and(j<10))and((t[i-1,j+1]='.')) THEN inc (ff);
        if ((i>1))and((t[i-1,j]='.')) THEN inc (ff);
        q:=char(ff+ord('0'));
        if q=t[i,j] then solucion:=true;
       End;
      End;
     End;
    End;
   End;


Begin

assign(te,'buscaminas.in'); reset(te);
assign(ts,'buscaminas.out'); rewrite(ts);
b:=true;
for i:=1 to 10 do
 Begin
  for j:=1 to 10 do read(te,t[i,j]);
  readln(te);
 End;

segur;

identificar;                                    {Parte principal del trabajo}

if cc<>0 then Begin binari; comprobar; End;

binar:=true;
If solucion=true then
 Begin
  for i:=1 to 10 do
```

```
  Begin
   for j:=1 to 10 do
    Begin
     if (solu[i,j]='.')or(solu[i,j]='?')or(solu[i,j]='x')or(t[i,j]='0')
     or (solu[i,j]='1')or(solu[i,j]='2')or(solu[i,j]='3')or(t[i,j]='4')
     or (solu[i,j]='5')or(solu[i,j]='6')or(solu[i,j]='7')or(t[i,j]='8')then write(ts,solu[i,j])
     else Begin binar:=false; write(ts,'Este sistema no es decidible dado que tiene
distintas soluciones'); break; End;
    End;
    if binar=false then break;
   writeln(ts);
  End;
 End
else write(ts,'Este sistema no es compatible');



close(te); close(ts);
End.
```