REPRESENTACIÓN DEL CONOCIMIENTO

TRABAJO PRÁCTICO

Imagen que contiene dibujo, alimentos

Descripción generada automáticamente

Grado en Ingeniería Informática

Francisco Jesús Beltrán Moreno

Índice

[Introducción 2](#_Toc41393741)

[Conjetura de Goldbach 2](#_Toc41393742)

[Predicados complementarios 2](#_Toc41393743)

[Fichero: p31.pl 2](#_Toc41393744)

[is\_prime(+P) 2](#_Toc41393745)

[has\_factor(+N, +L) 3](#_Toc41393746)

[Fichero: p40.pl 3](#_Toc41393747)

[goldbach(+N, -L) 3](#_Toc41393748)

[goldbach(+N, [-P,-Q], +P) 3](#_Toc41393749)

[goldbach(+N, +L, +P) 4](#_Toc41393750)

[next\_prime(+P, -P1) 4](#_Toc41393751)

[Fichero: p41.pl 4](#_Toc41393752)

[goldbach\_list(+A, +B) 4](#_Toc41393753)

[goldbach\_list(+A, +B, +L) 4](#_Toc41393754)

[g\_list(+A, +B, +L) 5](#_Toc41393755)

[print\_goldbach(+A, +P, +Q, +L) 5](#_Toc41393756)

[Pruebas de ejecución 6](#_Toc41393757)

[Bibliografía 7](#_Toc41393758)

# Introducción

El presente documento desarrolla la implementación de la Conjetura de Goldbach en Prolog. Como solución al problema se ha tomado el código de los problemas [31](https://www.ic.unicamp.br/~meidanis/courses/mc336/2009s2/prolog/problemas/p31.pl), [40](https://www.ic.unicamp.br/~meidanis/courses/mc336/2009s2/prolog/problemas/p40.pl) y [41](https://www.ic.unicamp.br/~meidanis/courses/mc336/2009s2/prolog/problemas/p41.pl) (is\_prime, Goldbach's conjecture, A list of Goldbach compositions) de [P-99:Ninety-Nine Prolog Problems](https://www.ic.unicamp.br/~meidanis/courses/mc336/2009s2/prolog/problemas/) de **Werner Hett**.

**-Incidir en el no uso del corte-**

**Nosotros no hacemos uso del corte, porque no pensamos en ejecución, vemos los algoritmos como algo estático, y es completamente opuesto a nuestra filosofía de programación impartida en la asignatura.**

Por los nombres de los problemas podemos ver que tenemos varias posibilidades, que más adelante abordaremos y explicaremos.

# Conjetura de Goldbach

En teoría de números, la Conjetura de Goldbach es uno de los problemas abiertos más antiguos en matemáticas.

Concretamente, Godfrey Harold Hardy, en 1921, en su famoso discurso pronunciado en la Sociedad Matemática de Copenhague, comentó que probablemente la conjetura de Goldbach no es solo uno de los problemas no resueltos más difíciles de la teoría de números, sino de todas las matemáticas.

Su enunciado es el siguiente:

Todo número par mayor que 2 puede escribirse como la suma de dos números primos.

[***Christian Goldbach (1742)***](https://es.wikipedia.org/wiki/Christian_Goldbach)

# Predicados complementarios

En este apartado se explicarán los predicados o llamadas a otros códigos de las que hace uso nuestro ejercicio para producir la solución.

## Fichero: p31.pl

### is\_prime(+P)

*Es cierto si* ***P*** *es un* ***número primo****.*

Para el desarrollo de este predicado se ha usado la inducción matemática partiendo del caso base en el que el **número 2 y 3 son primos**.

Para otros casos, se comprueba si el resto de dividir **P** entre 2 es distinto de 0 y no es divisible entre 3.

Imagen que contiene objeto, pantalla, monitor, reloj

Descripción generada automáticamente

### has\_factor(+N, +L)

*Es cierto si* ***N*** *es divisible entre* ***L****.*

El primer caso comprueba si el resto de la división entre **N** y **L** es **0**.

El segundo caso comprueba que el producto **L \* L** sea menor que **N**, y se vuelve a llamar con **L + 2** como nueva **L**.

Imagen que contiene reloj, objeto, leer, oscuro

Descripción generada automáticamente

## Fichero: p40.pl

Hace uso del fichero p31, lo carga con *:- ensure\_loaded(p31)*.

### goldbach(+N, -L)

*Es cierto si* ***L*** *unifica con una lista que contiene los posibles números primos que sumados resultan en* ***N, y N tiene que ser par****.*

Este predicado tiene como caso base, el caso en el que N=4, que es la suma del número 2, y cumpliría todas las reglas, por lo que hace uso del operador de corte “!” y termina.

Para cualquier otro caso, se comprueba si el resto de dividir N entre 2 es 0, que N es mayor que 4 y se llama a la siguiente función goldbach(N,L,3)**.**

Imagen que contiene reloj, objeto, oscuro, negro

Descripción generada automáticamente

### goldbach(+N, [-P,-Q], +P)

*Es cierto si* ***Q*** *unifica con un valor primo resultante de la resta de* ***N – P****.*

Imagen que contiene objeto, reloj, naranja, negro

Descripción generada automáticamente

### goldbach(+N, +L, +P)

*Es cierto si* ***P es menor que N****, existe un número* ***primo******mayor que P****, y esté* ***unifica*** *con la regla goldbach(N,[P,Q], P).*

Imagen que contiene objeto, reloj, naranja, medidor

Descripción generada automáticamente

### next\_prime(+P, -P1)

*Es cierto si existe un* ***número primo mayor que P****.*

Tenemos dos casos con esta función, el primero es si el siguiente número primo mayor a P es P + 2 se termina la ejecución.

El segundo es seguir buscando ese posible número primo.

Imagen que contiene reloj, objeto, negro, montado

Descripción generada automáticamente

## Fichero: p41.pl

Hace uso del fichero p40, lo carga con *:- ensure\_loaded(p40)*.

### goldbach\_list(+A, +B)

*Es cierto si muestra una lista con todos los* ***posibles números pares*** *en el* ***rango******A <= N <= B****.*

Esta función llama a goldbach\_list(A, B, 2), empieza con incremento 2.

### goldbach\_list(+A, +B, +L)

*Es cierto si existe una salida entre los* ***valores A y B con incremento L****.*

Tenemos 2 casos, si **A** **=< 4**, se termina la ejecución y vuelve a lanzarse pero llamando a **g\_list(4, B, L)**.

Por otro lado, incrementamos el valor de **A** en 1 y haciendo la división real entre 2, para llamar con ese nuevo valor **A1** a **g\_list(A1, B, L)**. Básicamente, vamos reduciendo el rango de valores.

### g\_list(+A, +B, +L)

*Es cierto si existe una salida cuyos valores estén entre el rango* ***A <= N <= B con incremento L****.*

Esta función se encarga de llamar a **goldbach(A, [P, Q])**. (p40), llama a la función **print\_goldbach(A, P, Q, L)**, incrementa A y vuelve a llamarse a sí misma.

### print\_goldbach(+A, +P, +Q, +L)

*Es cierto si* ***P >= L****, en cuyo caso termina y muestra por pantalla el resultado.*

Para el caso que venga cualquier cosa, también es verdadero.

Imagen que contiene pantalla, celular, tabla, teléfono

Descripción generada automáticamente

# Pruebas de ejecución

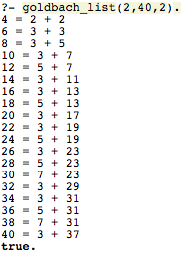
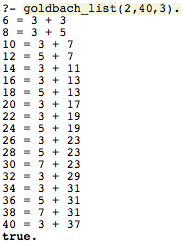
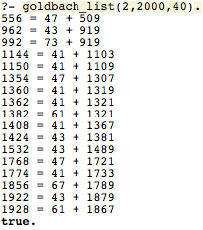


Imagen que contiene computadora

Descripción generada automáticamente



# Bibliografía

<https://es.wikipedia.org/wiki/Conjetura_de_Goldbach>

<https://es.wikipedia.org/wiki/Teor%C3%ADa_de_n%C3%BAmeros>

<https://es.wikipedia.org/wiki/Conjetura_de_Goldbach>

<https://www.ic.unicamp.br/~meidanis/courses/mc336/2009s2/prolog/problemas/>

<https://swish.swi-prolog.org/p/Tutorial%20de%20prolog.swinb>

<https://www.swi-prolog.org/pldoc/doc_for?object=root>

<http://www.gedlc.ulpgc.es/docencia/lp/documentacion/GB_Prolog.pdf>

Búsqueda de términos:

<https://www.wordreference.com/>