



UNIVERSIDAD DE GRANADA
DEPT. DE ALGEBRA

GRADO EN INGENIERÍA INFORMÁTICA
INGENIERÍA DEL SOFTWARE (4º CURSO)

Lógica y Programación: Ejercicios Teóricos y Prácticos

Autor:

Francisco Miguel GARCÍA OLMEDO

15 de septiembre de 2025

Índice

0. Normas	3
1. Problemas de λ-Calculus y Lógica Combinatoria	4
2. Ejercicios de Programación en Haskell y Prolog	5

0. Normas

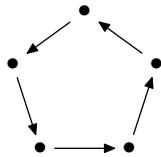
El alumnado deberá observar lo siguiente:

1. El alumno resolverá los ejercicios de la **Sección 1** del **1** al **9**. Los entregará manuscritos en un documento .pdf como parte del material evaluable. La resolución de estos ejercicios, junto a su defensa ante el profesor supondrá el 20 % de la nota del curso.
2. Como parte del material evaluable, el alumno: elegirá, desarrollará y entregará en formato .pdf un apartado de libre elección perteneciente a los temas 1 y 2 de los relacionados en el apartado “Temario de la Asignatura” incluido en la guía docente de la asignatura, página 4. Este material, junto a su defensa ante el profesor, supondrá el 30 % de la nota del curso.
3. De los ejercicios de la **Sección 2** el alumno resolverá y entregará los ficheros del proyecto de implementación como parte del material evaluable. Cada cual o cada grupo elegirá al menos dos de los ejercicios: **1**, **2**, y **4** siendo **3** de obligada elección; entregará los ficheros de cada proyecto en carpetas separadas. La evaluación hecha sobre ese material junto a su defensa será la base para la obtención del 50 % de la nota; a ella contribuirá las notas relacionadas que el profesor haya recogido durante el desarrollo del curso.
4. Los ejercicios aludidos en este documento, que constituyen la base para la evaluación y calificación, pueden ser hechos en grupo. Los grupos serán comunicados al profesor con antelación suficiente y serán por lo general de dos alumnos. Caso de existir el deseo o la necesidad de que el grupo sea más grande, es posible previa autorización del profesor. Cada integrante del grupo deberá hacer entrega de exactamente el mismo material que el resto de sus compañeros en el grupo para poder obtener nota y calificación.-
5. La evaluación tendrá especialmente en cuenta la eficiencia y la creatividad desarrollando —o incluso añadiendo a— lo pedido.-

6. La entrega de los contenidos será a través de la plataforma mediante las tareas que el profesor emitirá, pero para poder ser evaluados deben ser luego defendidos ante el profesor.-

1. Problemas de λ -Calculus y Lógica Combinatoria

1. Exponga y desarrolle justificadamente el tema de la “Notación de de Bruijn”.
2. Demuestre que para todo λ -término N , $\lambda x.x\mathbf{K}N \# \lambda x.x\mathbf{S}N$. (Nota: Recuérdese que $\mathbf{S} \equiv \lambda xyz.xz(yz)$ y que $\mathbf{K} \equiv \lambda xy.x$).
3. Dibuje razonadamente el grafo $G_\beta(WWW)$, donde $W \equiv \lambda xy.xyy$.
4. Encuentre razonadamente un λ -término M tal que $G_\beta(M)$ sea exactamente:



5. Un λ -término cerrado M es un operador (combinador) de punto fijo si, por definición, para todo λ -término F , $F(MF) = MF$. Sea el λ -término:

$$G \equiv \lambda yx.x(yx)$$

y

$$M \equiv (\lambda xy.y(xxy))(\lambda xy.y(xxy))$$

- a) Demuestre que M es un punto fijo de G ,
- b) Demuestre que si el combinador N es un punto fijo de G , entonces N es un operador de punto fijo.
- c) Demuestre que M es un combinador de punto fijo.
- d) Demuestre que si M es un combinador de punto fijo, entonces $M = GM$.

6. Considere el combinador:

$$Y \equiv \lambda y.(\lambda x.y(xx))(\lambda x.y(xx))$$

y demuestre que $GY = Y$.

7. Considere la sucesión de combinadores $\{Y^n\}_n$ definida para todo número natural n como sigue:

$$Y^n = \begin{cases} Y & , \text{ si } n = 0 \\ Y^{n-1}G & , \text{ si } n > 0 \end{cases}$$

Demuestre que para todo $n \geq 0$, Y^n es un combinador de punto fijo.

8. Encuentre razonadamente el CL-término $(\lambda xy.xyy)_{CL}$.
9. Esquematice la relación entre el sistema λ y la lógica combinatoria.

2. Ejercicios de Programación en Haskell y Prolog

1. Implemente en Haskell el tipo de dato conjunto, dotándolo del mayor número de rasgos y operaciones conocidas para este objeto matemático.
2. Implemente el cálculo del área de un triángulo según la [Fórmula de Herón](#). El proyecto deberá tener al menos las siguientes funcionalidades:
 - a) Una función que discierna si una terna de números dada es o no la correspondiente a los lados de un triángulo.
 - b) El cálculo de la raíz cuadrada mediante el método de [Newton–Raphson](#).
3. Desarrolle el siguiente proyecto:
 - a) Implemente en Haskell el criptosistema de Vigenère.
 - b) Implemente en Haskell un “laboratorio” con lo suculento para poder poner en claro cualquier criptograma cifrado mediante un criptosistema de Vigenère. Para ello básease en lo contenido en el libro [An Introduction to Mathematical Cryptography](#) de Hoffstein, Pipher y Silverman y en el artículo de Dalkilic y Gungor descargable desde [An Interactive Cryptanalysis Algorithm for the Vigenere Cipher](#). Ambos volúmenes puede ser descargados legalmente desde el dominio de la UGR con los enlaces provistos.
 - c) Lo siguiente:

```
UECWKDVLOTTVACKTPVGEZQMDAMRNPDUXLBUICAMRHOECBHSPQLVIWO
FFEAILPNTESMLDRUURIFAEQTPXADWIawlACCRPBHSRZIVQWOFROGTT
NNXEVIVIBPDTTGAHVIACLAYKGJIEQHGECMESNNOCTHSGGNVWTQHKBPR
HMVUOYWLIAFIRIGDBOEBQLIGWARQHNLOISQKEPEIDVXXNETPAXNZGDX
WWEYQCTIGONNGJVHSQGEATHSYGSDVVOAQCXLHSPQMDEMTRTMDUXTEQQ
JMFAEEAAIMEZREGIMUECICBXRVQRSMENNWTXTNSRNBPZHMVRDYNCEG
SPMEAVENTXKEQKCTTHSPCMQQHSQGTXMFPBGLWQZRBOEIZHQHGRTOBSG
TATTZRNFOSMLEDWESIWDRNAPBF OFHEGIXLFVOGUZLNUSRCRAZGZRTTA
YFEHKHMCQNTZLENPUCKBAYCICUBNRPCXIWEYCSIMFPRUTPLXSYCBGCC
UYCQJMWEKGTUBRHVATTLEKVACBXQHGPDZEANNTJZTDRNSDTFEVPDXK
TMVNAIQMUQNOHKKOAQMTBKOFsutuxprtmbxnpclrceaoiaWGGVVUS
GIOEWLIQFOZKSPVMEMBLQHLDVCYSMGOPJEFCXMRUIGDXNCCRPMLECEWT
PZMOQQSAWLPHPTDAWEYJOGQSOAVERCTNQQEAVTUGKLJAXMRTGTIEAFW
PTZYIPKESMEAFCGJILSBPLDABNVRJUXNGQSWIUIGWAAMLDRNNPDXGN
```

PTTGLUHUOBMXSPQNDKBDTEECLECGRDPTYBVRDATQHKQJMKEFROCLXN
FKNSCWANNAHXTRGKCJTTRUEMQZAEIPAWEYPAJBBLHUEHMVUNFRPVM
EDWEKMHRREOGZBDBROGCGANIUYIBNZQVXTGORUUCUTNBOEIZHEFWNB
GOZGTGXNRHERBPHGSIWXNPQMJVBCNEIDVVOAGLPONAPWYPXKEFKOC
MQTRTIDZBNQKCPLTTNOBXMGLNRRDNNNQKDPLTLNSUTAXMNPTXMGEZKA
EIKAGQ

es el resultado de cifrar determinado texto mediante un criptosistema de Vigenère. Basándose en lo implementado en el apartado anterior, descubra: la longitud de la clave utilizada, la clave misma y el contenido del mensaje que resultó cifrado en el texto anterior.

4. Con comentarios explicativos suficientes implemente en Prolog el puzzle de las Torres de Hanoi.