Ministerio de Cultura y Educación **(Programa del año 2023)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area V: Automatas y Lenguajes

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
COMPUTABILIDAD Y COMPLEJIDAD	LIC.CS.COMP.	32/12	2023	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
LEGUIZAMON, MARIO GUILLERMO	Prof. Responsable	P.Asoc Exc	40 Hs
APOLLONI, JAVIER MARIANO	Prof. Co-Responsable	P.Adj Exc	40 Hs
VILLEGAS, MARIA PAULA	Auxiliar de Práctico	A.1ra Semi	20 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
Hs	2 Hs	4 Hs	Hs	6 Hs

Tipificación	Periodo
C - Teoria con prácticas de aula	2º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
07/08/2023	18/11/2023	15	90

IV - Fundamentación

El perfil del licenciado incluye el conocimiento científico necesario que le permitir analizar y entender problemas, para los que luego, pueda desarrollar y obtener una solución adecuada, utilizando los beneficios de la tecnología y, en especial, de una computadora. En este contexto, una computadora puede ser vista como un dispositivo de cómputo que realiza tanto operaciones como cálculos de manera automática y que son plasmados en un algoritmo. Al intentar dar una formalización a la idea intuitiva de algoritmo se observa rápidamente que es necesario estudiar los modelos formales de cómputo y con ellos, los lenguajes formales.

Es así que, la actividad curricular “Autómatas y Lenguajes” da un primer paso en este sentido, analizando los lenguajes formales y conjuntamente con ellos, los modelos formales de cómputo basados en los dispositivos de reconocimiento o aceptación (autómatas o máquinas de estados) y los modelos basados en dispositivos generadores (gramáticas).

La presente actividad curricular, “Computabilidad y Complejidad” continúa el estudio de los lenguajes formales haciendo, en una primera etapa, hincapié y profundizando en los modelos avanzados de cómputo, sus vínculos con los problemas, los algoritmos y las soluciones. En una siguiente etapa, utiliza estos modelos avanzados de cómputos como herramientas para analizar la Teoría de la Computabilidad. Esta teoría estudia las propiedades sobre los lenguajes asociados a los problemas y permite clasificarlos en decidibles y no decidibles. Con ello se establece un límite entre los problemas efectivamente resoluble (o computables) de aquellos que no pueden ser resueltos (o no computables) mediante un algoritmo.

Por último, y en una tercera etapa analiza la Teoría de la Complejidad utilizando también los modelos formales y avanzados

de cómputo. En este caso, se estudian las propiedades de los lenguajes decidibles respecto al uso de los recursos, ya sea el tiempo, el espacio o algún otro. La finalidad del estudio es jerarquizar la complejidad de los lenguajes decidibles (y por ende de sus problemas asociados) para dividirlos en lenguajes (o problemas) con soluciones que usen eficientemente el recurso analizado de aquellos que no poseen hasta el momento una solución conocida que utilice eficientemente el recurso analizado.

Esta actividad curricular aborda todos los elementos teóricos y prácticos necesarios para que la/el estudiante pueda incorporar y comprender los conceptos mencionados en los párrafos anteriores. El abordaje se hará desde una perspectiva que aproxime significativamente a la/al estudiante al aprendizaje incremental y constructivista. Propiciará la integración de los contenidos de la actividad curricular con los conocimientos adquiridos en actividades curriculares previas de la carrera, principalmente en Matemática Discreta, Programación I y II, Estructura de Datos y Algoritmos, entre otras, y considerará además el contexto social de la/del estudiante.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que la/el estudiante alcance los siguientes objetivos generales y específicos:

Objetivos generales:
- Vincular los aspectos teóricos y prácticos de la Teoría de Lenguajes Formales, de la Teoría de la Computabilidad y de la Teoría de la Complejidad con la realidad y los problemas de la realidad.
- Fomentar el pensamiento analítico y de interpretación para resolver problemas (de la realidad) dados como enunciados en los ejercicios prácticos utilizando el marco teórico de la materia.
- Mejorar las capacidades y habilidades de la/del estudiante para su futuro laboral.

Objetivos específicos:

-Profundizar el conocimiento de la Teoría de los Lenguajes Formales
-Diseñar lenguajes (formales) para representar problemas de la realidad.
-Comprender conceptos y funcionamientos relacionados con los modelos avanzados de cómputo: Máquina de Turing, Funciones Recursivas, Lenguajes estructurados a frases y la correspondencia entre ellos y con los lenguajes de programación.
-Diseñar soluciones que resuelvan un problema de la realidad dado en forma de enunciado
utilizando el modelo avanzado de cómputos más adecuado al problema
- Comprender y analizar los conceptos prácticos y teóricos asociados a la Teoría de la Computabilidad utilizando la Máquina de Turing como el modelo avanzados de cómputo y a partir de ello establecer la decibilidad o no de los lenguajes (lenguajes decidibles y no-decidibles).
- Adquirir la capacidad para discernir cuándo un problema es resoluble y cuándo no lo es analizando propiedades de su lenguaje de decisión asociado
-Comprender y analizar los conceptos vinculados a la Teoría de la Complejidad utilizando modelos avanzados de computación como la Máquina de Turing y con ello establecer la tratabilidad o no de los problema al clasificarlos en P, NP, NP-completos y otros relacionados.
- Adquirir la capacidad para detectar cuándo un problema es tratable y cuándo no lo es analizando propiedades de los lenguajes decidibles.

VI - Contenidos

Contenidos mínimos (según OCD 32/12):
Computabilidad: Introducción a cardinales transfinitos. Formalización del concepto de problema. Problemas de decisión. Máquina de Turing. Variantes de Máquina de Turing. Conjuntos recursivos y recursivamente enumerables. Reducibilidad. Lenguaje Universal. Problema de Detención. Problemas decidibles y no decidibles. Funciones computables. Funciones recursivas primitivas. Operador de minimización. Funciones parcialmente computables. Lenguajes estructurados a frases. Sistemas de Post. Tesis de Church-Turing. Complejidad: Computaciones acotadas en tiempo y espacio en Máquinas de Turing. Clases de Complejidad. Relaciones entre las clases de complejidad. Reducción acotada en tiempo y espacio. Problemas tratables e intratables. Clases P y NP. NP-Completitud. Ejemplos de Problemas NP-Completos.

Estos contenidos mínimos se desarrollan en las siguientes unidades:

Unidad 1. Algoritmo y modelos formales de cómputo.
Introducción. Idea intuitiva de Algoritmo. Modelos formales de cómputo: Máquina de Turing. Máquina de Turing como reconocedor. Máquina de Turing como computadora de funciones. Variantes de la Máquina de Turing: extensiones sobre las Máquinas de Turing (Máquina de Turing No Determinística. Máquina de Turing con k-cintas) y restricciones sobre la Máquina de Turing (Autómata Linealmente Acotado).

Unidad 2. Otros modelos formales de cómputo: Funciones computables.
Funciones recursivas primitivas. Funciones parcialmente computables: Funciones mu-recursivas (operador de minimalización).

Unidad 3. Otros modelos formales de cómputo: Lenguajes estructurados a frases.
Sistema de Post. Sistema de Post como computador de funciones. Equivalencia de los Modelos Formales. Capacidad de los lenguajes de programación. Tesis de Church-Turing.

Unidad 4. Teoría de la Computabilidad: Lenguajes recursivos y recursivamente enumerables
Formalización del concepto de problema. Relación entre problema y lenguaje. Conjuntos/lenguajes recursivos y recursivamente enumerables. Introducción a cardinales transfinitos. Propiedades de los lenguajes recursivos y recursivamente enumerables. Recursividad de los lenguajes reconocidos por un Autómata Linealmente Acotado.

Unidad 5. Teoría de la computabilidad: Decibilidad
Problemas decidibles y no decidibles para lenguajes tipo 0, 1, 2 y 3. Lenguajes no recursivamente enumerables. Lenguaje Universal: Máquina de Turing Universal. Problema de Detención (Halting). Reducibilidad: Concepto de Reducción de un problema a otro. Problema de Correspondencia de Post y la no decibilidad del Problema de Correspondencia de Post. Aplicación del Problema de Correspondencia de Post a problemas de decisión vinculados a gramáticas libres del contexto.

Unidad 6. Teoría de la complejidad: Complejidad temporal y espacial
Introducción al concepto de complejidad computacional. Complejidad temporal y espacial en Máquina de Turing. Medida de complejidad temporal. Notación O-grande y jerarquía de complejidades temporales. Clases de complejidad. Relación entre las complejidades temporales y espaciales. Reducción acotada en el tiempo y espacio.

Unidad 7. Teoría de la Complejidad: Problemas de decisión tratables e intratables
Clases P y NP. NP-completitud. Problema de Satisfactibilidad. Problemas NP-completos. Ejemplos de Problemas NP-Completos: 3-SAT, Clique, Cobertura de vértices, etc.

VII - Plan de Trabajos Prácticos

Durante las semanas que correspondan al dictado se abordarán las 7 unidades del curso tanto desde el contenido teórico como desde las respectivas actividades prácticas. El material de cada unidad estará disponible en el repositorio del curso para que la/el estudiante lo pueda consultar y acceder durante todo el curso.

Las actividades prácticas permitirán el desarrollo y profundización de los contenidos teóricos y será el ámbito donde la/el estudiante alcance los objetivos propuestos para el curso. Las clases prácticas consistirán principalmente en la resolución de ejercicios poniendo en práctica los conocimientos logrados en este curso y aquellos previos utilizando el material didáctico accesible desde el repositorio. En los casos apropiados la/el estudiante podrá hacer uso de las herramientas informáticas que implementan los conceptos teóricos desarrollados en el curso. Se espera que la/el estudiante utilice estas clases prácticas también como ámbito para contrastar ideas y confraternizar con sus compañeros generando y propiciando un espacio de reflexión e intercambio de ideas.

A continuación se desarrolla el plan de trabajos prácticos apuntando en cada caso las habilidades que se esperan que el alumno desarrolle en cada caso.

Plan de Trabajos Prácticos de Aula:

Práctico 1: Máquina de Turing estándar - Extensiones a la MT,
-Revisar la notación, formas y estructura, ejecución y usos de la Máquina de Turing estándar (MTD).
-Pensar, diseñar y construir MTD que reconozcan y acepten las cadenas de lenguajes específicos.
-Pensar, diseñar y construir MTD que computen funciones numéricas y/o realicen procedimientos varios.
-Modelizar problemáticas de la realidad utilizando MTD.
-Entender la notación, formas y estructura, ejecución y usos de la extensión a la MTD: Máquina de Turing con k cintas (MTD-kcintas).
-Pensar, diseñar y construir MTD-kcintas que resuelvan problemas específicos.
-Entender la notación, formas y estructura, ejecución y usos de la extensión a la MTD: Máquina de Turing No Determinística (MTND).
-Pensar, diseñar y construir MTND que resuelvan problemas específicos.
-Aprender a discernir las facilidades descriptivas de los 3 modelos (MTD, MTD-kcintas, MTND) para diseñar y construir el modelo adecuado al problema a resolver.
-Analizar y comparar las similitudes y diferencias del modelo de cómputo más general existente (Máquina de Turing en cualquiera de sus variantes) y de una computadora tanto desde el poder computacional y como el descriptivo (o representacional) para resolver problemas de la realidad.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal, y la corroboración de lo realizado manualmente utilizando la herramienta JFLAP o similar, e implementar una versión simplificada y restringida del modelo de MT en una computadora haciendo uso de un lenguaje de programación.

Práctico 2: Funciones Recursivas,
-Entender la notación, los operadores de composición y recursión, la estructura, la ejecución y el usos de las Funciones Recursivas Primitivas (FRP).
-Pensar, diseñar y construir FRP que computen funciones numéricas (totales).
-Entender el operador de minimalización, su notación, su estructura, su ejecución y usos a fin de extender a la Funciones Generales (FG).
-Pensar, diseñar y construir FG que computen funciones numéricas totales y parciales.
-Extender el modelo de FRP a cualquier dominio y modelizar y resolver problemáticas de la realidad utilizando el modelo.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal, e implementar una versión simplificada y restringida del modelo de FRP en una computadora haciendo uso de un lenguaje de programación.

Práctico 3: Sistemas de Post,
-Entender la notación, formas y estructura, la ejecución y el usos de los Sistemas de Post (SP).
-Pensar, diseñar y construir SP que computen funciones numéricas (totales y parciales), reconozcan las cadenas de lenguajes específicos, realicen procedimientos, etc.
-Pensar y diseñar SP que simulen el comportamiento de otros modelos de cómputos tales como Máquinas de Turing, Autómatas Finitos, Autómatas Pila, Gramáticas,.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal.

Práctico 4: Lenguajes Recursivos y Recursivamente Enumerables
-Entender el vínculo entre problemas y lenguajes formales.
-Analizar, entender y aprender a discernir la diferencia entre lenguajes recursivos y recursivamente enumerables.
-Entender la relación entre lenguajes recursivos y algoritmos.

-Construir algoritmos en pseudo-código que realicen los procedimientos de enumeración para lenguajes asociados a problemas de la realidad.
-Construir algoritmos en pseudo-código que computen la función característica para lenguajes asociados a problemas de la realidad.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal, e implementar en una computadora y haciendo uso de un lenguaje de programación, la función característica para algún lenguaje asociado a un problema de la realidad.

Práctico 5: Problemas decidibles y no decidibles.
-Analizar, entender y aprender a diferenciar los problemas decidibles de los problemas no decidibles.
-Pensar y diseñar codificaciones de instancias de los problemas de la realidad y que sean válidas y adecuadas para el modelo de cómputo general de Máquina de Turing.
-Analizar y diseñar soluciones a lenguajes asociados a problemas decidibles utilizando el modelo general de cómputo de la Máquina de Turing.
-Analizar, entender y resolver y encontrar soluciones (si es posible) a instancias del Problema de Correspondencia de Post (PCP).
-Analizar y entender la no decibilidad del lenguaje asociado al PCP.
-Analizar y diseñar reducciones desde el PCP a problemas relacionados a Lenguajes Libres de Contextos para determinar que los lenguajes asociados a esos problemas son no decidibles.
-Analizar y diseñar soluciones utilizando el modelo general de cómputo de la Máquina de Turing y aplicando el concepto de reducción de un problema a otro para determinar la no decidilidad de ciertos problemas.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal.

Práctico 6: Complejidad Temporal y Espacial.
-Entender el cálculo de la complejidad temporal en una Máquina de Turing de acuerdo al tamaño de la instancia de entrada (longitud de la cadena de entrada).
-Entender la manera en la que se construye la ecuación para la función de complejidad temporal asociada a una Máquina de Turing M para cadenas de una longitud n (tc_M(n)).
-Analizar una Máquina de Turing M (en cualquiera de sus variantes) y construir la ecuación para la función de complejidad temporal asociada a esa Máquina de Turing M para cadenas de una longitud n (tc_M(n)).
-Analizar la afectación que tiene la variante de Máquina de Turing utilizada para resolver el problema en la complejidad temporal asociada a ese problema.
-Analizar y entender la jerarquía de velocidades de crecimiento de la función de complejidad temporal.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal.

Práctico 7: Problemas NP-Completos.
-Analizar y entender las clases de lenguajes o problemas P, NP y NP-Completo.
-Entender el análisis necesario para determinar que un problema es NP-Completo utilizando ejemplos teóricos.
-Analizar y entender el efecto que un problema NP-Completo causa a quién esté resolviendo un problema de la realidad con esas características.
-Vincular los desarrollos realizados para ejemplos teóricos con los necesarios para ejemplos prácticos de la realidad.

Metodología: Resolver los ejercicios utilizando lápiz y papel y compartiendo ideas con los compañeros de aula en un trabajo grupal.

VIII - Regimen de Aprobación

La propuesta de trabajo, en lo que respecta al régimen de aprobación, está basada por un lado en la evaluación sumativa y por otro en la evaluación continua o formativa. Esta último es con el objetivo de relacionar la información sobre la evolución del proceso de aprendizaje de los/las estudiantes con las características de la acción didáctica, a medida que se desarrollan y progresan las actividades de enseñanza y aprendizaje. En este sentido, el estudiante puede regularizar (para luego rendir el examen final) o promocionar. Las condiciones y la metodología propuesta en la que se desarrollará el curso, son las siguientes:

A. Régimen para Estudiantes Regulares

*Aprobar 1 examen parcial práctico o alguna de sus dos respectivas recuperaciones, tal como lo establece la reglamentación vigente.

*Entregar, en tiempo y forma, y aprobar el 100% de las evaluaciones periódicas propuestas por la cátedra las cuales consistirán en ejercicios de los prácticos de aula y en caso de ser necesario, ejercicios extras referidos a las temáticas abordadas en los prácticos.

*Como parte del proceso de aprendizaje y con la finalidad de realizar un trabajo formativo y colaborativo de comprensión, los docentes efectuarán devoluciones evaluativas de las entregas. Todas las evaluaciones pueden tener una instancia de defensa oral posterior a la entrega donde la/el estudiante exponga, explique y responda dudas o inquietudes de sus compañeros y/o docentes acerca del trabajo realizado en sus entregas.

*Los/las estudiantes deben contar además con un porcentaje mínimo igual o superior al 60% de asistencia y participación activa en las clases prácticas y/o actividades prácticas propuestas por los docentes durante el cursado de la materia.

Nota:
1. Las evaluaciones se aprueban con un porcentaje mínimo de setenta por ciento (70%).
2. El examen parcial se aprueba con un porcentaje mínimo de setenta por ciento (70%) de los ejercicios a resolver. Además se debe desarrollar correctamente al menos el 50% de cada uno de los ejercicios involucrados en el parcial para considerar su aprobación.
3. Si cualquier punto no fuera cumplimentado implica que la/el estudiante pase a condición de libre.

B. Régimen para Estudiantes Promocionales
1. Ídem a lo requerido para estudiantes regulares excepto que se debe alcanzar el porcentaje de asistencia solicitado por la reglamentación vigente (80%).
2. La/El estudiante tendrá que rendir y aprobar un coloquio integrador oral o escrito donde muestre su capacidad de vinculación, integración y análisis de los conceptos teóricos dados durante el transcurso del curso.

**La nota final de promoción se computará promediando las notas obtenidas en el examen parcial, evaluaciones periódicas requeridas para la regularidad y la nota obtenida en el coloquio integrador del ítem 2 del punto B. Cada instancia de evaluación (examen parcial, evaluaciones periódicas, coloquio integrador de promoción, etc.) se aprueba con una nota igual a 7 o superior, según lo establece la normativa vigente.

C. Régimen para rendir examen en condición de estudiantes Libres:
1. Realizar un examen escrito de la parte práctica.
2. Habiendo aprobado el examen anterior, pasará a un examen oral o escrito, el cual tendrá las mismas características de un examen final para estudiantes regulares.

D. El examen final puede ser oral y/o escrito.

IX - Bibliografía Básica

[1] J.E. Hopcroft, J. D. Ullman and R. Motwani. "Introduction to automata theory, languages and computation". Tercera Edición. Addison Wesley Publishing Company (2006).

[2] D. Wood. "Theory of computation". John Wiley & Sons, Inc. (1986)
[3] T. A. Sudkamp. "Languages and Machines: An Introduction to the Theory of Computer Science". Tercera Edición. Addison Wesley. (2005)
[4] M. Sipser. "Introduction to the Theory of computation". Tercera Edición. PWS Publishing Company (2012).
[5] P. J. Denning, J. B. Dennis and J. E. Qualitz. "Machine, languages and computation". Prentice-Hall (1978).
[6] Apuntes de la cátedra

X - Bibliografia Complementaria

[1] C. H. Papadimitriou and K. Steiglitz. "Combinatorial Optimization: Algorithms and Complexity". Prentice Hall (1998)
[2] M. D. Davis and E. J. Weyuker. "Computability, Complexity, and Languages: Fundamentals of Computer Science". Academic Press Inc (1993).
[3] H. R. Lewis and C. H. Papadimitriou. "Elements of the theory of computation". Segunda Edición. Prentice Hall (1998).
[4] C. Teuscher. "Alan Turing: Life and Legacy of a Great Thinker". Springer Verlag (2004).
[5] A. Hodges. "Turing". Routledge (1999)
[6] C. H. Papadimitriou. "Computational Complexity". Addison Wesley Publishing Company (1994).
[7] N. D. Jones, Computability and Complexity: From a Programming Perspective. The MIT press. (1997)
[8] J.E. Hopcroft, J. D. Ullman and R. Motwani. "Introduction to automata theory, languages and computation". Segunda Edición. Addison Wesley Publishing Company (2001).
[9] T. A. Sudkamp. "Languages and Machines: An Introduction to the Theory of Computer Science". Segunda Edición. Addison Wesley. (1997)
[10] M. Sipser. "Introduction to the Theory of computation". Primera Edición. PWS Publishing Company (1997).

XI - Resumen de Objetivos

Al finalizar el curso se espera que la/el estudiante sea capaz de: (a) comprender en profundidad la teoría de los lenguajes formales y su jerarquización; (b) entender el funcionamiento de los modelos avanzados y formales de cómputo (máquina de turing, autómata linealmente acotado, funciones recursivas, etc.) y sus relaciones con los lenguajes formales; (c) relacionar los modelos avanzados y formales de cómputo con la idea intuitiva de los algoritmos y con las estructuras y formas de los lenguajes de programación; (d) reconocer y comṕrender la relación entre problemas, problemas decidibles y lenguajes formales, lenguajes decidibles; (e) comprender con cierta profundidad la teoría de la computabilidad y sus implicancias en la posibilidad de resolver o no problemas (lenguajes decidibles y no decidibles); y (f) comprender también con cierta profundidad la teoría de la complejidad computacional y a través de ella categorizar a los problemas de acuerdo a su propiedades (problemas tratables e intratables) y sus dificultades para resolverlos (problemas de la clases P, NP y NP-Completo).

XII - Resumen del Programa

Unidad 1.
Algoritmo y modelos formales de cómputo: Máquina de Turing (extensiones y restricciones).

Unidad 2.
Otros modelos formales de cómputo: Funciones computables y otros.

Unidad 3.
Otros modelos formales de cómputo: Lenguajes estructurados a frases.Tesis de Church.

Unidad 4.
Teoría de la Computabilidad: Problemas y lenguajes. Relación entre problemas y lenguajes. Lenguajes recursivos y recursivamente enumerables.

Unidad 5.
Teoría de la computabilidad: Decibilidad. Problemas decidibles y no decidibles. Reducción. Lenguaje Universal. Problema de Halting. Problema de Correspondencia de Post (y su vínculo con problemas relacionados a gramáticas libres del contexto).

Unidad 6.
Teoría de la complejidad computacional: Computaciones acotadas en tiempo y espacio. Medida de complejidad temporal. Jerarquía de complejidades temporales. Relaciones entre complejidades.

Unidad 7.Teoría de la Complejidad: Problemas de decisión tratables e intratables. Clases P y NP. NP-completitud. Problemas NP-completos. Ejemplos.

XIII - Imprevistos

XIV - Otros

Correo de contacto: apojavi@gmail.com (Javier Apolloni)