La esencia del curso

Semana $(15)_2 = 1111$

Lógica para Ciencia de la Computación - IIC2213

Prof. Sebastián Bugedo

Programa

Programa

HOY AL FINAL VOCALIZAMOS!

Llegamos al final de nuestro viaje

Examen del curso

Temas de las 4 preguntas

1. Verdadero-falso: aprendizajes esenciales del curso
2. Lógica proposicional
3. Complejidad
4. Lógica de primer orden y teorías

Les daremos un formulario con teoremas y definiciones claves.
¡Salvo teoremas/resultados básicos del curso que se usan en la P1!

Examen del curso

Formato

- Escrito a mano
- 2 horas máximo
- Se entregan las 4 preguntas

Hora y fecha

- miércoles 12 de julio
- 15:30-17:30 (**ES EN LA TARDE**)

Aquellos con tope o imposibilidad de asistir en esa fecha,
ya se contactaron conmigo y les ofrecí una alternativa. Úsenla

Examen del curso

Teorema

Sea $\Sigma = \{w \mid w \text{ es wawín del curso}\}$.Para todo $w \in \Sigma$,

$$w \in \{p \mid p \text{ aprueba el semestre}\}$$

Demostración (Turing me dijo en un sueño)

Las notas están el 14 de julio en la mañana y ahí son

- recorrecciones
- exámenes orales

Turing dice que aprueban todos. $\square$

Objetivos de la clase

- ☐ Conocer estructura del examen
- ☐ Recordar momentos destacados del semestre
- ☐ Resolver dos ejercicios de reducciones
- ☐ Compartir un momento de música

Programa

Última Obertura

Acto único

- Lógica en computación
- Dos ejercicios finales

Último Epílogo

Lógica... ¿Para qué?

Planteamos a la lógica como nuestro **lenguaje** para...

- Estudiar el concepto de **conclusión**
- Modelar y obtener conclusiones
- Formalizar conceptos

¿Cómo comenzamos?

El plan para una lógica

¿Cómo formalizar una lógica?

1. Sintaxis: ¿cómo se escriben *cosas* en la lógica?

   Símbolos   Fórmula

   Conectivos   Construcciones inductivas

2. Semántica: ¿qué significan las *cosas* y cuándo son verdaderas?

   Noción de verdad   Significado de símbolos

   Equivalencia   Consecuencia lógica

3. Deducción: ¿qué se puede deducir basado solo en **forma**?

   Sistemas deductivos

   Correctitud   Completitud

En el semestre vimos 2 lógicas desde este plan

Unidad I

Con la **lógica proposicional** estudiamos el lenguaje más básico

- Conjunto de variables proposicionales $P = \{p_1, \ldots, p_n\}$
- Sintaxis que solo involucra las variables y conectivos
- Semántica y tablas de verdad
- Sistemas deductivos: Resolución (correcto y completo)

¿Qué distingue a esta lógica?

Unidad II

Para adentrarnos en las potencialidades de la lógica, estudiamos la **computabilidad**

- Nos centramos en **lenguajes** = **problemas de decisión**
- Algunos no tienen solución algorítmica
- Podemos comparar su complejidad relativa

¿Qué concepto permite comparar complejidades?

Reducciones

Las **reducciones** nos permitieron clasificar problemas en **clases**

- Reducciones de mapeo:

$$L_1 \leq_m L_2 \quad \Leftrightarrow \quad \text{existe } f \text{ tal que } w \in L_1 \leftrightarrow f(w) \in L_2$$

- Reducciones polinomiales

$$L_1 \leq_p L_2 \quad \Leftrightarrow \quad L_1 \leq_p L_2 \text{ y } f \text{ es polinomial}$$

L_2 es al menos tan difícil como L_1

Decidibilidad

Aquí nos importa distinguir cuándo se puede **decidir** un lenguaje:

¿Existe un algoritmo que diga
si es que un input está o no en el lenguaje?

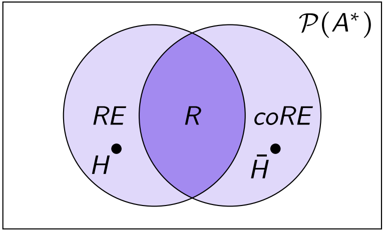
Este concepto es clave, pues habla de la utilidad de un algoritmo:
da una respuesta **SIEMPRE**

Clases de complejidad

Lenguajes **RE**: aceptados por alguna máquina

Lenguajes **coRE**: su complemento es aceptado por alguna máquina

Lenguajes **R** o **decidibles**: aceptados por alguna máquina que siempre se detiene



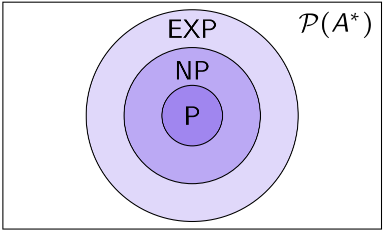
Hasta aquí no hablamos de cuánto tiempo toman las máquinas

Algunas clases dentro de R

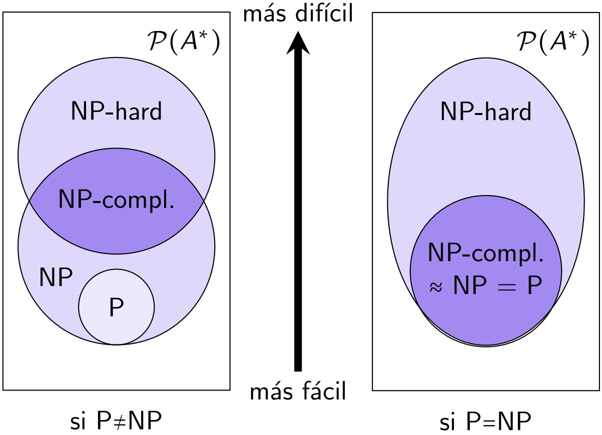
Lenguajes **EXP**: decididos por $\mathcal{M}$ MT en t. exponencial

Lenguajes **NP**: aceptados por $\mathcal{N}$ NMT en t. polinomial

Lenguajes **P**: decididos por $\mathcal{M}$ MT en t. polinomial



El problema abierto P = NP



Unidad III

Con la **lógica de primer orden** extendimos el lenguaje

- Estructuras con dominios arbitrarios
- Sintaxis que involucra variables y cuantificadores
- Semántica más compleja: ¿decidible?
- Sistemas deductivos: Hilbert (correcto y completo)

LPO nos permitió comprender qué se puede definir

Unidad III

Estudiamos el problema de **definibilidad**

- ¿Cuándo se puede definir algo en LPO?
- Conjuntos de elementos y tuplas
- Estructuras

Contamos con estrategias para demostrar definibilidad y no-definibilidad

Unidad IV

Introdujimos el estudio de **teorías** de primer orden

- Propiedades de una teoría
- Posibilidad de axiomatizarlas
- Teorías completas
- Decidibilidad de teorías

Las teorías son la representación natural de un sistema de información

Programa

Última Obertura

Acto único

- Lógica en computación
- Dos ejercicios finales

Último Epílogo

Complejidad

Los siguientes lenguajes son indecidibles

- $A = \{(\mathcal{M}, w) \mid \mathcal{M}$ es una TM det. tal que $\mathcal{M}$ acepta $w\}$
- $L_{\varnothing} = \{(\mathcal{M}) \mid \mathcal{M}$ es una TM det. tal que $L(\mathcal{M}) = \varnothing\}$
- $L_{\epsilon} = \{(\mathcal{M}) \mid \mathcal{M}$ es una TM det. tal que acepta la palabra vacía $\epsilon\}$
- $L_{\text{eq}} = \{(\mathcal{M}_1, \mathcal{M}_2) \mid \mathcal{M}_1, \mathcal{M}_2$ son TM det. con $L(\mathcal{M}_1) = L(\mathcal{M}_2)\}$

Ejercicio

Demuestre que el siguiente lenguaje es indecidible

$L_2 = \{(\mathcal{M}_1, \mathcal{M}_2) \mid \mathcal{M}_1, \mathcal{M}_2$ son TM det. tales que $L(\mathcal{M}_1) \cap L(\mathcal{M}_2) \neq \varnothing\}$

¿Desde qué lenguaje reducimos?

Ejercicio

Para demostrar que el siguiente problema es indecidible

$$L_2 = \{(\mathcal{M}_1, \mathcal{M}_2) \mid \mathcal{M}_1, \mathcal{M}_2 \text{ son TM det. tales que } L(\mathcal{M}_1) \cap L(\mathcal{M}_2) \neq \varnothing\}$$

reduciremos desde

$$L_\epsilon = \{(\mathcal{M}) \mid \mathcal{M} \text{ es una TM det. tal que acepta la palabra vacía } \epsilon\}$$

Consideremos la siguiente máquina que solo acepta la palabra vacía

- Tiene dos estados, q_0 y q_f, donde solo q_f es final
- Tiene una sola transición $\delta(q_0, \sqcup) = (q_f, \sqcup, \triangleright)$
- Esto garantiza que solo avanza al estado final si el primer símbolo es vacío. En otro caso, se detiene en q_0 y rechaza

Llamaremos $\mathcal{M}_\epsilon$ a esta máquina determinista fija.

Ejercicio

Ahora consideremos la reducción $f : \{0,1\}^* \to \{0,1\}^*$ dada por

$$f(w) = (w, \mathcal{M}_\epsilon)$$

Analizamos los dos casos siguientes:

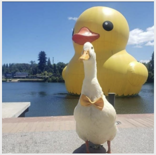
- Si $w \in L_\epsilon$, w es la codificación de una máquina determinista tal que $\epsilon \in L(w)$. Luego, como $L(\mathcal{M}_\epsilon) = \{\epsilon\}$, se cumple que $L(w) \cap L(\mathcal{M}_\epsilon) \neq \varnothing$ y por lo tanto $f(w) \in L_2$.
- Si $w \notin L_\epsilon$, tenemos dos opciones
  - Si w no es una máquina, claramente $f(w) \notin L_2$.
  - Si w es máquina, significa que no acepta la palabra vacía. Como $L(\mathcal{M}_\epsilon) = \{\epsilon\}$, tenemos que $L(w) \cap L(\mathcal{M}_\epsilon) = \varnothing$ y por lo tanto $f(w) \notin L_2$.

Esto demuestra que f es reducción de L_ϵ a L_2. Como L_ϵ es indecidible, L_2 también lo es. $\square$

Ejercicio

Demuestre que el siguiente problema es indecidible

$$\text{CONS} = \{(\Sigma, \varphi) \mid \Sigma \text{ conj. de oraciones y } \varphi \text{ oración tal que } \Sigma \models \varphi\}$$



Ejercicio

Para demostrar que el siguiente problema es indecidible

$$\text{CONS} = \{(\Sigma, \varphi) \mid \Sigma \text{ conj. de oraciones y } \varphi \text{ oración tal que } \Sigma \vDash \varphi\}$$

reduciremos desde

$$\mathrm{VAL} = \{\varphi \mid \varphi \text{ es una oración válida}\}$$

Consideremos la reducción $f : \{0,1\}^* \to \{0,1\}^*$ dada por

$$f(w) = (\{\}, w)$$

Ejercicio

Consideremos la reducción $f : \{0,1\}^* \to \{0,1\}^*$ dada por

$$f(w) = (\{\}, w)$$

Analizamos los dos casos siguientes:

- Si $w \in \mathrm{VAL}$, w es la codificación de una oración válida (i.e. satisfecha por toda estructura). Como $\{\} \vDash \varphi$ si y solo si φ es válida, tenemos que $f(w) = (\{\}, w) \in \mathrm{CONS}$.
- Si $w \notin L_\epsilon$, tenemos dos opciones
  - Si w no es una oración, en cuyo caso $f(w) \notin \mathrm{CONS}$.
  - Si w es oración, significa que no es válida y existe una estructura $\mathfrak{A}$ tal que $\mathfrak{A} \nvDash w$. Luego, como $\mathfrak{A} \vDash \{\}$, pero $\mathfrak{A} \nvDash w$, tenemos que $\{\} \nvDash w$ y $f(w) \notin \mathrm{CONS}$.

Esto demuestra que f es reducción de VAL a CONS. Como VAL es indecidible, CONS también lo es. □

Programa

Objetivos de la clase

- ☐ Conocer estructura del examen
- ☐ Recordar momentos destacados del semestre
- ☐ Resolver dos ejercicios de reducciones
- ☐ Compartir un momento de música

Pasen adelante!

Olvidemos todo y abramos los sentidos

¿Qué me hizo sentir esta última obra?

Ve a

www.menti.com

Introduce el código

6116 6021

O usa el código QR