Problemas recursivamente enumerables

Definición

Un problema L es recursivamente enumerable si existe una máquina de Turing M tal que $L = L(M)$.

Nótese que M en la definición no necesariamente se detiene en todas las entradas.

Problemas recursivamente enumerables

Definición

Un problema L es recursivamente enumerable si existe una máquina de Turing M tal que $L = L(M)$.

Nótese que M en la definición no necesariamente se detiene en todas las entradas.

Ejercicio

Demuestre que U es recursivamente enumerable.

Una caracterización de los problemas recursivamente enumerables

Suponga que L es un lenguaje sobre un alfabeto A.

L es recursivamente enumerable si y sólo si existe un algoritmo que entrega una lista de strings $w_0, w_1, \ldots$ en A^* tal que:

- Cada w_i en la lista de salida está en L
- Para cada $w \in L$, existe $i \in \mathbb{N}$ tal que $w = w_i$

Vale decir, el algoritmo enumera los elementos de L.

Una caracterización de los problemas recursivamente enumerables

Suponga que L es un lenguaje sobre un alfabeto A.

L es recursivamente enumerable si y sólo si existe un algoritmo que entrega una lista de strings $w_0, w_1, \ldots$ en A^* tal que:

- Cada w_i en la lista de salida está en L
- Para cada $w \in L$, existe $i \in \mathbb{N}$ tal que $w = w_i$

Vale decir, el algoritmo enumera los elementos de L.

Ejercicios

1. Muestre que la afirmación anterior es cierta
2. De un algoritmo que enumere los elementos de U

Problemas decidibles versus problemas recursivamente enumerables

Teorema

Si un lenguaje es decidible entonces es recursivamente enumerable.

Además, existen lenguajes recursivamente enumerables y no decidibles

- U es un ejemplo.

Problemas decidibles versus problemas recursivamente enumerables

Teorema

Si un lenguaje es decidible entonces es recursivamente enumerable.

Además, existen lenguajes recursivamente enumerables y no decidibles

- U es un ejemplo.

¿Es H recursivamente enumerable? ¿Podemos deducir esto a partir del hecho que U es recursivamente enumerable?

La noción de reducción nos puede ayudar . . .

Teorema

Suponga que L_1 es reducible a L_2. Si L_2 es recursivamente enumerable, entonces L_1 es recursivamente enumerable.

Del teorema también deducimos que si L_1 no es recursivamente enumerable, entonces L_2 no es recursivamente enumerable.

La noción de reducción nos puede ayudar . . .

Teorema

Suponga que L_1 es reducible a L_2. Si L_2 es recursivamente enumerable, entonces L_1 es recursivamente enumerable.

Del teorema también deducimos que si L_1 no es recursivamente enumerable, entonces L_2 no es recursivamente enumerable.

Ejercicio

Demuestre el teorema

La noción de reducción nos puede ayudar . . .

Podemos concluir entonces que H es recursivamente enumerable.

- Puesto que existe una reducción de H a U

La noción de reducción nos puede ayudar . . .

Podemos concluir entonces que H es recursivamente enumerable.

- Puesto que existe una reducción de H a U

¿Existen lenguajes indecidibles que no son recursivamente enumerables?

La noción de reducción nos puede ayudar . . .

Podemos concluir entonces que H es recursivamente enumerable.

- Puesto que existe una reducción de H a U

¿Existen lenguajes indecidibles que no son recursivamente enumerables?

- Un argumento de conteo nos dice que sí

La noción de reducción nos puede ayudar . . .

Podemos concluir entonces que H es recursivamente enumerable.

- Puesto que existe una reducción de H a U

¿Existen lenguajes indecidibles que no son recursivamente enumerables?

- Un argumento de conteo nos dice que sí

¿Puede dar algún ejemplo de un lenguaje indecidible y no recursivamente enumerable?

Un teorema útil

Vamos a ver un teorema que nos va a dar ejemplos de lenguajes indecidibles y no recursivamente enumerables.

- Puede ser útil para demostrar que un lenguaje es decidible

Dado un lenguaje L sobre un alfabeto A, definimos $\overline{L}$ como $(A^* \setminus L)$.

Un teorema útil

Vamos a ver un teorema que nos va a dar ejemplos de lenguajes indecidibles y no recursivamente enumerables.

- Puede ser útil para demostrar que un lenguaje es decidible

Dado un lenguaje L sobre un alfabeto A, definimos $\overline{L}$ como $(A^* \setminus L)$.

Teorema

Si un lenguaje L y su complemento $\overline{L}$ son recursivamente enumerables, entonces L es decidible.

Un teorema útil

Vamos a ver un teorema que nos va a dar ejemplos de lenguajes indecidibles y no recursivamente enumerables.

- Puede ser útil para demostrar que un lenguaje es decidible

Dado un lenguaje L sobre un alfabeto A, definimos $\overline{L}$ como $(A^* \setminus L)$.

Teorema

Si un lenguaje L y su complemento $\overline{L}$ son recursivamente enumerables, entonces L es decidible.

Ejercicio

Demuestre el teorema

Un teorema útil

Dado que U es indecidible y recursivamente enumerable, obtenemos lo siguiente:

Corolario

$\overline{U}$ es indecidible y no recursivamente enumerable

Un teorema útil

Dado que U es indecidible y recursivamente enumerable, obtenemos lo siguiente:

Corolario

$\overline{U}$ es indecidible y no recursivamente enumerable

Obtenemos otro corolario muy interesante: $\overline{U}$ no es reducible a U

- ¿Por qué?

Un teorema útil

Dado que U es indecidible y recursivamente enumerable, obtenemos lo siguiente:

Corolario

$\overline{U}$ es indecidible y no recursivamente enumerable

Obtenemos otro corolario muy interesante: $\overline{U}$ no es reducible a U

- ¿Por qué?

Vale decir, no es cierto que existan reducciones entre cualquier par de problemas indecidibles.

Motivando la noción de completitud

Sea L un lenguaje sobre un alfabeto A.

Lema

L es recursivamente enumerable si y sólo si existe una máquina de Turing M tal que para todo $w \in A^*$:

$$w \in L \text{ si y sólo si } M \text{ se detiene con entrada } w$$

Motivando la noción de completitud

Sea L un lenguaje sobre un alfabeto A.

Lema

L es recursivamente enumerable si y sólo si existe una máquina de Turing M tal que para todo $w \in A^*$:

$$w \in L \text{ si y sólo si } M \text{ se detiene con entrada } w$$

Ejercicio

Demuestre el lema.

Motivando la noción de completitud

Consideramos los lenguajes sobre el alfabeto $\{0, 1\}$.

Teorema

Si L es un lenguaje recursivamente enumerable, entonces L es reducible a U.

Motivando la noción de completitud

Consideramos los lenguajes sobre el alfabeto $\{0, 1\}$.

Teorema

Si L es un lenguaje recursivamente enumerable, entonces L es reducible a U.

Podemos decir entonces que U es un problema "completo" para la clase de los problemas recursivamente enumerables.

- Recuerde que U es recursivamente enumerable

Motivando la noción de completitud

Consideramos los lenguajes sobre el alfabeto $\{0, 1\}$.

Teorema

Si L es un lenguaje recursivamente enumerable, entonces L es reducible a U.

Podemos decir entonces que U es un problema "completo" para la clase de los problemas recursivamente enumerables.

- Recuerde que U es recursivamente enumerable

Vamos a formalizar esta noción de completitud más adelante.

- La vamos a estudiar en detalle para los problemas decidibles

Demostración del teorema

Sea L un lenguaje recursivamente enumerable sobre el alfabeto $\{0,1\}$.

Por lema, existe una máquina de Turing M tal que para todo $w \in \{0,1\}^*$:

$$w \in L \text{ si y sólo si } M \text{ se detiene con entrada } w$$

Considere la función $f : \{0,1\}^* \to \{0,1\}^*$ definida de la siguiente forma:

$$f(w) \quad = \quad C(M)0000w$$

Demostración del teorema

Tenemos que f es una función computable.

Pero además tenemos que:

$$\begin{aligned} w \in L &\Leftrightarrow M \text{ se detiene con entrada } w \\ &\Leftrightarrow C(M)0000w \in U \\ &\Leftrightarrow f(w) \in U \end{aligned}$$

Concluimos que f es una reducción de L a U. □

Un corolario interesante

Del teorema obtenemos la siguiente caracterización de los problemas recursivamente enumerables.

- Consideramos los lenguajes sobre el alfabeto $\{0, 1\}$

Corolario

Un lenguaje L es recursivamente enumerable si y solo si L es reducible a U.

Algunas propiedades útiles de reducción

Proposición

1. *Si L_1 es reducible a L_2 y L_2 es reducible a L_3, entonces L_1 es reducible a L_3*
2. *Si L_1 es reducible a L_2, entonces $\overline{L_1}$ es reducible a $\overline{L_2}$.*

Algunas propiedades útiles de reducción

Proposición

1. *Si L_1 es reducible a L_2 y L_2 es reducible a L_3, entonces L_1 es reducible a L_3*
2. *Si L_1 es reducible a L_2, entonces $\overline{L_1}$ es reducible a $\overline{L_2}$.*

Ejercicio

Demuestre la proposición.

Un problema indecidible más "difícil"

Defina D como el siguiente lenguaje:

$$D = \{(M_1, w_1, M_2, w_2) \mid M_1 \text{ se detiene con entrada } w_1 \text{ y } M_2 \text{ no se detiene con entrada } w_2\}$$

Utilizamos una notación de tupla para definir D.

- ¿Cómo se podría definir este lenguaje si sólo utilizamos strings?

Un problema indecidible más "difícil"

Tenemos los siguiente resultados sobre D:

Un problema indecidible más "difícil"

Tenemos los siguiente resultados sobre D:

1. U es reducible a D

Concluimos que D es indecidible

Un problema indecidible más "difícil"

Tenemos los siguiente resultados sobre D:

1. U es reducible a D

   Concluimos que D es indecidible

2. Del punto 1 obtenemos que $\overline{U}$ es reducible a $\overline{D}$

   Concluimos que $\overline{D}$ no es recursivamente enumerable

Un problema indecidible más "difícil"

Tenemos los siguiente resultados sobre D:

1. U es reducible a D

   Concluimos que D es indecidible

2. Del punto 1 obtenemos que $\overline{U}$ es reducible a $\overline{D}$

   Concluimos que $\overline{D}$ no es recursivamente enumerable

3. $\overline{U}$ es reducible a D

   Concluimos que D no es recursivamente enumerable

Un problema indecidible más "difícil"

4. Del punto 3 obtenemos que si D es reducible a U, entonces $\overline{U}$ es reducible a U
   - Concluimos que D no es reducible a U

Un problema indecidible más "difícil"

4. Del punto 3 obtenemos que si D es reducible a U, entonces $\overline{U}$ es reducible a U
   - Concluimos que D no es reducible a U

5. Del punto 2 obtenemos que si D es reducible a $\overline{U}$, entonces $\overline{U}$ es reducible a U
   - Concluimos que D no es reducible a $\overline{U}$

Un problema indecidible más "difícil"

4. Del punto 3 obtenemos que si D es reducible a U, entonces $\overline{U}$ es reducible a U
   - Concluimos que D no es reducible a U

5. Del punto 2 obtenemos que si D es reducible a $\overline{U}$, entonces $\overline{U}$ es reducible a U
   - Concluimos que D no es reducible a $\overline{U}$

¡D puede ser considerado un problema indecidible más difícil que U!

Motivando la noción de clase de complejidad

Vamos a considerar los lenguajes sobre el alfabeto $\{0, 1\}$.

Motivando la noción de clase de complejidad

Vamos a considerar los lenguajes sobre el alfabeto $\{0, 1\}$.

Hasta ahora hemos considerado las siguientes clases de complejidad:

$$\begin{aligned} \mathrm{R} &= \{L \mid L \text{ es un lenguaje decidible}\} \\ \mathrm{RE} &= \{L \mid L \text{ es un lenguaje recursivamente enumerable}\} \end{aligned}$$

Motivando la noción de clase de complejidad

Vamos a considerar los lenguajes sobre el alfabeto $\{0, 1\}$.

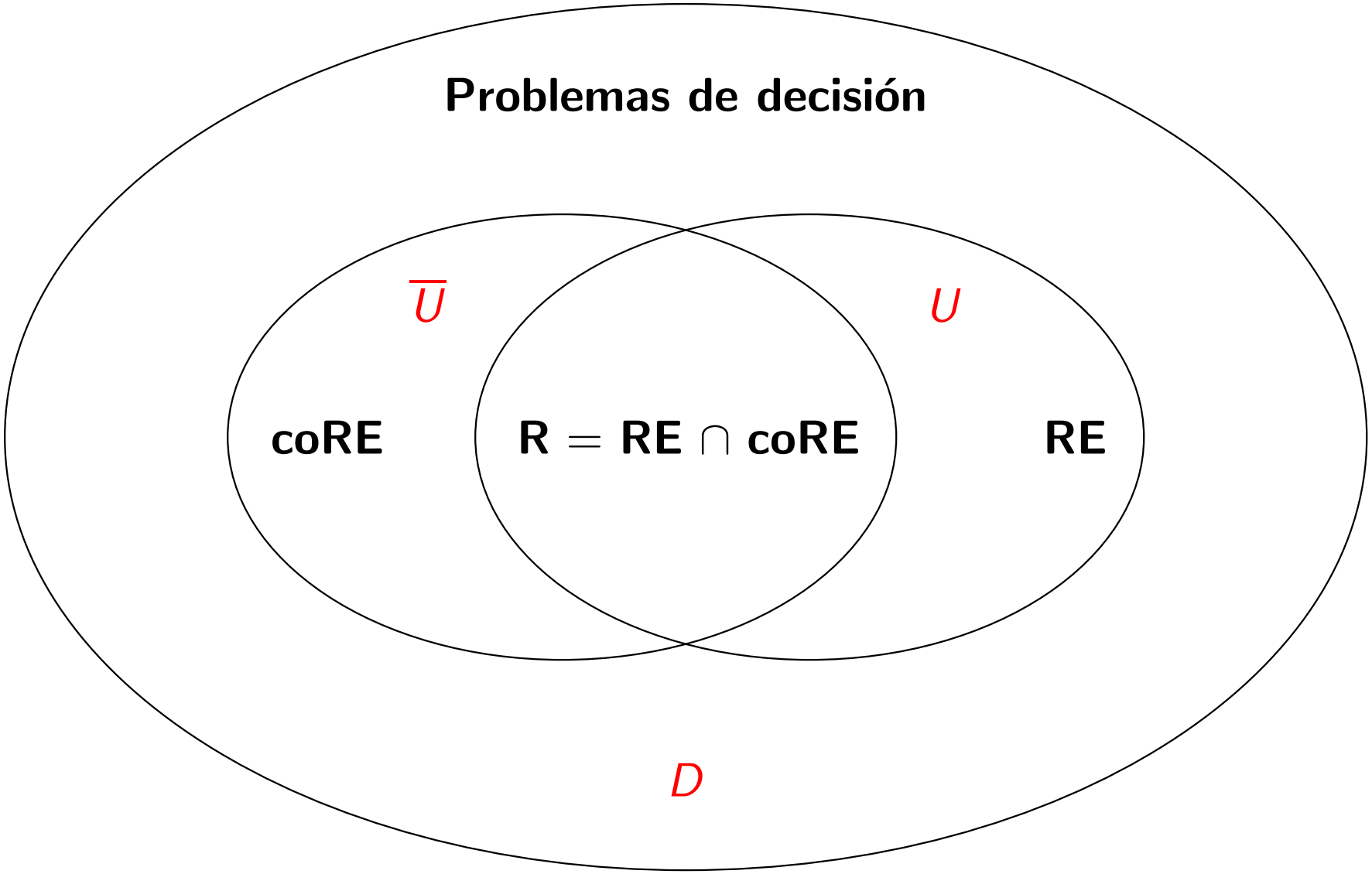
Hasta ahora hemos considerado las siguientes clases de complejidad:

$$\begin{aligned} \mathsf{R} &= \{L \mid L \text{ es un lenguaje decidible}\} \\ \mathsf{RE} &= \{L \mid L \text{ es un lenguaje recursivamente enumerable}\} \end{aligned}$$

También consideramos el complemento de los lenguajes recursivamente enumerables, lo que da origen a la siguiente clase de complejidad:

$$\mathsf{coRE} = \{L \mid \overline{L} \text{ es un lenguaje recursivamente enumerable}\}$$

Un resumen de los resultados obtenidos



Un resumen de los resultados obtenidos

Además tenemos que:

- U es completo para RE: $U \in \text{RE}$ y para cada $L \in \text{RE}$ se tiene que L es reducible a U
  - Concluimos que $\overline{U}$ es completo para coRE: $\overline{U} \in \text{coRE}$ y para cada $L \in \text{coRE}$ se tiene que L es reducible a $\overline{U}$
- $\overline{U}$ no es reducible a U
  - Concluimos que U no es reducible a $\overline{U}$

Un resultado final para completar la figura

Teorema (Friedberg & Muchnik)

Existe un lenguaje $L \in RE$ tal que $L \notin R$ y L no es completo para RE

- *Vale decir, existe $L' \in RE$ tal que L' no es reducible a L*