UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
Facultad de Ingeniería

Ingeniería en Ciencias de la Computación

TEORÍA DE LA COMPUTACIÓN
Problemas no resolubles por una máquina de Turing y problema de la parada

Trabajo de: ADRIAN A. GONZÁLEZ DOMÍNGUEZ *[359834]*
Asesor: MARIO ANDRES CUEVAS GUTIERREZ

11 de noviembre de 2024

Problemas no resolubles por una MT y el problema de la Parada

Teoría de la computación | Noviembre 11, 2024

Existen diversos ejemplos de problemas irresolubles o dentro de una categoría de estos, indecidibles, siendo el más claro y conocido ejemplo, el problema de la parada que más adelante se explica. Otros problemas irresolubles son:

- Lenguaje que no se acepta a sí mismo. Dados 2 lenguajes
  $NSA_1 = \{w \mid w \in \{a,b\}^*, w = d(T)$ para una máquina de Turing T y T no acepta $w\}$
  $NSA_2 = \{w \mid w \in \{a,b\}^*, w \neq d(T)$ para cualquier máquina de Turing $T\}$
  donde $d(T)$ es la descripción de la máquina de Turing T.
  NSA_1 es el conjunto de cadenas que describen a la máquina T pero no son aceptadas por esta. NSA_2 es el conjunto de cadenas que no describe ni una máquina de Turing. Ninguno está vacío.
  Suponiendo que T acepta el conjunto $\{a\}$, y $w = d(T)$, w estaría describiendo al T que acepta w, pero al no ser más larga, no es aceptable. Ahora para NSA_2 $w = a$, por lo que no hay máquina descrita por la cadena a. Para describir una MT requerimos más de un solo símbolo, por lo que es aceptable por NSA_2. Ni NSA_1 ni NSA_2 están vacíos.
  $NSA = NSA_1 \cup NSA_2$ es decir que no existen MT que acepten el lenguaje NSA
  Exista una MT T_0 y una descripción $w_0 = d(T_0)$. T_0 acepta NSA.
  Como NSA es un lenguaje, w_0 está o no en NSA. Si T_0 acepta w_0, $w_0 \in NSA$ porque T_0 acepta NSA, lo cual es una contradicción. Si T_0 no acepta w_0, w_0 no esta en NSA porque T_0 acepta NSA, pero $w_0 = d(T_0)$ porque así seleccionamos w_0 según NSA_1. Pero T_0 no acepta w_0, así que está en NSA_1. Sin embargo, es "Self Accepting", lo que nuevamente se contradice.

Otros problemas:

- Acepta(Λ). Pregunta si una MT acepta la cadena vacía Λ. Se demuestra irresoluble disminuyendolo al problema de la parada.
- AceptaTodo(). Pregunta si una MT acepta todas las cadenas posibles de todo alfabeto dado. Se demuestra irresolable reduciendolo a Acepta(Λ)
- AceptaNada() y Equivalencia(). Preguntan el primero si una máquina de turing no acepta nada y Equivalence si 2 MT aceptan el mismo lenguaje. Son reducibles al caso anterior.
- Problemas sobre gramáticas libres de contexto, como la pertenencia de un lenguaje a oto y la complementariedad e intersección de lenguajes son irresolables.

El problema de la parada busca determinar si, dada una MT M y una cadena de entrada w, la máquina M se detendrá en algún momento o se ejecutará indefinidamente al procesar una entrada. Para demostrar la indecibilidad de el problema, se hace una prueba de contradición.

Suponemos una máquina T que dice si M se detiene con w o no, es decir resuelve el problema de la parada.

En base a T construimos la máquina Tm:

- Si T dice que M se detiene en w, Tm entra en bucle.
- Si T dice que M no se detiene en w, Tm se detiene

Con Tm construimos Te como sigue. Te toma d(M), la descripción de una MT, y la convierte en d(M)*d(M) (* separa las copias), y el resultado lo pasa a Tm

- Si Te recibe d(Te), genera d(Te)*d(Te) y se la da a Tm
- Ya que Tm hace lo contrario de T, Tm hara lo contrario a lo que T predice de Te.

La contradicción
- Si Te se detiene sobre d(Te), entonces Tm entra en bucle infinito.
- Si Te no se detiene en d(Te), Tm se detiene.
Te no debería detenerse y no detenerse al mismo tiempo.

Por último, ¿qué impacto dejan los problemas irresolubles en las ciencias de la computación?
- Nos dan límites a la resolución de problemas. Sabemos que no podemos resolver todos los problemas, por lo que tendremos que aprender a hacer aproximaciones de aquellos que no han sido resueltos.
- Permiten el avance tecnológico. Métodos como Machine Learning o algoritmos genéticos de selección, combinación, sexuales, etc. se inspiran en esto para aproximar.
- Cercanía a la realidad. La realidad no tiene respuestas exactas a todo, una mala respuesta explica más que su ausencia.

Fuentes de información:

- Casero, A. (2024, 15 marzo). ¿Qué son los problemas algorítmicos no resolubles? KeepCoding Bootcamps. https://keepcoding.io/blog/problemas-algoritmicos-no-resolubles/
- CS390 Introduction to Theoretical Computer Science (2013, 14 enero). https://www.cs.odu.edu/~toida/nerzic/390teched/web_course.html
- What unsolved problems in theoretical computer science other than P≠NP would make a real world impact in the field? (2020). Quora. https://www.quora.com/What-unsolved-problems-in-theoretical-computer-science-other-than-P-NP-would-make-a-real-world-impact-in-the-field/