



# EntscheidungProblem

Kuliah Teori Bahasa dan Automata  
Program Studi Ilmu Komputer  
Fasilkom UI

Prepared by:  
Suryana Setiawan

# Adakah Yang Lebih Powerful dari TM?

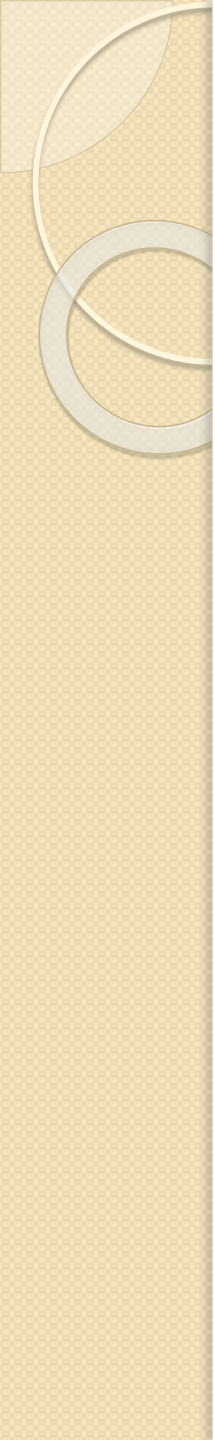
- Ingat, berdasarkan powernya maka:  $FSM < PDA < TM$
- Banyaknya TM yang legal adalah countably infinite.
  - Karena, string-string dapat dienumerasi secara proper-order dan mengkaitkannya dengan mesin-mesin Turing.
- Banyaknya semua kemungkinan bahasa dalam suatu alfabet tak kosong adalah uncountably infinite (uncountable)
- Jadi, lebih banyak bahasa dibanding TM (atau masih terdapat banyak bahasa yang tidak ada TM-nya!!!).
- Maka, adakah mesin yang lebih powerful dari TM?
- Dari segi komputasi, adakah algoritma yang tidak dapat diimplementasikan oleh TM?
  - Jika ada, maka mesin yang lebih powerful itu juga ada!

# Entscheidungsproblem

- “Decision problem” dalam bahasa Jerman.
- Terdapat 3 cara ekivalen dalam menyatakannya:
  - I: Apakah terdapat algoritma untuk memutuskan, untuk sembarang kalimat  $w$  dalam first-order-logic, apakah  $w$  valid?
  - II: Diketahui satu himpunan aksioma  $A$  dan kalimat  $w$ , adakah algoritma untuk memutuskan  $w$  benar sebagai akibat dari  $A$ ?
  - III: Diketahui satu himpunan aksioma  $A$  dan kalimat  $w$ , adakah algoritma untuk memutuskan bahwa  $w$  dapat dibuktikan dari  $A$ ?

# Formalisasi Algoritma

- Entscheidungproblem menarik perhatian logician (ahli ilmu logika) termasuk Alan Turing dan Alonzo Church.
- Dalam rangka menjawabnya, perlu memformalisasi apa yang dimaksud dengan algoritma.
  - Formalisasi yang dibuat Turing (1936), kemudian menjadi Mesin Turing → lebih prosedural
  - Formalisasi yang dibuat Church (1936), kemudian menjadi Lambda Calculus → lebih fungsional
- Keduanya nampak berbeda tetapi ternyata kemudian dibuktikan ekuivalen



# Church-Turing Thesis

- Suatu fungsi bilangan natural adalah komputabel (i.e., oleh manusia menggunakan kertas&pensil) **iff** fungsi tsb komputabel oleh Mesin Turing.
- Terdapat banyak alternatif formalisme selain TM, namun tidak ada yang lebih powerful.

# Contoh-contoh Formalisme yang Ekuivalen dengan TM

- Komputer modern (dengan asumsi memory tak berhingga)
- Lambda Calculus (basis dari Functional Programming)
- Fungsi Rekursif Parsial
- Tag Systems (PDA dengan FIFO, bukannya LIFO)
- Unrestricted Grammars (Rule berbentuk  $\alpha \rightarrow \beta$  dengan  $\alpha \in V^+$ ,  $\beta \in V^*$ )
- Post Production Systems
- Conway's Game of Life (Two dimensional Cellular Automata)
- One Dimensional Cellular Automata
- Berbagai model teoritis Komputasi berbasis DNA
- Lindenmayer Systems

# Mengingat Kembali

- TM  $M$  memutuskan (decides)  $L$ , **iff**:  $\forall w \in \Sigma^*$ :
  - Jika  $w \in L$  maka  $M$  menerima  $w$ , dan
  - Jika  $w \notin L$  maka  $M$  menolak  $w$ .
- $L$  bahasa **decidable** ( $L \in D$ ) **iff** terdapat TM  $M$  yang memutuskan  $L$ .
- TM semi-memutuskan (semidecides)  $L$ , **iff**:  $\forall w \in \Sigma^*$ :
  - Jika  $w \in L$  maka  $M$  menerima  $w$ , dan
  - Jika  $w \notin L$  maka  $M$  tidak menerima  $w$  (menolak atau tidak halt).
- $L$  bahasa **semidecidable** ( $L \in SD$ ) **iff** terdapat TM  $M$  yang semi-memutuskan  $L$ .



# Bagaimana dengan Bahasa-bahasa Ini?

- $L_1 = \{ \langle M, w \rangle : \text{Mesin Turing } M \text{ halt pada string input } w \}$
- $L_2 = \{ \langle M \rangle : \text{tidak ada string yang membuat Mesin Turing } M \text{ halt} \}$
- $L_3 = \{ \langle M_a, M_b \rangle : M_a \text{ dan } M_b \text{ adalah dua TM yang halt pada string-string yang sama} \}$
- Note:
  - $\langle M \rangle$  adalah string hasil pengkodean TM  $M$ .
  - $\langle M, w \rangle$  adalah sepasang string  $\langle M \rangle; \langle w \rangle$  dengan  $\langle w \rangle$  adalah string hasil pengkodean input string  $w$  dengan cara pengodean yang sama dengan  $\langle M \rangle$ .
  - $\langle M_a, M_b \rangle$  sepasang string  $\langle M_a \rangle; \langle M_b \rangle$  keduanya merupakan hasil pengkodean dua TM  $M_a$  dan  $M_b$ .

# Halt Problem

- $H = \{ \langle M, w \rangle : \text{Mesin Turing } M \text{ halt pada string input } w \}$
- Bahasa  $H$  ini (atau  $L_1$  di halaman sebelumnya):
  - Mudah dinyatakan dan dipahami
  - Kepentingan praktis (program-correctness checker)
  - Semidecidable (SD)
  - Nondecidable ( $\neg D$ )
- Implikasi:  $H$  merupakan kunci perbedaan antara D dan SD.
  - Karena, bila  $H$  adalah D maka semua SD juga adalah D! Nyatanya  $H$  bukan D.

# Apa Kaitannya dengan Entscheldungsproblem?

- TM didefinisikan demi menjawab “Diketahui satu himpunan aksioma  $A$  dan kalimat  $w$ , adakah algoritma untuk memutuskan  $w$  benar sebagai akibat dari  $A$ ?”
- Formalisasi algoritma membawa pada definisi TM.
- Sementara berada dalam  $H$ , apakah TM  $M$  halt untuk input  $w$ ?
- Jadi, tidak ada solusi untuk Entscheldungsproblem.