



Decidable & Semidecidable (3)

Kuliah Teori Bahasa dan Automata
Program Studi Ilmu Komputer
Fasilkom UI

Prepared by:
Suryana Setiawan

Metoda Reduksi Lain

- Pada sejumlah kasus mapping reduction langsung tidak memungkinkan.
- Diperlukan fungsi lain pada aplikasi tsb Oracle.
- Contoh: Pertanyaan “Apakah M **menerima** string-string yang panjangnya **bukan** bilangan genap?” bukan D.
- Mapping reduction langsung menyebabkan Oracle bereaksi tebalik.

Pertanyaan Terkait Mesin Turing

- Banyak pertanyaan akan sifat-sifat mesin-mesin Turing yang undecidable.
- Apakah untuk setiap mesin Turing selalu demikian? **Tidak.**
- Pertanyaan terkait struktur fisik mesin **cenderung decidable.**
- Contoh decidable:
 - pertanyaan “banyaknya status dari mesin Turing M ”.
 - Pertanyaan “apakah mesin Turing M halt dalam sekian langkah tertentu?”
 - Pertanyaan “mesin turing M bergerak ke kanan tepat dua kali ketika bekerja untuk input w ”

Bahasa Not Semidecidable

- Membuktikan bahasa L_2 bukan SD dengan *mapping reducibility* (ide pembuktian sama dengan pembuktian suatu bahasa bukan D)
 - Sudah diketahui $L_1 \notin \text{SD}$, dan
 - L_1 dapat direduksi menjadi L_2 .
- Cara lain untuk membuktikan bahwa suatu bahasa L_2 bukan SD adalah dengan menunjukkan bahwa tidak ada prosedur enumerasi satu per satu elemen L_2 (*uncountable set*)

Teorema Rice

- Teorema Rice:

Untuk suatu properti non trivial P , bahasa $L = \{ \langle M \rangle : P(L(M)) = \text{TRUE} \}$ bukan D

- Untuk menerapkan teorema Rice

- Spesifikasikan properti P
- Tunjukkan bahwa domain P adalah himpunan bahasa SD
- Tunjukkan bahwa P non trivial:
 - P bernilai TRUE untuk sekurang-kurangnya satu bahasa
 - P bernilai FALSE untuk sekurang-kurangnya satu bahasa

Contoh Penggunaan Teorema Rice

- $L_1 = \{ \langle M \rangle : M \text{ adalah mesin Turing dan } L(M) \text{ hanya mengandung string panjang ganjil} \}$
- $L_2 = \{ \langle M \rangle : M \text{ adalah mesin Turing dan } L(M) \text{ reguler} \}$
- $L_3 = \{ \langle M \rangle : \text{Mesin Turing } M \text{ terdiri dari 10 states} \}$
- $L_4 = \{ \langle M \rangle : \text{Mesin Turing } M \text{ accept } \varepsilon \text{ dalam 10 langkah komputasi} \}$

- L_1 dan L_2 memiliki properti P, sedangkan L_3 dan L_4 tidak
 - Pada L_1 , P bernilai TRUE if $(\forall w \in L_1, |w| \text{ ganjil})$, FALSE sebaliknya
 - Pada L_2 , P bernilai TRUE if $(L(M) \text{ reguler})$, FALSE sebaliknya
- Teorema Rice hanya dapat diterapkan pada L_1 dan L_2 . L_3 dan L_4 terkait dengan properti mesin

Penjelasan Contoh Penggunaan Teorema Rice

- $L_2 = \{ \langle M \rangle : M \text{ adalah mesin Turing dan } L(M) \text{ reguler} \}$
- P bernilai TRUE if $(L(M) \text{ reguler})$, FALSE sebaliknya
- Domain P adalah himpunan bahasa SD karena :
 - regularitas suatu bahasa bisa ditunjukkan dengan FSM
 - seluruh bahasa yang bisa dikomputasi dengan FSM pasti bisa dikomputasi oleh suatu mesin Turing
 - suatu bahasa yang bisa dikomputasi dengan mesin Turing adalah bahasa SD
- Secara non trivial, dapat dibuktikan
 - $P(a^*)$ bernilai TRUE
 - $P(a^n b^n)$ bernilai FALSE
- Kesimpulan: L_2 bukan bahasa D.

Practical Impact of These Results

$P \rightarrow$ Program

1. Does P , when running on x , halt?
 2. Might P get into an infinite loop on some input?
 3. Does P , when running on x , ever output a 0? Or anything at all?
 4. Are P_1 and P_2 equivalent?
 5. Does P , when running on x , ever assign a value to n ?
 6. Does P ever reach S on any input (in other words, can we chop it out?)
 7. Does P reach S on every input (in other words, can we guarantee that S happens)?
- Can the Patent Office check prior art?
 - Can the CS department buy the definitive grading program?

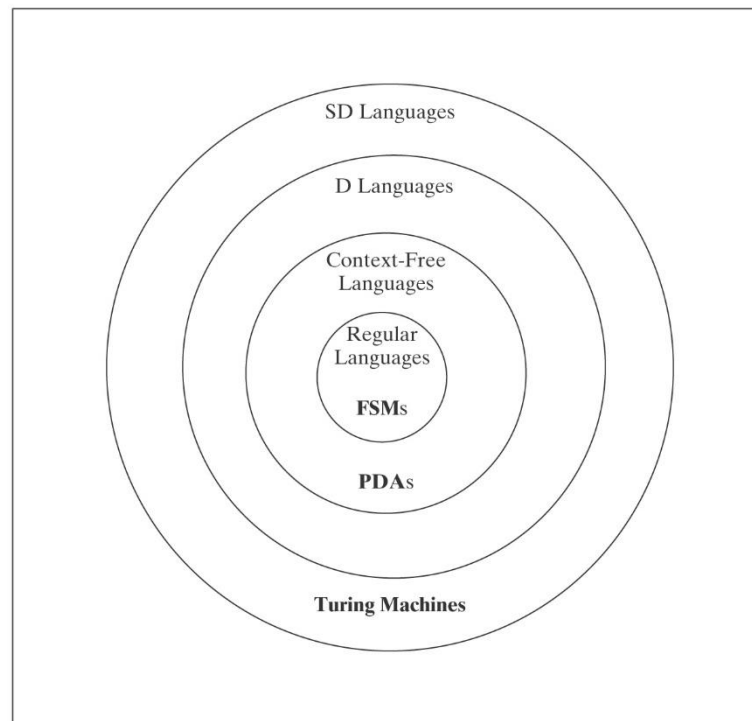
Bahasa \neg SD

Bahasa \neg SD berjumlah *uncountably infinite*.

Bahasa \neg SD tidak dapat direpresentasikan oleh Mesin Turing

Ada *countably infinite* TM (berarti Bahasa SD).

Dengan demikian Bahasa yang tidak termasuk SD (\neg SD) lebih banyak lagi.



Pembuktian Bahasa \neg SD

- Secara intuitif, Bahasa yang termasuk dalam \neg SD dapat dibuktikan dengan adanya infinite search atau diketahui bahwa mesin turingnya akan mengalami infinite loop.
- Contoh
 - \neg H = $\{ \langle M, w \rangle : \text{TM } M \text{ tidak halt untuk suatu } w \}$.
 - $\{ \langle M \rangle : L(M) = \Sigma^* \}$.
 - $\{ \langle M \rangle : \text{tidak ada string yang membuat TM halts} \}$.

Bahasa-bahasa \neg SD

- Setiap bahasa L di dalam $(SD - D)$ berimplikasi
Kasus 1: $\neg L$ adalah bahasa \neg SD, maka dengan dalam usaha membuktikan L' lalu ternyata “ $\neg L'$ merupakan $(SD - D)$ ”, maka “ L' adalah \neg SD”.
Kasus 2: jika $\neg L$ adalah \neg SD maka L bisa juga \neg SD atau SD!
- Untuk memeriksa L di kasus kedua, maka reduksi sebelumnya dapat diterapkan dengan $\neg H$ sebagai basis reduksinya.
 - “Bila $\neg H$ dapat direduksi jadi L , jika asumsi L sebagai SD dapat menyebabkan $\neg H$ juga SD, berarti asumsi tersebut tidak benar. Berarti juga L adalah \neg SD.”

$\{ \langle M \rangle : L(M) = \Sigma^* \}$ adalah \neg SD

- Problem view: “diberikan suatu program/mesin M apakah ia dapat menerima string apapun?”
- Untuk menjawab ini maka TM yang memeriksanya harus memanggil string generator Σ^* dan memanggil UTM untuk memeriksa setiap string w yang dihasilkan dengan simulasi M .
- Walaupun $\langle M \rangle$ benar anggota bahasa ini, pemeriksaannya tidak akan pernah selesai.
- Komplementnya, $\{ \langle M \rangle : L(M) \neq \Sigma^* \}$ juga \neg SD karena walaupun cukup memerlukan satu string yang tidak diterima M , tidak ada jaminan string tsb membuat M halt.

$\{ \langle M \rangle : \text{tidak ada string yang membuat } M \text{ halt} \}$ adalah $\neg\text{SD}$

- Problem view: “diberikan suatu program/mesin M apakah ia tidak halt untuk string apapun?”
- Untuk menjawab ini maka TM yang memeriksanya harus memanggil string generator Σ^* dan memanggil UTM secara dovetailing untuk memeriksa setiap string w yang dihasilkan dengan simulasi paralel M .
- Jika $\langle M \rangle \in L$, maka pemeriksaannya tidak akan pernah halt, jika tidak, maka akan halt.
- Bahasa ini $\neg\text{SD}$ tapi komplemennya SD.