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 lansung 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 $L_1 \notin$ 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 = {<*M*>: $P(L(M))$ = TRUE} bukan *D*

- Untuk menerapkan teorema Rice
  - Spesifikasikan properti *P*
  - Tunjukkan bahwa domain *P* adalah himpunan bahasa SD
  - Tunjukkan bahwa:
    - *P* bernilai TRUE untuk sekurang-kurangnya satu bahasa
    - *P* bernilai FALSE untuk sekurang-kurangnya satu bahasa

Contoh Penggunaan Teorema Rice

- L_1 = {<M>: M adalah mesin Turing dan $L(M)$ hanya mengandung string panjang ganjil}
- L_2 = {<M>: M adalah mesin Turing dan $L(M)$ reguler}
- L_3 = {<M>: Mesin Turing M terdiri dari 10 states}
- L_4 = {<M>: Mesin Turing M accept ε 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 ($\forall w \in L_1$, $|w|$ ganjil), FALSE sebaliknya
  - Pada L_2 , P bernilai TRUE if ($L(M)$ reguler, FALSE sebaliknya
- Teorema Rice hanya dapat diterapkan pada L_1 dan L_2

Penjelasan Contoh Penggunaan Teorema Rice

- L_2 = {<M>: M adalah mesin Turing dan $L(M)$ reguler}
- P bernilai TRUE if ($L(M)$ 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.

Bahasa-bahasa ¬SD

- Karena setiap bahasa L di dalam (SD – D) berimplikasi

  Kasus 1: ¬L adalah bahasa ¬SD, maka dengan dalam usaha membuktikan L' lalu ternyata "¬L' merupakan (SD – D)", maka "L' adalah ¬SD".

  Kasus 2: jika ¬L adalah ¬SD maka L bisa juga ¬SD atau SD!

- Untuk memeriksaL di kasus kedua, maka reduksi sebelumnya dapat diterapkan dengan ¬H sebagai basis reduksinya.
  - "Bila ¬H dapat direduksi jadi L, jika asumsi L sebagai SD dapat menyebabkan ¬H juga SD, berarti asumsi tersebut tidak benar. Berarti juga L adalah ¬SD."

$\{<M>: L(M) = \Sigma^*\}$ 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 $<M>$ benar anggota bahasa ini, pemeriksaannya tidak akan pernah selesai.
- Komplemennya, $\{<M>: L(M) \neq \Sigma^*\}$ juga $\neg$SD karena walaupun cukup memerlukan satu string yang tidak diterima M, tidak ada jaminan string tsb membuat M halt.

{<*M*>: tidak ada string yang membuat *M halt*} adalah ¬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 <*M*> $\in$ L, maka pemeriksaannya tidak akan pernah halt, jika tidak, maka akan halt.
- Bahasa ini ¬SD tapi komplemennya SD.