



Turing Machines (2)

Kuliah Teori Bahasa dan Automata
Program Studi Ilmu Komputer
Fasilkom UI

Prepared by:
Suryana Setiawan

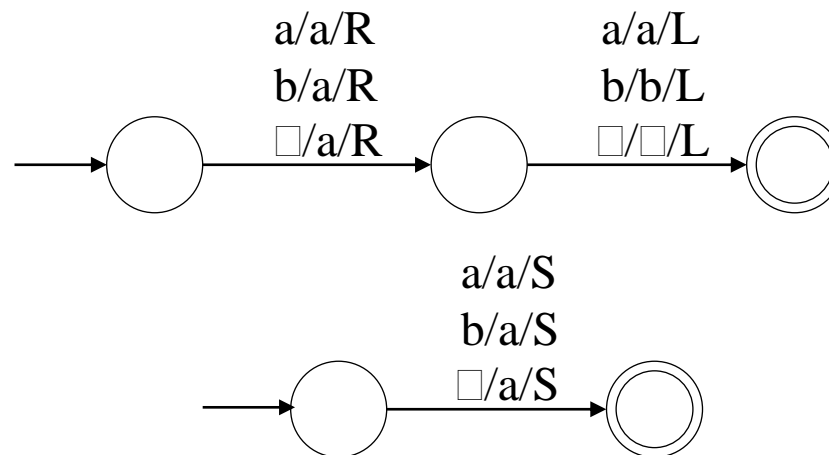


Notasi Makro

- Suatu Mesin Turing dapat dipandang sebagai susunan sejumlah Mesin Turing yang lebih sederhana.
- Mesin-mesin Turing sederhana itu bisa muncul berulang di dalam satu Mesin Turing demikian.
- Mesin Turing sederhana dengan fungsionalitas/proses yang jelas dapat digantikan suatu notasi makro guna menyederhanakan gambaran Mesin Turing besarnya.
- Dalam buku teks, satu himpunan notasi digunakan agar suatu Mesin Turing yang kompleks dapat lebih mudah digambarkan.

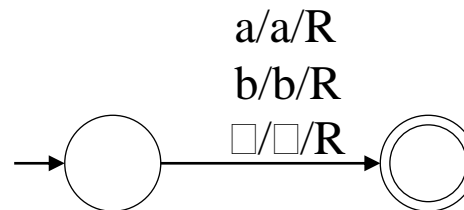
Mesin Penulis Simbol

- Untuk setiap $x \in \Gamma$, $[x]$ didefinisikan sebagai mesin yang menulis x pada posisi head, kemudian, dengan head tetap diposisi yang sama, mesin *halt*.
 - Untuk $\Gamma = \{a, b, \square\}$, akan ada mesin $[a]$, $[b]$ dan $[\square]$.
 - Misalnya $[a]$ sbb:



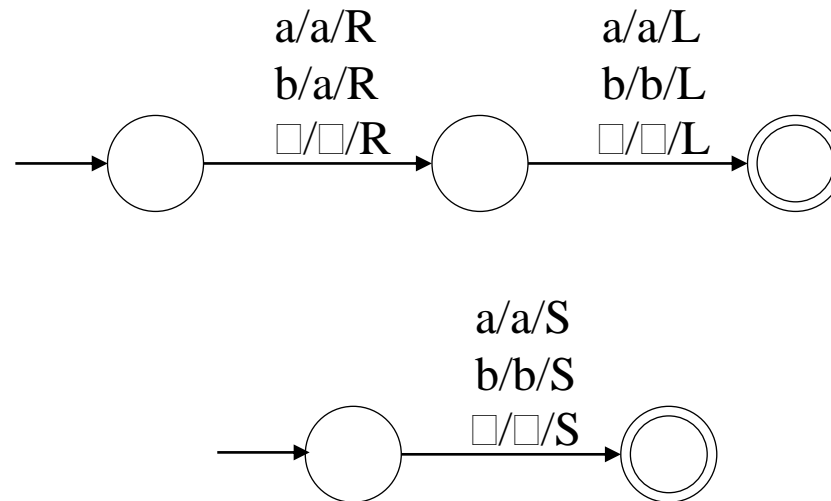
Mesin Pemindah Head

- Dibuat [R] untuk pindah ke kanan, dan [L] untuk pindah ke kiri, keduanya berpindah tanpa mengubah isi tape.
 - Untuk $\Gamma = \{a, b, \square\}$, misalnya [R]:



Mesin untuk Halt

- Ada tiga macam halt $[h]$ (halt saja), $[n]$ (halt dengan reject), dan $[y]$ (halt dengan accept).
 - Untuk $\Gamma = \{a, b, \square\}$, misalnya $[h]$:





Mesin Komposisi

- Jika mesin M terbentuk dari komposisi sikuensial antara M_1 , M_2 , ..., M_n maka dituliskan sebagai konkatenasi sbb.

$$>M_1M_2 \dots M_n$$

- Tanda mata panah “>” diberikan pada mesin yang pertama akan dijalankan.
- Contoh: $>[R][a][R][b][L][L]$ menyatakan operasi
 - berpindah ke kanan satu posisi,
 - menuliskan a,
 - berpindah ke kanan satu posisi,
 - menuliskan b, dan,
 - berpindah ke kiri dua posisi.
- Note: jika sudah jelas, tanda [dan] dapat dihilangkan.

Mesin Komposisi Kondisional

- Mesin komposisi kondisional dari M_1 dan M_2 yang digambarkan dengan panah kontinyuitas (transisi untuk versi mesin makro)
 - menjalankan M_1 , kemudian jika M_1 halt, periksa kondisi, dan jika kondisi true, maka jalankan M_2 .

$$>M_1 \xrightarrow{\text{kondisi}} M_2$$

- Selanjutnya:

$$M_1 \xrightarrow{a} M_2$$

Identik dengan

$$M_1 \xrightarrow{a/a/S} M_2$$

$$M_1 \xrightarrow{a, b} M_2$$

Identik dengan

$$M_1 \xrightarrow[b]{a} M_2$$

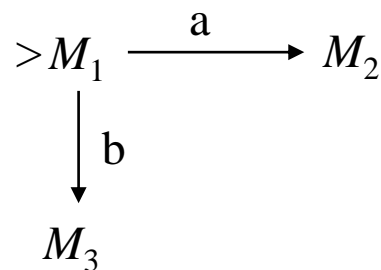
$$M_1 \xrightarrow{\neg a} M_2$$

Jika $\Gamma = \{a, b, \square\}$,
identik dengan

$$M_1 \xrightarrow{b, \square} M_2$$

Mesin Pencabangan

- Dengan komposisi kondisional, beberapa cabang dapat dibuat sesuai simbol yang dibaca head
 - Contoh untuk $\Gamma = \{a, b, \square\}$:

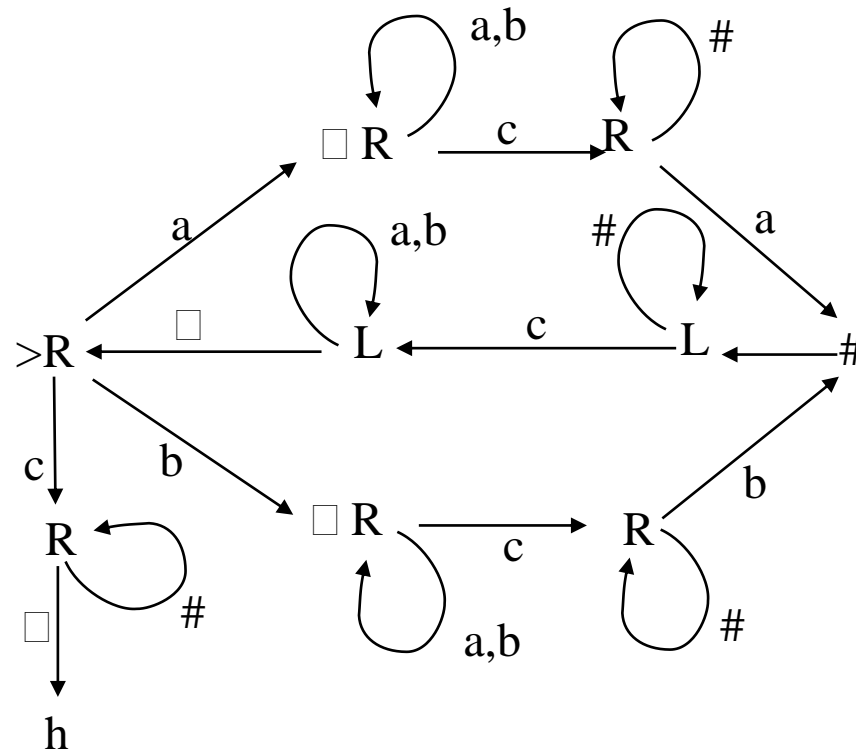


Mulai dari start state M_1 , jalankan M_1 .
Kemudian

- Jika simbol pada head a, maka jalankan M_2 .
- Jika simbol pada head b, maka jalankan M_3 .

Contoh: WcW

- Kembali dengan Bahasa WcW, dengan ide yang sama membuat mesin Turing dalam notasi makro.



- Nampaknya disini notasi makro belum begitu berbeda!

Mesin Iterator (1)

- Melakukan perulangan menjalankan suatu mesin selama kondisi true. Notasi dengan **panah self-cyclic** dengan label kondisinya atau dengan **men-subscript negasi kondisi** setelah nama mesinnya.
- Contoh:
 - Pindahkan head ke **kanan**, lalu selama disitu bukan blank, lakukan kembali.
 - Pindahkan head ke **kiri**, lalu selama disitu bukan blank, lakukan kembali.
 - Pindahkan head ke **kanan**, lalu selama disitu blank, lakukan kembali.
 - Pindahkan head ke **kiri**, lalu selama disitu blank, lakukan kembali.

$\text{>R} \curvearrowright \neg \square$ atau R_{\square}

$\text{>L} \curvearrowright \neg \square$ atau L_{\square}

$\text{>R} \curvearrowright \square$ atau $\text{R}_{\neg \square}$

$\text{>L} \curvearrowright \square$ atau $\text{L}_{\neg \square}$

Mesin Iterator (2)

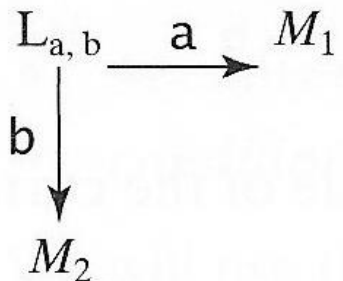
- Contoh iterator-iterator dan kombinasi dengan penjabangan.

L_a

- Temukan simbol a pertama mulai dari posisi sebelah kiri posisi head.

$R_{a,b}$

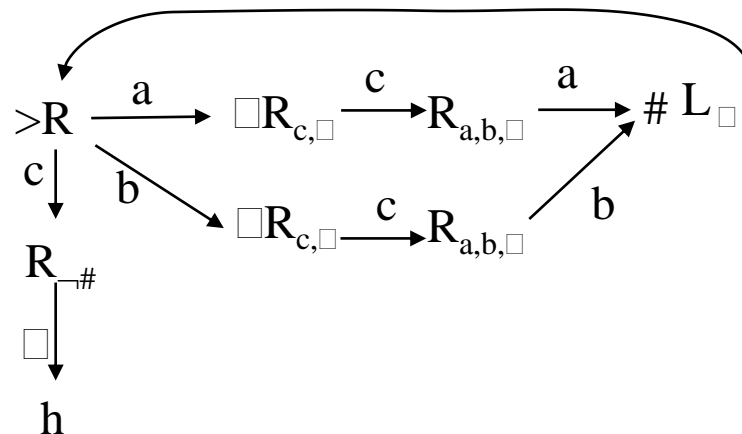
Temukan simbol a atau b pertama mulai dari posisi di sebelah kanan posisi head.



- Menemukan posisi a atau b pertama mulai dari posisi sebelah kiri posisi head, kemudian jika itu a, jalankan M_1 , jika itu b, jalankan M_2 .

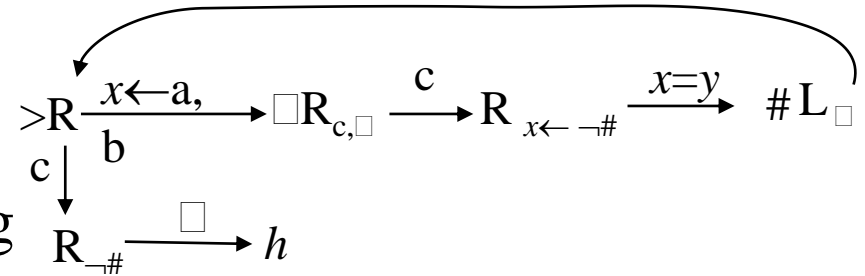
Contoh: WcW

- Kembali dengan Bahasa WcW, dengan ide yang sama membuat mesin Turing dalam notasi makro lebih lengkap (kecuali variable).

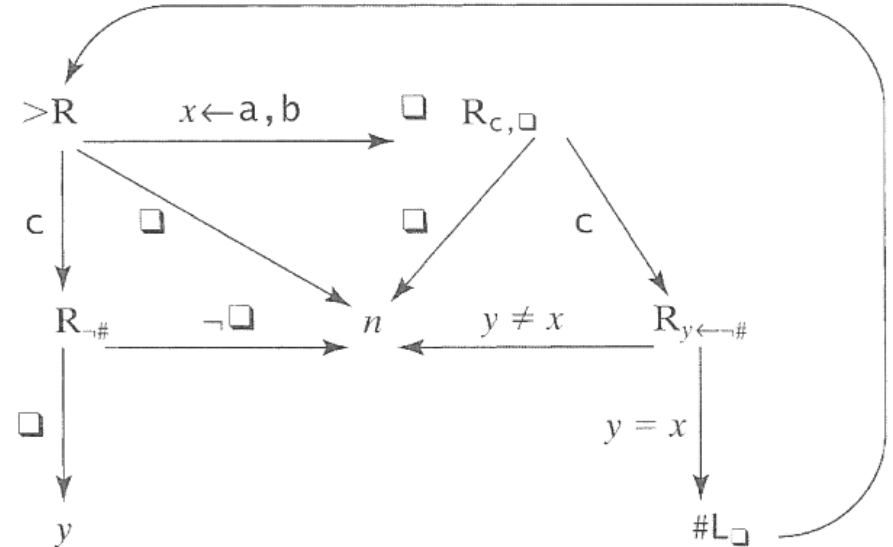


Contoh: WcW

- Bahasa WcW, dengan ide yang sama membuat mesin Turing dalam notasi makro dengan variable.



- Mesin yang kedua adalah versi apabila crash digantikan dengan halt-yes (mengacu pada contoh dalam buku teks).



Variabel Penyimpan Harga

- Simbol yang dibaca head dapat “direkam” dalam suatu “variabel” untuk digunakan kemudian.
 - Perekaman ditandai dengan tanda assignment “ \leftarrow ” setelah variabel, dan diikuti oleh simbolnya.
- Contoh 1:
 - berikut ini $M_1 \xrightarrow{x \leftarrow \neg a} M_2 \dots M_3 \xrightarrow{x = y} M_4$ menggambarkan setelah M_1 , periksa kondisi “jika bukan a” adalah true, rekam sebagai x , dan lanjutkan ke M_2 , suatu saat setelah M_3 , simbol dalam x sama dengan simbol dalam y , maka jalankan M_4 .
- Contoh lain:
- setiap simbol yang dibaca bukan blank, simpan sebagai x lalu setelah bergeser satu posisi ke kanan tuliskan x disitu.
- **Warning:** Rx berbeda dengan R_x !



Realisasi Variabel dalam Notasi TM Standard

- Dapat dilakukan dengan
 - penyimpanan di posisi tertentu pada tape, atau
 - Dengan membuat pencabangan ke status-status berbeda untuk setiap kemungkinan.
- Kendala:
 - Penyimpanan di tape menyebabkan head hilir-mudik dari posisi sebenarnya ke posisi penyimpanan lalu kembali ke posisi sebenarnya, tapi
 - Pencabangan ke status-status berbeda dapat melipatgandakan jumlah status dari mesin.



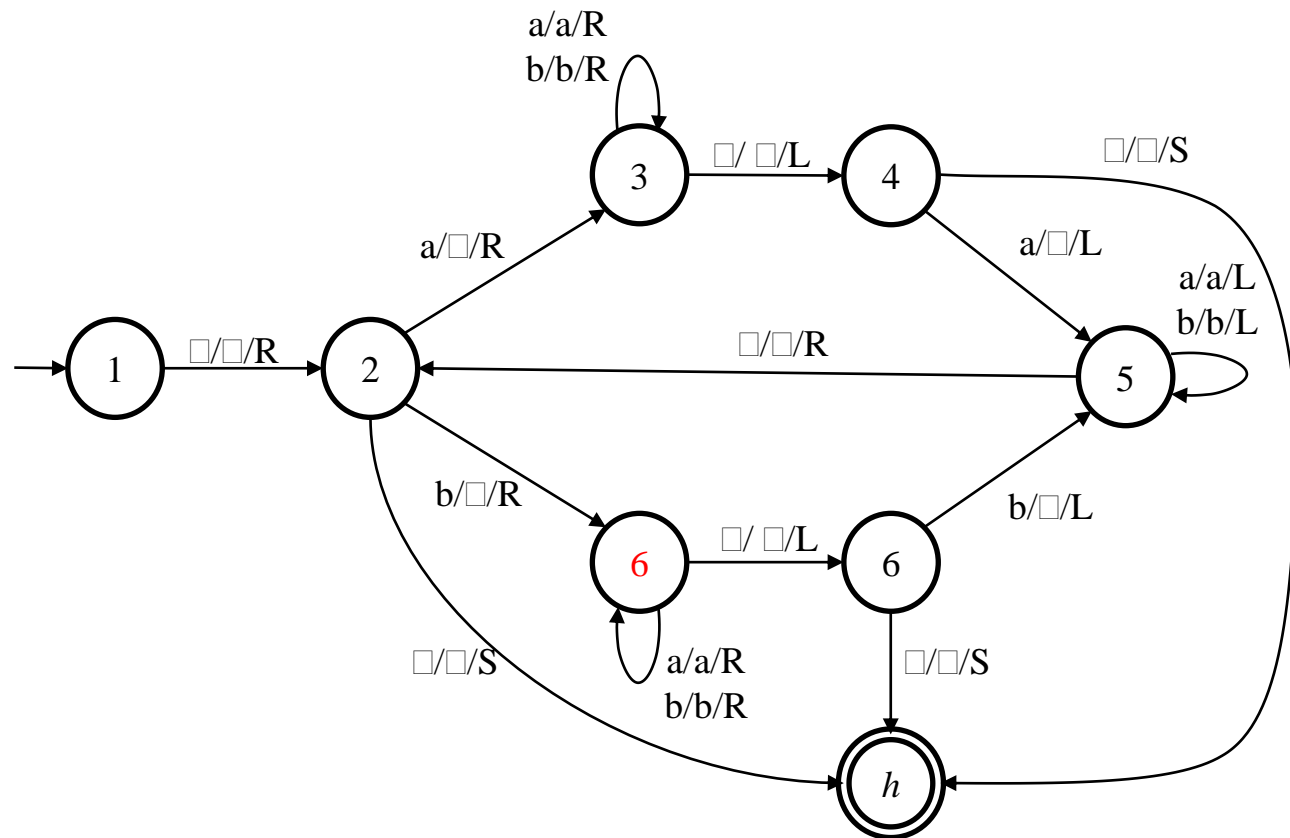
Mesin Iterator (3)

- Lebih lanjut kita bisa menuliskan iterator-iterator berikut ini.
 - Menemukan posisi a atau b pertama mulai dari posisi sebelah kiri posisi head, kemudian catat sebagai x .
 - Lalu pindahkan head ke sebelah kanannya, dan
 - Tuliskan yang dicata dalam x disitu.

$$L_{x \leftarrow a, b} R x$$

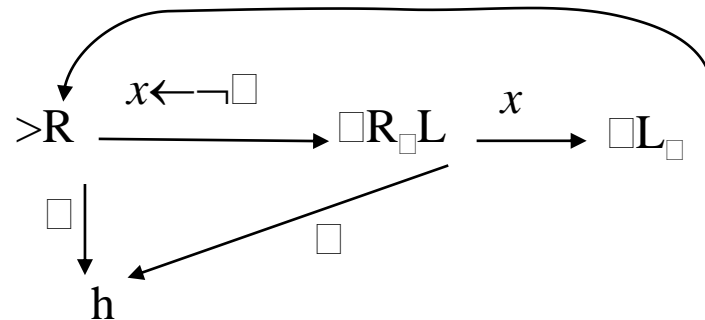
Contoh: Penggunaan Variable

- Untuk mengenali Bahasa apakah mesin berikut ini?



Contoh: Penggunaan Variable

- Mesin itu untuk menerima Bahasa Palindrom PAL dengan alfabet input $\{a,b\}$ hanya bercabang 2.
- Jika alfabet input $\{a,b,c,\dots,z\}$ seperti apa bentuknya?
- Mesin yang sama yang dibuat dengan notasi makro dengan alfabet input berukuran apapun:

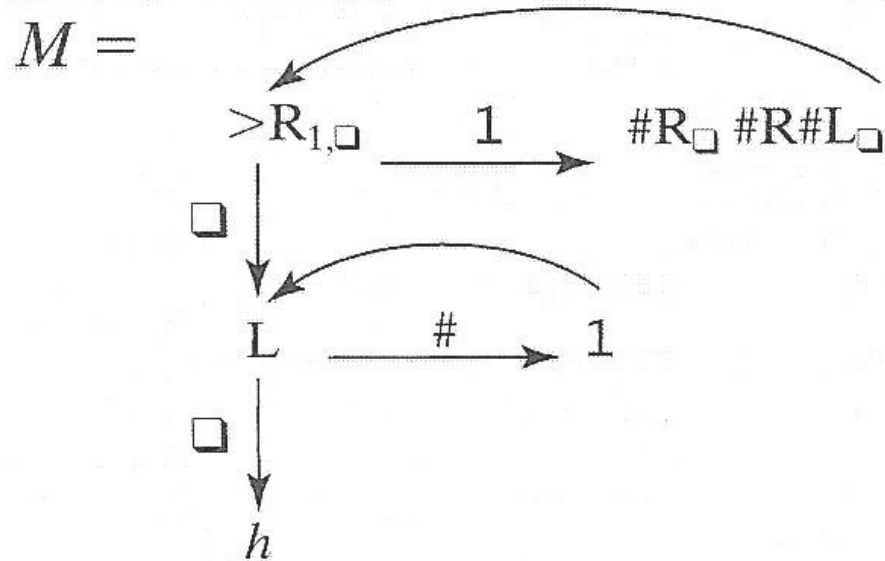




Contoh Mesin Triplikasi (1)

- Mesin untuk men-triplikasi suatu string $w \in \{1\}^*$
 - Input: $\underline{\quad}w$ Output: $\underline{\quad}www$
 - Contoh input $\underline{\quad}111$ dan output $\underline{\quad}111111111$
- Dalam loop:
 - Temukan simbol 1 atau $\underline{\quad}$ pertama arah ke kanan.
 - Jika simbol $\underline{\quad}$ (semua 1 sudah di-copy),
maka keluar dari loop.
Jika tidak (simbol 1), tandai sebagai # (untuk tidak dicopy ulang).
 - Temukan blank pertama di kanan.
 - Lalu, tuliskan dua buah # (pada dua blank berurutan).
- Lakukan dalam satu pass mengubah # kembali ke 1.

Contoh Mesin Triplikasi (2)



Contoh: Geser Satu Posisi ke Kiri

- Kita akan membangun mesin penggeser S_{\leftarrow}
- Input: $\square u \square w \square$ Output: $\square u w \square$, untuk $u, w \in \Sigma^*$.
- Contoh input $\square 11 \square 00 \square$ dan output $\square 1100 \square$ dengan $\Sigma = \{0,1\}$.
- Ide: iterasi dari kiri ke kanan dalam w , copy setiap simbol ke sebelah kirinya.

