



Turing Machines (4)

Kuliah Teori Bahasa dan Automata
Program Studi Ilmu Komputer
Fasilkom UI

Prepared by:
Suryana Setiawan

Mesin Turing Multitape

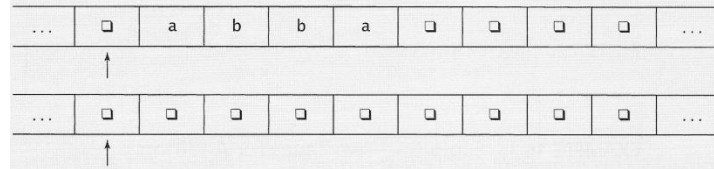
- Mesin Turing dengan k-tape (seperti halnya 1-tape) adalah 6-tuple $M = (K, \Sigma, \Gamma, \delta, s, H)$. Perbedaannya:
 - adanya tape sebanyak k buah,
 - masing-masing memiliki satu head sendiri,
 - konfigurasi mesin terdiri atas
 - status current, isi dari tape-tape tsb, dan posisi head setiap tape
 - Transisi merupakan fungsi

$$\begin{aligned} ((K-H) \times \Gamma_1 \times \dots \times \Gamma_k) \rightarrow & (K \times \Gamma_1 \times \{\rightarrow, \leftarrow, \uparrow\} \\ & \times \dots \\ & \times \Gamma_k \times \{\rightarrow, \leftarrow, \uparrow\}) \end{aligned}$$

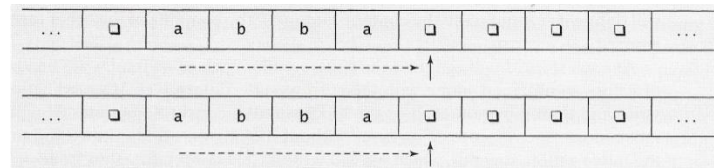
Note: untuk multi-tape terdapat pilihan gerakan head \uparrow (atau S) yang artinya tetap pada posisi tsb.

Manfaat 2-tape TM

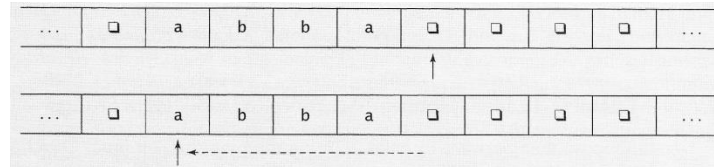
- Sebagai tempat sementara, contoh: operasi duplikasi string.
 - Di awal tape 1 berisi input string dan tape 2 kosong.



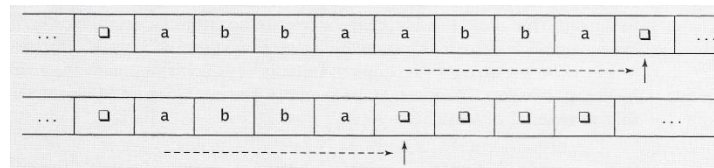
- Isi tape 1 dicopy ke tape 2, simbol demi simbol.



- Head tape 2 di “rewind” ke awal string.

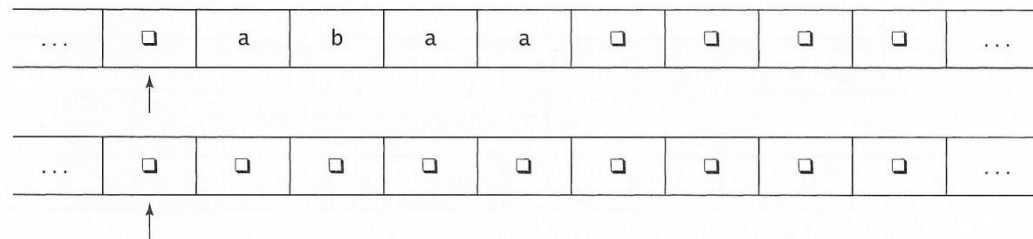


- Isi tape 2 dicopy ke tape no 1 mulai posisi current.



Teorema: Ekuivalensi TM Multitape & TM Single-tape

- Setiap TM Mutitape dapat disimulasikan oleh TM single-tape dengan ide:
 - Single tape dipandang sebagai multitrack tape dengan tambahan track-track untuk status posisi-posisi head masing-masing track.



(a)

...	□	□	a	b	a	a	□	□	□	...
		1	0	0	0	0	0	0		
		□	□	□	□	□	□			
		1	0	0	0	0	0	0		

(b)

Nondeterministik Pada Mesin Turing

- Ingat bahwa:
 - Nondeterminisme dalam FSM hanya mempermudah perancangan mesin tetapi tidak menambah “power”.
 - Nondeterminisme dalam PDA menambah “power” (terdapat bahasa yang diterima oleh NDPDA tetapi tidak dapat diterima oleh DPDA)
- Untuk TM?
 - Tidak menambah “power” (terkait penerimaan bahasa)
 - Kompleksitas DTM vs NDTM terjadi peningkatan jumlah step secara eksponensial.

NDTM Deciding dan Semideciding

- NDTM decides L , **iff**: $\forall w \in \Sigma^*$:
 - Terdapat sejumlah berhingga path yang dijalani M untuk input w ,
 - Setiap path halt (*halt* di y atau di n)
 - $w \in L$ **iff** M menerima w (*halt* di y)
- NDTM semidecides L , **iff**: $\forall w \in \Sigma^*$:
 - $w \in L$ **iff** setidaknya ada satu $(s, \sqsubseteq w) \vdash_M^* (y, \dots)$

Ekivalensi NDTM dengan DTM

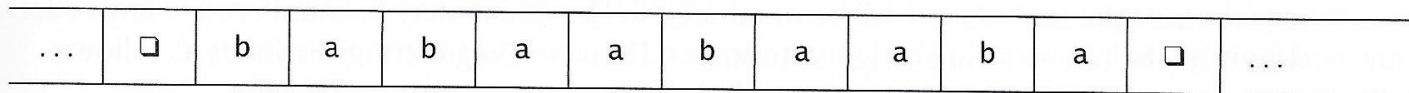
- Setiap ada NDTM M yang *decide* bahasa L maka akan terdapat juga DTM M' yang *decide* L .
- Setiap ada NDTM M yang *semidecide* bahasa L maka akan terdapat juga DTM M' yang *semidecide* L .
- Note:
 - Tentu saja ekivalensi tsb dipandang dari sudut teoritis (bahwa lingkupan bahasa NDTM dan DTM sama!)
 - Dari sudut pandang kompleksitas (misalnya jumlah step dalam komputasi!) maka perbedaannya adalah antara polinomial (nondeterministik) dengan eksponensial (deterministik).
 - Istilah NP dalam teori kompleksitas berasal dari waktu **P**olinomial untuk **N**ondeterministic TM (artinya waktu eksponensial untuk algoritma sikuensial).

Mesin Turing One-way

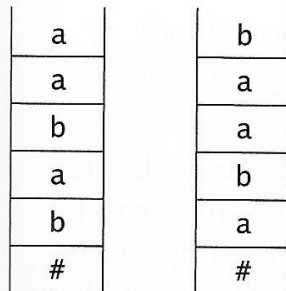
- Mesin Turing yang telah dibahas adalah two-way.
- Mesin Turing one-way memiliki tape berujung berhingga di kiri.
- Bisa terjadi situasi crash (non-halting termination) akibat head “Keluar” dari tape (yaitu ketika di posisi terkiri, head bergerak ke kirinya).
- Teorema: Setiap TM two-way dapat disimulasikan oleh TM One-way

PDA Double Stack vs TM

- Setiap TM dapat disimulasikan oleh PDA double stack.
 - Perpindahan head ke kanan disimulasikan dengan pop elemen dari stack 2 lalu push ke stack 1.



(a)



(b)

Stack 1

Stack 2

Mesin Turing Universal (UTM)

- UTM dapat dibuat untuk memproses input string yang terdiri dari sepasang string:
 - String $\langle M \rangle$: merupakan kasil pengkodean dari TM M
 - String $\langle w \rangle$: merupakan hasil pengkodean dari input string w
- Proses yang dijalankan UTM adalah mensimulasikan bekerjanya $\langle M \rangle$ saat memproses input $\langle w \rangle$.
 - Analogi: interpreter (PHP/Javascript/JVM/Perl) yang mengeksekusi script (berupa teks) dengan input data (berupa teks).
- Aturan pengkodean $\langle M \rangle$ dan $\langle w \rangle$ dapat dibuat misalnya pada slide berikutnya (spt pada textbook).

Pengkodean $M \rightarrow \langle M \rangle$

- Diberikan $M = (K, \Sigma, \Gamma, \delta, s, H)$
- Status-status dalam K dikodekan secara biner dalam $\lceil \log |K| \rceil$ digit,
 - misalnya jika $|K| = 9$, dikodekan menjadi $q0000$, $q0001$, $q0010, \dots, q1000$.
- Simbol-simbol dalam Γ dikodekan secara biner dalam $\lceil \log |\Gamma| \rceil$ digit,
 - Misalnya $\Gamma = \{ \square, a, b, c \}$, dikodekan menjadi $\square = a00$, $a = a01$, $b = a10$, $c = a11$
- Rule-rule dalam δ , setiap komponennya dikodekan dengan pengkodean status dan simbol di atas, serta satu simbol untuk arah (R atau L).

Pengkodean K dan Γ

- Encoding status-status dalam K dalam format **xBB..B** dengan BB..B representasi biner dari nomor urut status dan jumlah digit sesuai dengan $|K|$ sementara x berharga q (*non-halting state*) atau y (*accepting state*) atau n (*rejecting state*).
- Start state dibuat yang pertama (eg. q000 untuk 3 digit).
 - Contoh: $|K| = 9$, status-status adalah q000, q0001, q0010, y0011, n0100, q0101, ..., q1000 dengan q000 start state, y0011 accepting state, n0100, rejecting state.
- Encoding alfabet Γ dalam format **aBB..B** dengan BB..B representasi biner dari nomor urut simbol dan jumlah digit sesuai dengan $|\Gamma|$.
 - Contoh: $\Gamma = \{\square, a, b, c\}$, alfabet menjadi a00, a01, a10, a11

Pengkodean Arah dan δ

- Arah head selalu hanya 2: “ \rightarrow ” atau “ \leftarrow ” maka dapat menggunakan satu digit biner (untuk readability bisa tetap dengan simbol arah tsb).
- δ dapat merupakan list dari 5-tuple terpisahkan koma (untuk readability)
- Secara implisit $\langle M \rangle$ cukup diwakili pencodean δ karena anggota K dan anggota Γ dapat disimpulkan dari δ .

Contoh

Consider $M = (\{s, q, h\}, \{a, b, c\}, \{\square, a, b, c\}, \delta, s, \{h\})$, where $\delta =$

state	symbol	δ
s	\square	$(q, \square, \rightarrow)$
s	a	(s, b, \rightarrow)
s	b	(q, a, \leftarrow)
s	c	(q, b, \leftarrow)
q	\square	(s, a, \rightarrow)
q	a	(q, b, \rightarrow)
q	b	(q, b, \leftarrow)
q	c	(h, a, \leftarrow)

state/symbol	representation
s	q00
q	q01
h	q10
\square	a00
a	a01
b	a10
c	a11

$\langle M \rangle =$

$(q00, a00, q01, a00, \rightarrow), (q00, a01, q00, a10, \rightarrow),$
 $(q00, a10, q01, a01, \leftarrow), (q00, a11, q01, a10, \leftarrow),$
 $(q01, a00, q00, a01, \rightarrow), (q01, a01, q01, a10, \rightarrow),$
 $(q01, a10, q01, a10, \leftarrow), (q01, a11, q10, a01, \leftarrow)$

Pengkodean Input $\langle w \rangle$

- Jika $w = x_1x_2x_3\dots x_n$ maka $\langle w \rangle$ diperoleh sebagai string tersusun atas pengkodean masing-masing x_i mengikuti pengkodean simbol dalam Γ .
- Untuk UTM maka $\langle M \rangle$ dan $\langle w \rangle$ ditulis sebagai sebuah string $\langle M, w \rangle$ (dua string terpisahkan satu separator).