



TRANSITION GRAPH (TG)

FIRDAUS SOLIHIN

[KOMPONEN TG]

- STATE AWAL
- STATE AKHIR
- STATE TRANSISI
- INPUT STRING (€)
- TABEL TRANSISI

[TG vs FA]

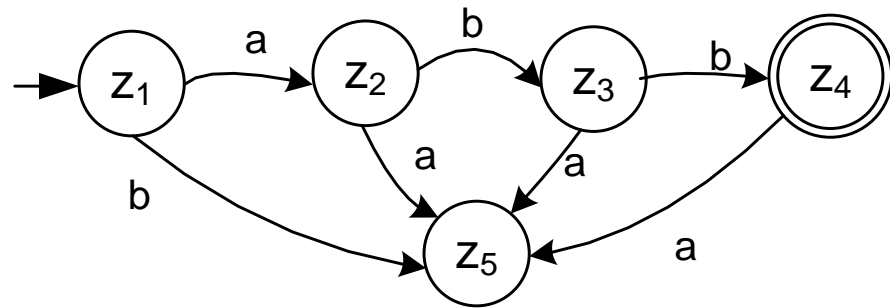
	FA	TG
State awal	1	>1
Input	Abjad Tunggal	Abjad Tunggal SubString RE String Kosong

[THEOREMA KLEENE 1]

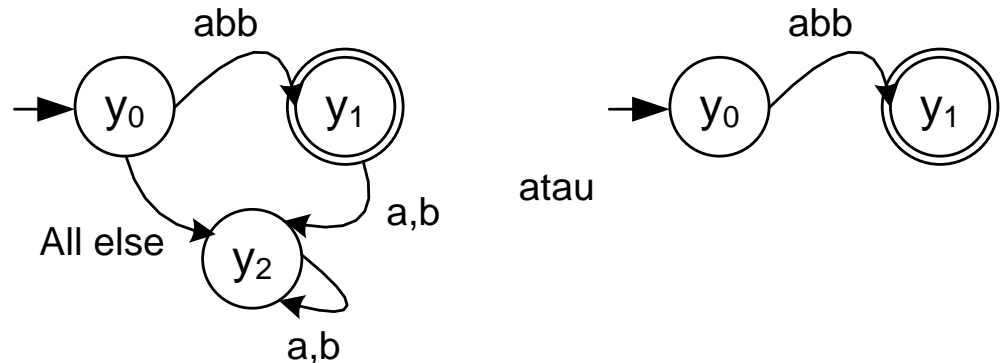
- Suatu bahasa yang didefinisikan melalui Regular Expression (RE) mempunyai bahasa ekuivalen yang digambarkan dalam bentuk Finite Automata (FA) maupun Transition Graph (TG), begitu juga sebaliknya

Contoh 1

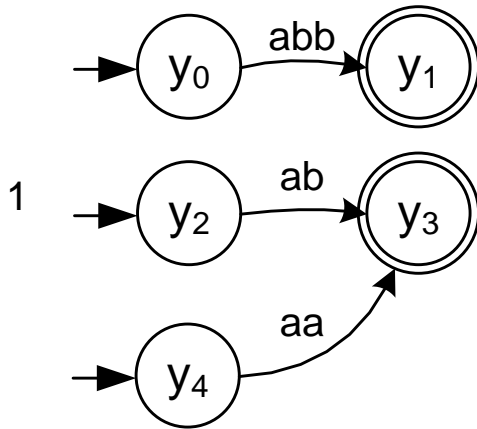
- Bahasa yang memiliki string abb saja
- RE = abb
- Gambar FA



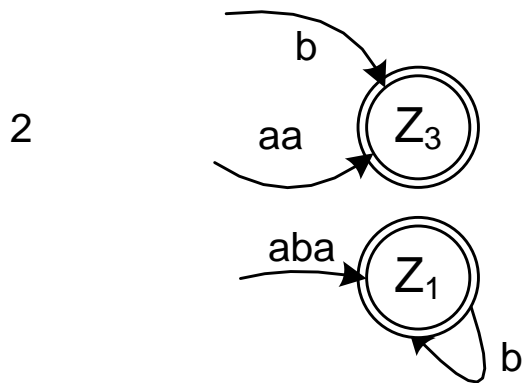
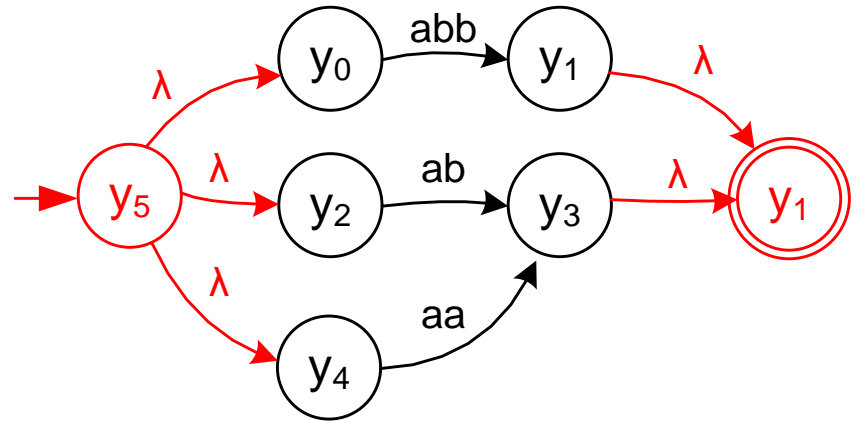
- Gambar TG



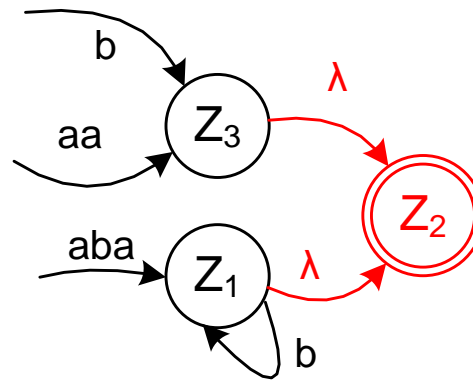
State Awal / Akhir > 1



Dapat digabung menjadi

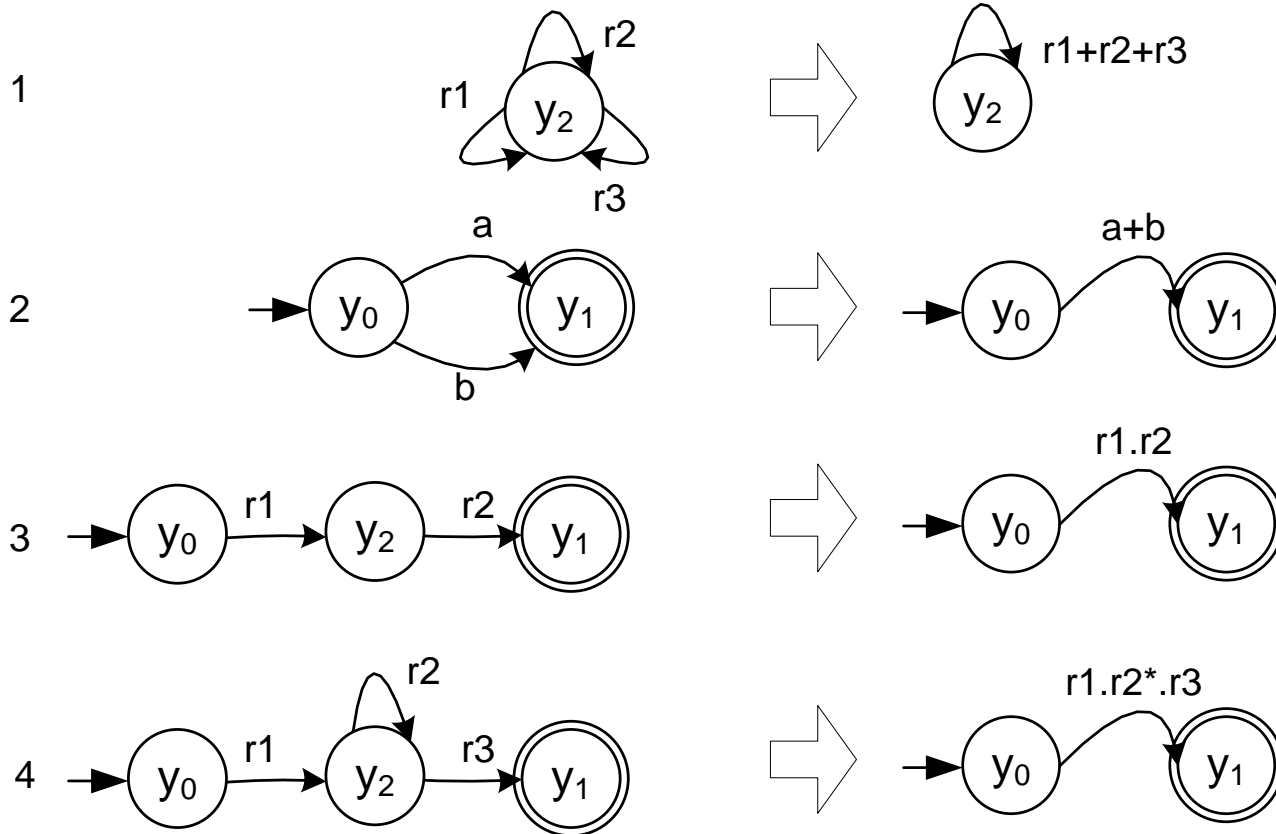


Dapat digabung menjadi

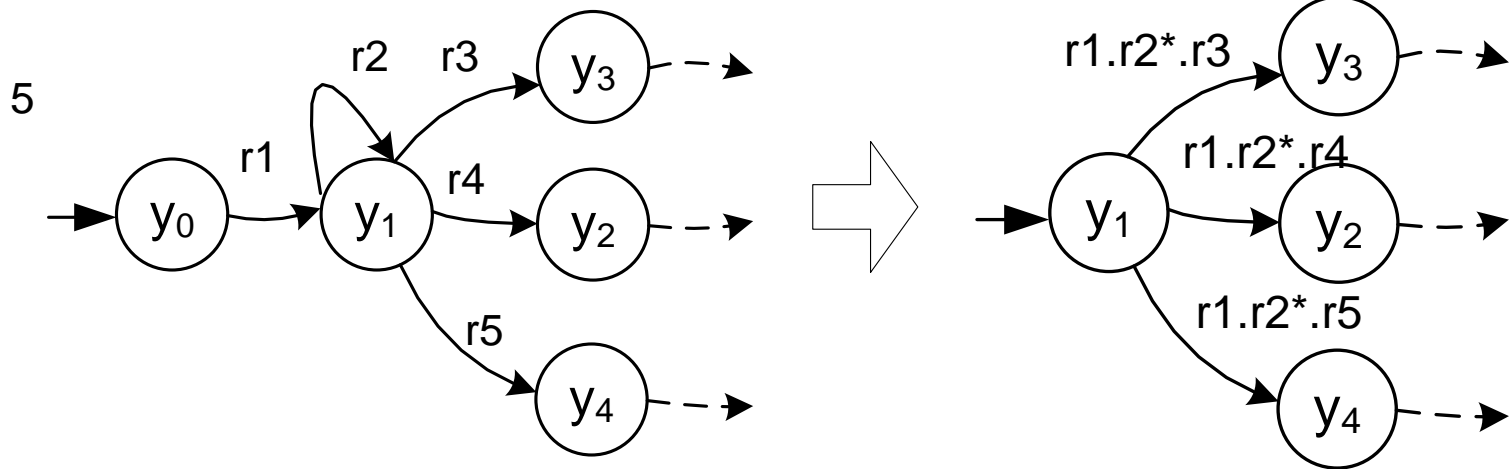




[FA \rightarrow RE dan TG]



[FA \Rightarrow TG]



[Soal]

1. Buatlah RE, FA dan TG dari bahasa yang mempunyai string yang diawali dan diakhiri dengan huruf yang berbeda
2. Buatlah RE, FA dan TG dari bahasa yang panjang stringnya selalu genap



REGULAR LANGUAGE

[BAHASA]

- REGULAR LANGUAGE
- NON REGULAR EXPRESSION

[REGULAR LANGUAGE]

- Bahasa yang dapat didefinisikan dalam bentuk Regular Expression (RE), (berdasarkan teorema kleene, Regular language dapat juga didefinisikan dalam bentuk Finite Automata (FA) maupun Transition Graph (TG))

[Operasi RL]

Jika L_1 dan L_2 adalah Regular Language maka:

- $L_1 + L_2 = \text{Regular Language}$
- $L_1 \cdot L_2 = \text{Regular Language}$
- $L_2 \cdot L_1 = \text{Regular Language}$
- $L_2' \text{ dan } L_1' = \text{Regular Language}$
- $L_1 \cap L_2 = \text{Regular Language}$

[NON REGULAR LANGUAGE]

- Bahasa yang tidak dapat didefinisikan dalam bentuk Regular Expression (RE), Finite Automata (FA) maupun Transition Graph (TG)

[Contoh]

- $L_1 = \{a^{n^2}, n= 0,1,2,3, ..\}$
- $L_2 = \{a^n b^{2n}, n= 0,1,2,3, ..\}$
- $L_3 = \{a^n b^{2n} a^{3n}, n= 0,1,2,3, ..\}$
- $L_4 = \{a^n b^n, n= \text{bil prima}, ..\}$

[Penggambaran NonRL]

- Salah satu cara untuk merumuskan Non Regular language adalah dengan menggunakan Context Free Grammar (CFG)