Non-regular Language

A powerful intuition

- Regular languages นั้นจะสอดคล้องกับปัญหาที่สามารถถูกแก้ได้ด้วยหน่วยความจำที่จำกัด
- นั่นคือ ต้องการจำของหลายอย่างแต่มีจำนวนจำกัด
- Nonregular languages จะสอดคล้องกับปัญหาที่ไม่สามารถแก้ได้ด้วยหน่วยความจำที่จำกัด
- อาจจะต้องจำของหลายอย่างแบบไม่จำกัด

What is the limit of DFA/NFA/Reg Ex?

DFA/NFA/Reg Ex ทุกตัวที่กล่าวมามีพลังเท่ากัน (ทำไม)

Recognize ภาษา set เดียวกัน

มีภาษาอะไรไหมที่ machine เหล่านี้ recognize ไม่ได้

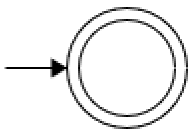
มี ตัวอย่างเช่น

$$B = \{0^n 1^n | n > 0\}$$

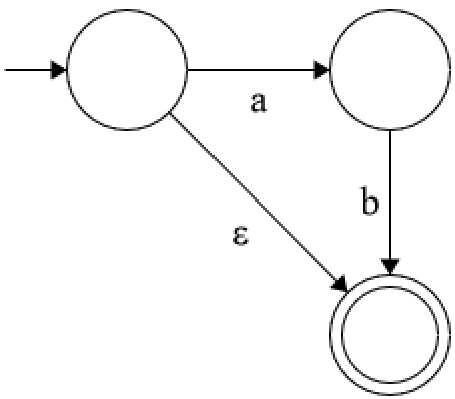
แต่เอ.. ถ้า 0 มันเยอะๆๆ จะจำยังไง ยาวเท่าไรก็ได้ แต่ M มี state จำกัด จริงหรือไม่ เรา**ไม่**สามารถบอกได้จนกระทั่งเราพิสูจน์มัน

ลองสร้าง NFA ที่ recognize ดู $B = \{0^n1^n | n > 0\}$

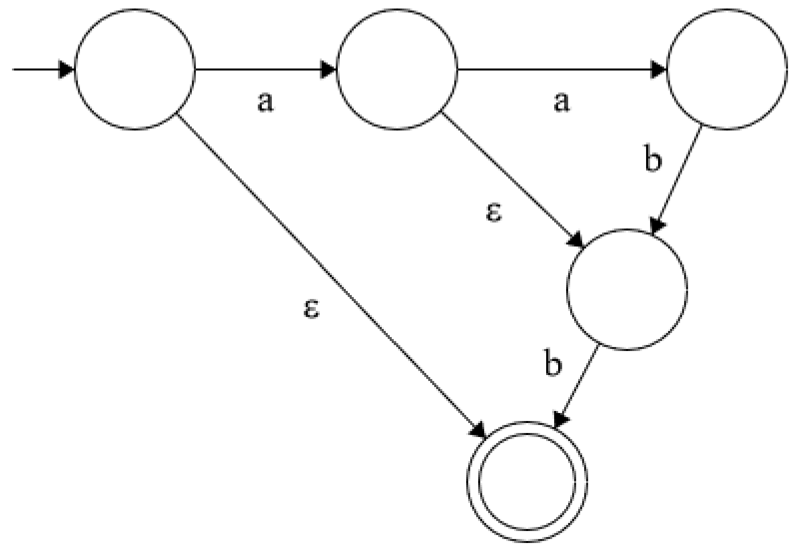
n=0



n≤1



n≤2



ในแต่ละค่า n ที่มากขึ้น เราเพิ่มมาอีก 2 state

เราใช้ state ในการนับจำนวน 0 จากนั้น จากนั้นก็ใช้จำนวนเดียวกันในการนับ 1

แต่รูปแบบนี้ไม่ประสบความสำเร็จ เนื่องจากในการสร้าง NFA สำหรับ $\{0^n1^n\}$ เมื่อ n มีขนาดไม่จำกัด

NFA นั้นมี state ที่จำกัดและ fixed ไว้ก่อน

ไม่มีจำนวนที่เพียงพอในการนับ n ที่มีจำนวนไม่จำกัด

แต่นี่ไม่ใช่การพิสูจน์

ลองพิจารณา 2 ภาษานี้ดู

$$C = \{w | w \text{ has an equal number of } 0's \text{ and } 1's\}$$

$$D = \{w | w \text{ has a equal number of occurrences of 01 and 10 as substring}\}$$

ดูเหมือนว่าทั้งสองอันต้องการการจำความยาวที่ค่อนข้างยาวเหมือนกันนี่

คำตอบ C ไม่เป็น regular

แต่ D เป็น!!

แสดงว่าที่เราคิดว่ามันจำนวนเยอะๆ แล้วต้องใช้ state เยอะจะทำไม่ได้ ไม่จริง

การที่จะบอกว่าภาษา D นี้เป็น regular ทำได้โดย

หา regular expression หรือ DFN NFA ที่ recognize มันได้

แต่ถ้าจะบอกว่าภาษา C นี้ไม่เป็นหละ

Pigeonhole principle

Pigeonhole principle

- ในสาขาคณิตศาสตร์ หลักรังนกพิราบ (pigeonhole principle) กล่าวว่าหากมีนกพิราบอยู่ n ตัว แล้วต้องการนำนกพิราบเหล่านี้ไปใส่ในรังนกพิราบ m รัง โดยที่ n > m แล้ว จะได้ว่าจะมีอย่างน้อยหนึ่งรังที่มีนกพิราบอยู่มากกว่าหนึ่งตัว

ตัวอย่าง

- ถ้าน้องโอ้มีถุงเท้าเยอะมาก มีสีแดง น้ำเงิน เหลืองและดำปนกัน ในตู้เก็บเสื้อผ้า
- น้องโอ้ต้องการหยิบถุงเท้าให้ได้สีเดียวกัน ต้องหยิบถุงเท้าครั้งละ 1 ข้าง อย่างน้อยกี่ครั้งเพื่อการันตีว่าจะได้ 1 คู่แน่ๆที่สีเดียวกัน

แสดงว่าภาษาเป็น non-regular(1)

- เราต้องการบอกว่า ภาษา $B = \{0^n1^n | n > 0\}$ เป็น non-regular

ตัวอย่าง 0011 อยู่ใน B แต่ 010 ไม่อยู่ใน B

เราจะแสดงด้วย contradiction โดยใช้ pigeonhole principle

Key idea คือ ต้องหา string ที่ยาวพอที่จะทำให้ DFA ทำซ้ำกับ state เดิม

สมมติว่า B เป็น regular

ดังนั้น B จะมี DFA $M = (Q, \Sigma, \delta, q_0, F)$ ที่ recognize B

ให้ นกพิราบเป็นทุก string ใน 0^*

รังนกเป็น state ใน Q

เราจะใส่นกพิราบ Q^i ลงในรัง $\delta^*(q_0, 0^i)$ นั่นคือ มีรังที่สอดคล้องกับ state ที่ไปถึงด้วย input 0^i

เนื่องจากตามนิยาม DFA ทำให้เรามี |Q| รัง แต่มีนกพิราบมากกว่า |Q| (จริงๆ แล้วมีได้ infinity ตัว)

ดังนั้นจาก Pigeonhole principle จะต้องมีบางรังที่มีนกพิราบอยู่ 2 ตัวขึ้นไป (วิ่งไปตาม state แล้วซ้ำนั่นเอง)

สมมติว่าเป็น 0^iและ 0^jที่ i<j

นั่นหมายความว่า 0^iและ 0^j เป็น state เดียวกัน

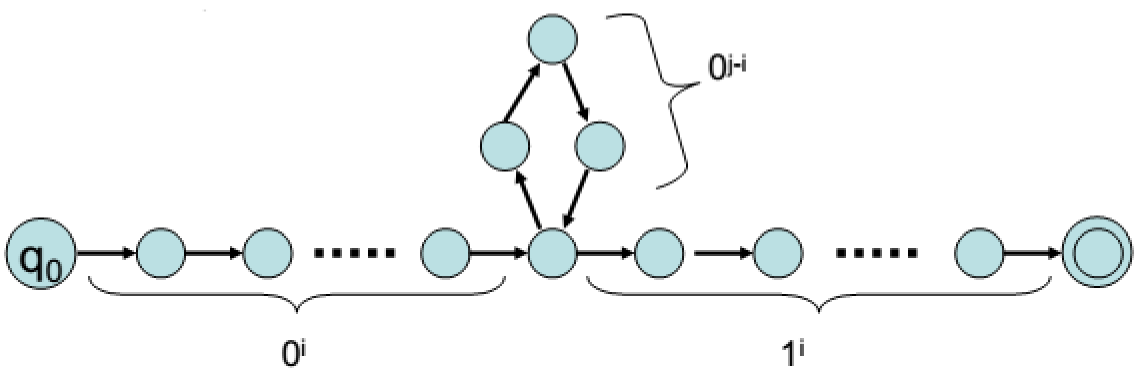
แสดงว่าว่า M รับ 0^i1^iแต่มันยังรับ 0^j1^iด้วย ซึ่งไม่ถูกต้อง

contradiction

สรุป

เราต้องการบอกว่า ภาษา $B = \{0^n1^n | n > 0\}$ เป็น nonregular

- สมมติว่า B เป็น regular
- ดังนั้น B จะมี DFA $M = (Q, \Sigma, \delta, q_0, F)$ ที่ recognize B
- พบ 0^iและ 0^j เป็น state เดียวกัน
- แสดงว่าว่า M รับ 0^i1^iแต่มันยังรับ 0^j1^iด้วย ซึ่งไม่ถูกต้อง contradiction



0^i1^i leads to the final state, so 0^j1^i does also.

แสดงว่าภาษาเป็น non-regular(2)

- ภาษา $C = \{010010001 \ldots 0^i 1 | i \text{ is any positive integer}\}$ เป็น non-regular

เราจะแสดงด้วย contradiction โดยใช้ pigeonhole principle

สมมติว่า C เป็น regular ดังนั้น C จะมี DFA $M = (Q, \Sigma, \delta, q_0, F)$ ที่ recognize C

ให้ นกพิราบเป็นทุก string ใน C

รังนกเป็น state ใน Q

นำเอาข้อความเทียบกับ state (เทียบได้กับจับนกใส่รัง) แต่ข้อความยาวกว่าจำนวน state ที่มี เนื่องจาก pigeonhole principle จะมี นกสองตัวที่อยู่ในรังเดียวกัน สมมติว่าเป็น $01\ldots0^i1$ และ $01\ldots0^j1$ ที่ i<j

นั่นคือ $01\ldots0^i1$ และ $01\ldots0^j1$ อยู่ที่ state เดียวกัน

เนื่องจาก M accept $01\ldots0^i10^{i+1}1$ ดังนั้น M จะรับ $01\ldots0^j10^{i+1}1$ ด้วย ซึ่งเกิด contradiction

แสดงว่าภาษาเป็น non-regular(3)

- ภาษา $D = \{ww | w \in \{0,1\}^*\}$ เป็น non-regular

เราจะแสดงด้วย contradiction โดยใช้ pigeonhole principle

สมมติว่า D เป็น regular ดังนั้น D จะมี DFA $M = (Q, \Sigma, \delta, q_0, F)$ ที่ recognize D

ให้ นกพิราบเป็นทุก string ในรูปของ 0^i1 เมื่อ i ไม่เป็นจำนวนเต็มลบ นั้นคือ 1,01,001

รังนกเป็น state ใน Q

นำเอาข้อความเทียบกับ state (เทียบได้กับจับนกใส่รัง) แต่ข้อความยาวกว่าจำนวน state ที่มี เนื่องจาก pigeonhole principle จะมี นกสองตัวที่อยู่ในรังเดียวกัน สมมติว่าเป็น 0^i1 และ 0^j1 ที่ i<j

นั่นคือ 0^i1 และ 0^j1 อยู่ที่ state เดียวกัน

เนื่องจาก M accept 0^i10^i1 ดังนั้น M จะรับ 0^j10^i1 ด้วยซึ่งเกิด contradiction

เครื่องมือหลักที่จะใช้: the pumping lemma

- The pumping lemma:

For any regular language, there is a string length, called the pumping length, such that for any string as long as the pumping length can be "pumped".

String ใดที่อยู่ในภาษา regular ที่ยาวกว่า pumping length มันจะ pump ได้

"pumped"– string ที่ประกอบด้วยส่วนที่สามารถทำซ้ำจำนวนเท่าไรก็ได้ ในขณะที่ผลลัพธ์ของ string ที่ได้ยังคงอยู่ในภาษา

Pumping Lemma

Theorem

If A is a regular language, then there is a number p (the **pumping length**) where, if x is any string in A of length at least p, the x maybe divided into three pieces x = uvw, satisfying the following conditions:

1. For each i $\geq$0, $uv^i w \in A$
2. |v| > 0
3. |uv| $\leq$ p

Theorem (Pumping Lemma):

– Let L be a regular language, recognized by a DFA with p states.

– Let $x \in L$ with $|x| \geq p$.

– Then x can be written as $x = u\ v\ w$ where $|v| \geq 1$, so that for all $m \geq 0$,

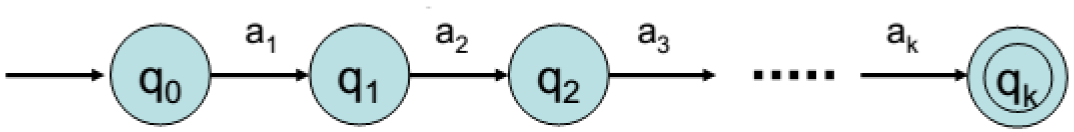
$u\ v^m\ w \in L$

พิสูจน์

สมมติว่า $x = a_1a_2a_3 \ldots a_k$ อยู่ใน L, โดยที่ $k \geq p$.

เราจะพบว่ามี $q_i = q_j$, $i < j$.

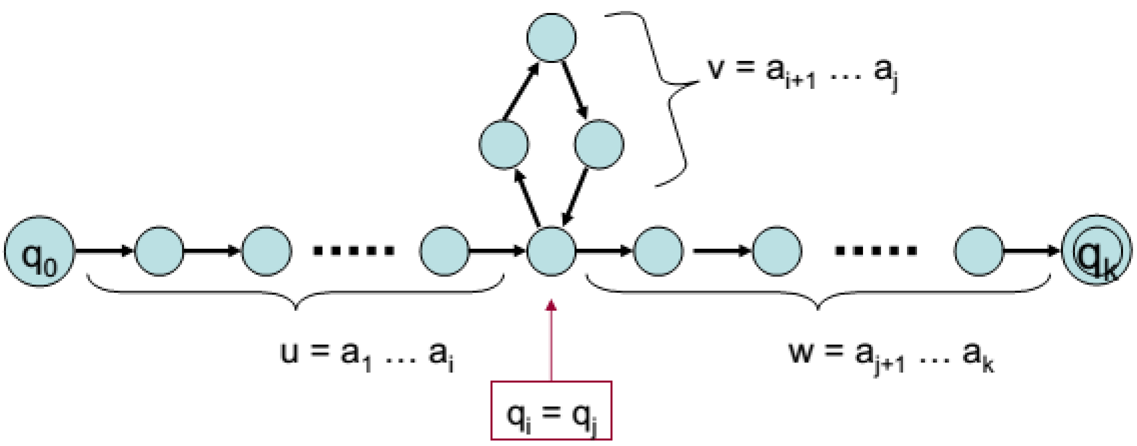
สามารถเขียนได้ว่า $u = a_1 \ldots a_i$, $v = a_{i+1} \ldots a_j$, และ $w = a_{j+1} \ldots a_k$.



loop

Claim this works:

- $x = u\ v\ w$, obviously.
- $|v| = |\ a_{i+1} \ldots a_j\ | \geq 1$, since $i < j$.
- $u\ v^m\ w$ นั้น accept เนื่องจากมันวน loop m ครั้ง (possibly 0 times).



Extra condition

– Let L be a regular language, recognized by a DFA with p states.

– Let $x \in L$ with $|x| \geq p$.

– Then x can be written as $x = u\ v\ w$ where $|v| \geq 1$, so that for all $m \geq 0$, $u\ v^m\ w \in L$

– In fact, it is possible to subdivide x in a particular way, with the total length of u and v being at most p: $|\ u\ v\ | \leq p$.

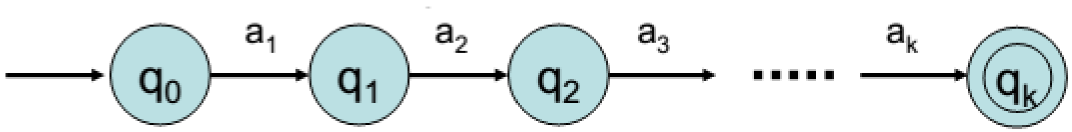
พิสูจน์

สมมติว่า $x = a_1a_2a_3 \ldots a_k$ อยู่ใน L, โดยที่ $k \geq p$.

เราจะพบว่ามี $q_i = q_j$, $i < j$. จาก pigeonhole principle

เราสามารถเลือก state 2 อันที่เกิดขึ้นระหว่าง p+1 ตัวแรกได้

ดังนั้น $|uv|<p$



พิสูจน์ว่า B ไม่เป็น regular

กำหนดให้ $B = \{0^n 1^n | n > 0\}$ ต้องการพิสูจน์ว่า B ไม่เป็น regular โดยใช้ pumping lemma

ภาษานี้คือ 0 ที่ตัวก็ได้ตามด้วย 1 ในจำนวนที่เท่ากัน

พิสูจน์ว่า B ไม่เป็น regular

กำหนดให้ $B = \{0^n 1^n | n > 0\}$ ต้องการพิสูจน์ว่า B ไม่เป็น regular โดยใช้ pumping lemma

เราจะพิสูจน์โดย contradiction

กำหนดให้ B เป็น regular จะได้ว่ามี DFA ที่ recognize B ด้วย p states (เราจะหยิบ string s ใน B จากนั้น pump มันเพื่อที่จะเกิด contradiction)

จาก pumping lemma จะได้ว่า จะมี pumping length p

พิจารณา string s = 0^p1^p เรารู้ว่า $s \in B$ และ $|s| > p$

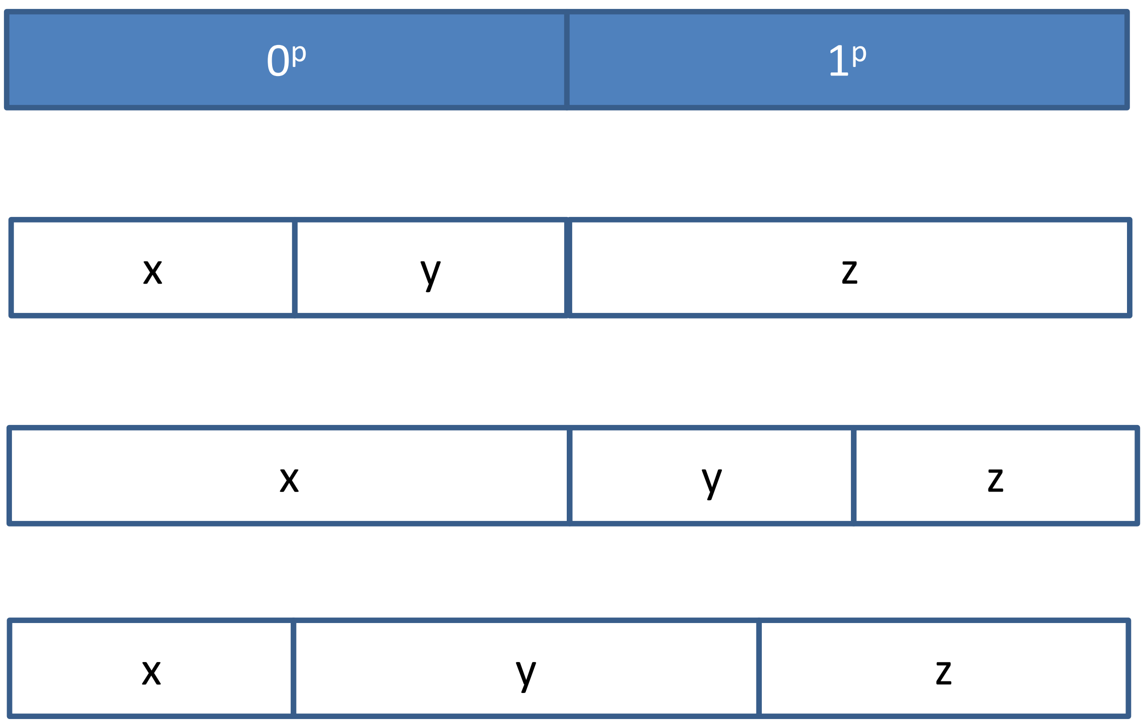
(พอมี2อย่างนี้) เราจะใช้ pumping lemma กับ s

เราสามารถแบ่ง s ได้เป็น s=xyz และสำหรับ i ใดๆ $xy^iz \in B$

พิสูจน์ว่า B ไม่เป็น regular

เรารู้ว่า $xyz \in B$ และ $xy^{i}z \in B$ (เราจะปั๊ม y)

แล้ว y เป็นอะไรได้บ้าง เราลอง y ทุกแบบ



พิสูจน์ว่า B ไม่เป็น regular

เรารู้ว่า $xyz \in B$ และ $xy^iz \in B$ (เราจะปั๊ม y)

แล้ว y เป็นอะไรได้บ้าง เราลอง y ทุกแบบ

Case 1: ถ้า $y = 0^k$ สำหรับบางค่า k>0, เราจะได้ว่า xy^2z มี 0 มากกว่า 1 ดังนั้น case นี้เป็นไปไม่ได้

Case 2: ถ้า $y = 1^k$ สำหรับบางค่า k>0, เราจะได้ว่า xy^2z มี 1 มากกว่า 0 ดังนั้น case นี้เป็นไปไม่ได้

Case 3: ถ้า $y = 0^j1^k$ สำหรับบางค่า j>0 และ k>0, สังเกตว่า case นี้ เราจะได้ว่า $xy^2z = x0^j1^k0^j1^kz \in B$ ดังนั้น case นี้เป็นไปไม่ได้

พิสูจน์ว่า B ไม่เป็น regular

ไม่ว่า case ใด เราจะได้ว่ามันเป็นไปไม่ได้ contradiction

เรา assume ว่า B เป็น regular แต่ B มี string บางตัว ซึ่ง B ไม่มี

ดังนั้น B ไม่เป็น regular

สรุปวิธีการใช้ pumping lemma

โดยทั่วไปจะทำดังนี้

- สมมติว่า ภาษาเป็น regular
- จะมี pumping length p
- **(Step นี้ยากสุดแล้ว)** หา string s ที่ยาวอย่างน้อย p ที่หลังจาก pump แล้ว string ที่ได้จะไม่อยู่ในภาษา ไม่ว่าจะปั๊มแบบไหนก็ตาม
- ได้ contradiction

$D = \{ w w \mid w \in \{ 0, 1 \}^* \}$ is non-regular.

D = { w w | w $\in$ { 0, 1 }* } is non-regular.

สมมติว่ามี DFA ที่ recognize L_3 ด้วย p states.

หยิบ string x ใน L_3 จากนั้นจะ pump จนได้ contradiction

หยิบ $x = 0^p10^p1$, โดยที่ p เป็นจำนวน states.

จาก Pumping Lemma บอกว่า x สามารถเขียนได้ในรูป u v w,

ที่ $|v| \geq 1$, แล้ว $u\ v^m\ w$ ยังอยู่ใน L_3, สำหรับทุกๆ m

แล้วทำไงต่อ

Pumping Lemma อาจจะให้ v = x, u = w = ε.

จากนั้นเพิ่ม v เท่าไรก็ได้เอาออกเท่าไรก็ได้ แล้วก็จะได้ string in L_3 ตัวอย่างเช่น ถ้า x = 001001 และ v = x แล้ว u v v w = 001001001001 ซึ่งก็อยู่ใน L_3

แย่ละ ไม่ contradiction

เราจะใช้ extra condition เพื่อที่จะทำให้ส่วนที่ทำซ้ำนั้นอยู่ใกล้ๆ ส่วนเริ่มต้น: $| u v | \leq p$.

นั้นหมายความว่า uv จะต้องมีอย่างน้อย 0s.

จากนั้น u v v w จะทำให้ได้ contradiction

เนื่องจากมันเพิ่ม 0 อย่างน้อย 1 ตัวในส่วนด้านหน้าอย่างเดียวมีผลทำให้มันมี 0 ไม่เท่ากัน

Note

สิ่งที่สำคัญมากคือการหยิบ string ที่จะ pump ให้ถูก

ตัวอย่างเช่น ถ้าเราเลือก x = 010101…, เป็นการซ้ำของ 01 เป็นจำนวนคู่ เมื่อเรา pump แล้วจะไม่ได้ contradiction.

pumping lemma จะให้ x = u v w โดยที่ v = 0101.

เนื่องจากการเพิ่ม 0101 ด้วยจำนวนอะไรก็ตาม string ที่ได้ก็ยังอยู่ใน L_3.

Exercise

$E = EQ = \{ w \in \{0,1\}^* \mid w$ contains the same number of 0s and 1s$\}$ is non-regular.