Homework 1 - 5

劉韋成 蘇俊杰

Menu

1. Hw1
   - 1
   - 2
   - 3
   - 4
   - 5
2. Hw2
   - 1
   - 2
   - 3
   - 4
   - 5
   - 6
3. Hw3
   - 1
   - 2
   - 3
   - 4
   - 5
   - 6
   - 7
   - 8
4. Hw4
   - 1
   - 2
   - 3
   - 4
   - 5
   - 6
5. Hw5
   - 1
   - 2
   - 3
   - 4
   - 5
   - 6

Hw1 Problem1

(Exercise 0.7; 30 points) For each part, give a binary relation that satisfies the condition. *Please illustrate the relation using a directed graph.*

(a) Reflexive and symmetric but not transitive

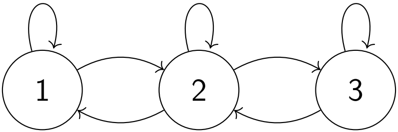
(b) Reflexive and transitive but not symmetric

(c) Symmetric and transitive but not reflexive

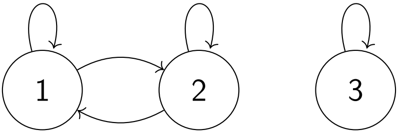
Hw1 Problem1

用有向圖來表達符合特定條件的 relation

Reflexive O Symmetric O Transitive X



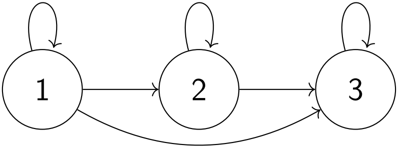
以下是錯誤範例



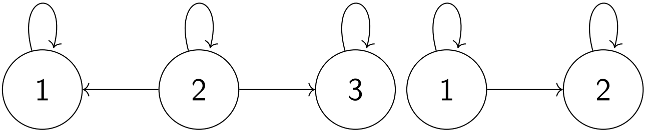
第二個 relation 並沒有違反 Transitive

Hw1 Problem1

Reflexive O Symmetric X Transitive O

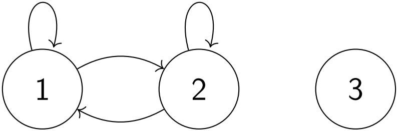


這張圖要注意箭頭方向，不要畫到最後變成大迴圈。
其他畫法：

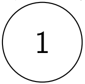


Hw1 Problem1

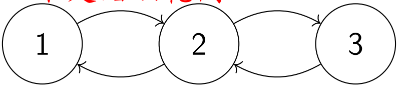
Reflexive X Symmetric O Transitive O



最省事的畫法：



以下是錯誤範例



只把自迴圈拔掉，會違反 Transitive（1 到 2，2 又到 1，那 1 應該有自迴圈）

Hw1 Problem2

(20 points) For each part, determine whether the binary relation on the set of reals or integers is an equivalence relation. If it is, please provide a proof; otherwise, please give a counterexample.

(a) The two real numbers are approximately equal.

(b) The two numbers are mapped to the same value under a fix given function.

Hw1 Problem2

給定一個 relation，判斷是不是 equivalence relation
是則證，不是則給反例

首先定義三種 relation:

Reflexive：xRx.
Symmetric：$\forall x, y\ xRy$ iff yRx.
Transitive：$\forall x, y, z\ xRy$ and $yRz \rightarrow xRz$.

Hw1 Problem (Cont.)

第一小題題目是錯的
來舉反例：
先定義近似於 $\approx$：假設兩個數字 a, b 有 $\approx$ 的話，則 $|a - b| \leq \epsilon$.
看看 Transitive：$|x - y| \leq \epsilon$ 和 $|y - z| \leq \epsilon$ 不代表 $|x - z| \leq \epsilon$

第二小題是對的：
Reflexive：$f(x) = f(x)$.
Symmetric：$f(x) = f(y)$ iff $f(y) = f(x)$.
Transitive：$f(x) = f(y)$ and $f(y) = f(z)$ implies $f(x) = f(z)$.

Hw1 Problem3

(20 points) In class, following Sipser's book, we first studied the formal definition of a function and then treated relations as special cases of functions. Please give instead a direct definition of relations and then define functions as special cases of relations. Your definitions should cover the arity of a relation or function and also the meaning of the notation $f(a) = b$.

Hw1 Problem3

重新定義 relation 並以此基礎延伸定義 function，使 function 為 relation 的特例，並且定義要涵蓋 arity 與 f(a)=b 的表示法

從頭定義可以很自由，但題目的要求全數必須達成
你可以將 relation 定義成一個 input-output 的關係，且 output 為布林值
也可以將 relation 定義成集合，裡頭包含許多的 tuples
你可以把 n-ary function 定義成一種 n+1-ary relation，也可以定義成一種 binary relation，只要寫得合理就有分

Hw1 Problem3

一個 relation R 是若干集合的 Cartesian product 的子集合
若 $R \subseteq A_1 \times ... \times A_k$，則 R 為 k-ary relation
若 $A_1 = ... = A_k = A$，則稱 R 是 (k-ary) relation on A
2-ary relation 也被稱為 binary relation

一個 function f 是滿足下列條件的一種 binary relation
$f \subseteq (A_1 \times ... \times A_k) \times B$，換句話說，在這個 relation 當中，每個 tuple 的第一項元素 t 也是一個 k-ary tuple，所有 t 都屬於同一群集合的 Cartesian product
且對於所有 t，如果 (t, a) 與 (t, b) 都屬於 function f，則 a=b
當 t 是 k-ary tuple，我們稱此 function 是 k-ary function
我們使用 f(a) = b 代表 ((a), b) 屬於 function(binary relation) f 當中
$f(a_1, ..., a_n) = b$ 則代表 $((a_1, ..., a_n), b) \in f$

Hw1 Problem4

(Problem 0.10; 20 points) Show that every graph having two or more nodes contains two nodes with the same degree. (Note: we assume that every graph is simple and finite, unless explicitly stated otherwise.)

Hw1 Problem4

對一個 simple finite graph（沒有自迴圈）而言，是否總會有兩個點他們的 degree 相同？
這題我們用反證法來證最為容易。

n 個點要分 n 種可能不重複，就只能是 0 到 n-1 各自分到一個點被分到 n-1 的點，因為這個點沒有自迴圈，所以必然被其他的 n-1 個點連到了
但這 n-1 個點包含了 degree 為 0 的點，矛盾發生了，所以一定總會有兩個點他們的 degree 相同

Hw1 Problem5

(Problem 0.11; 10 points) Find the error in the following proof that all horses are the same color.

CLAIM: In any set of h horses, all horses are the same color.

PROOF: By induction on h.

Basis ($h = 1$): In any set containing just one horse, all horses clearly are the same color.

Induction step ($h > 1$): We assume that the claim is true for $h = k$ ($k \geq 1$) and prove that it is true for $h = k + 1$. Take any set H of $k + 1$ horses. We show that all the horses in this set are the same color. Remove one horse from this set to obtain the set H_1 with just k horses. By the induction hypothesis, all the horses in H_1 are the same color. Now replace the removed horse and remove a different one to obtain the set H_2. By the same argument, all the horses in H_2 are the same color. Therefore all the horses in H must be the same color, and the proof is complete.

Hw1 Problem5

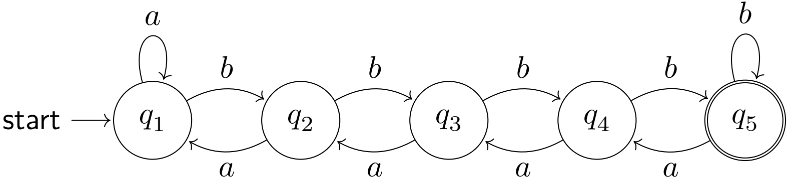
世界上的馬都變成白馬了，你能找出原因嗎？
若是三匹馬以上，H_1 與 H_2 的交集是有別的馬的
藉由 Inductive Hypothesis，這些別的馬與被移除的第一匹馬同色，
也和被移除的第二匹馬同色，所以這群馬同色
但是
在兩匹馬的時候，H_1 與 H_2 的交集是空的
從而無法真正確定兩匹馬顏色一樣
既然二就爆掉了，那麼三以上的證明也就爆了

Hw2 Problem1

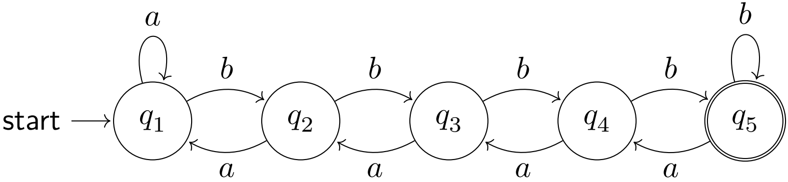
(Exercise 1.3; 10 points) The formal definition of a DFA M is $(\{q_1, q_2, q_3, q_4, q_5\}, \{\mathtt{a}, \mathtt{b}\}, \delta, q_1, \{q_5\})$ where δ is given by the following table. Draw the state diagram of M and give an intuitive characterization of the strings that M accepts.

	a	b
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

Hw2 Problem1



Hw2 Problem1



Intuitive characterization of the strings that M accepts:

Treat b as $+1$ and a as -1. We start from 0 and control the range between 0 and 4 (which means that 0 - 1 $=$ 0 and 4 $+$ 1 $=$ 4). The strings that M accepts will make the number become 4 finally.

e.q. $bbbbbab$ is an accepting string:

	$b(+1)$	$b(+1)$	$b(+1)$	$b(+1)$	$b(+1)$	a(-1)	$b(+1)$
0	1	2	3	4	4	3	4

Hw2 Problem2

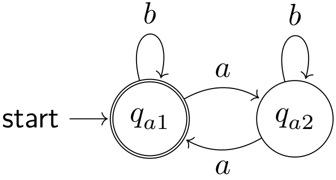
(Exercise 1.4; 20 points) Each of the following languages is the intersection of two simpler regular languages. In each part, construct DFAs for the simpler languages, then combine them using the construction discussed in class (see also Footnote 3 in Page 46 of [Sipser 2006, 2013]) to give the state diagram of a DFA for the language given. In all parts, the alphabet is $\{\mathtt{a}, \mathtt{b}\}$.

(a) $\{w \mid w$ has an even number of $\mathtt{a}$'s and one or two $\mathtt{b}$'s$\}$.

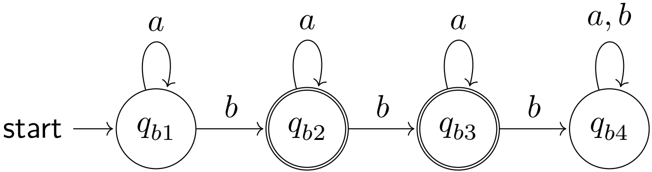
(b) $\{w \mid w$ has an odd number of $\mathtt{a}$'s and ends with a $\mathtt{b}\}$.

Hw2 Problem2 (a)

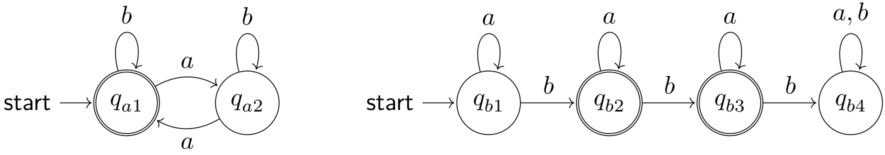
Simpler language: $\{w \mid w \text{ has an even number of } a\text{'s}\}$.



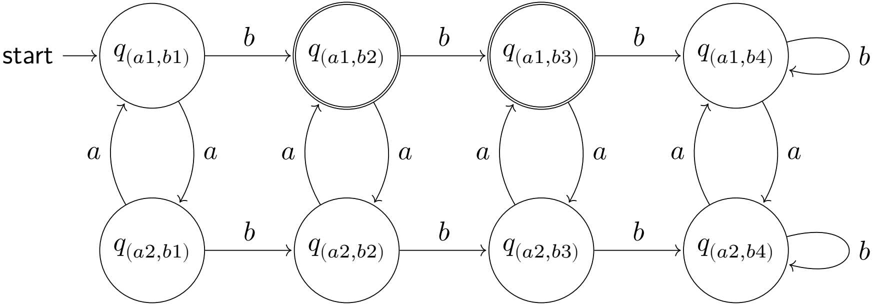
Simpler language: $\{w \mid w \text{ has one or two } b\text{'s}\}$.



Hw2 Problem2 (a)

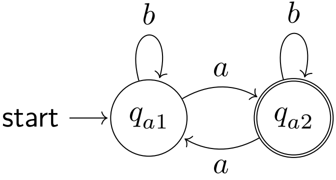


Language: $\{w \mid w$ has an even number of a's and one or two b's$\}$.

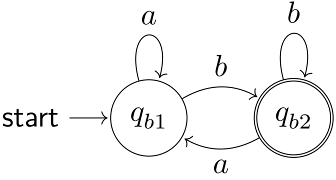


Hw2 Problem2 (b)

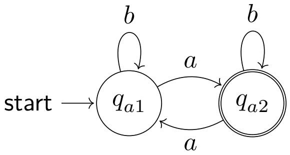
Simpler language: $\{w \mid w \text{ has an odd number of } a\text{'s}\}$.

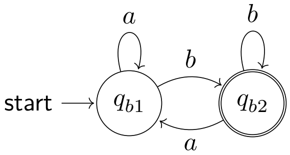


Simpler language: $\{w \mid w \text{ ends with a } b\}$.

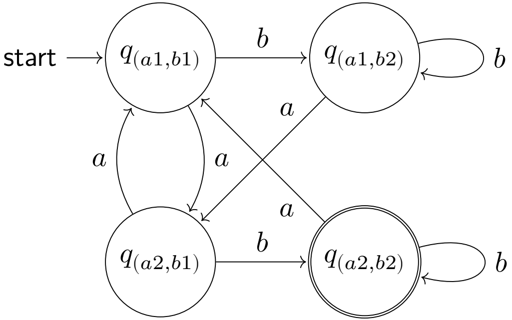


Hw2 Problem2 (b)





Language: $\{w \mid w$ has an odd number of a's and ends with a $b\}$.



Hw2 Problem3

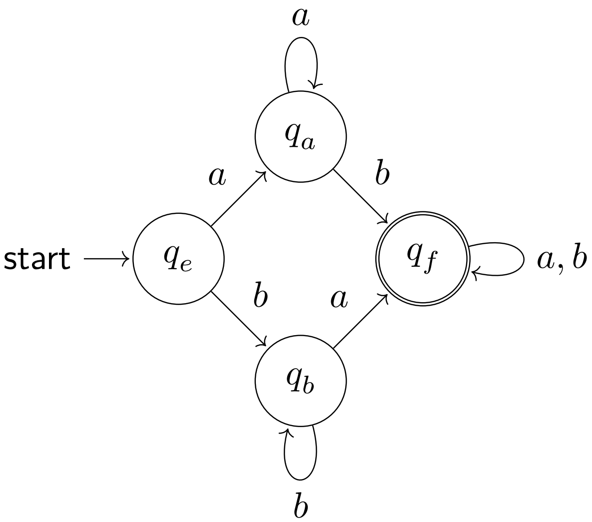
(Exercise 1.5; 20 points) Each of the following languages is the complement of a simpler regular language. In each part, construct a DFA for the simpler language, then use it to give the state diagram of a DFA for the language given. In all parts, the alphabet is $\{\mathtt{a}, \mathtt{b}\}$.

(a) $\{w \mid w$ contains neither the substring $\mathtt{ab}$ nor $\mathtt{ba}\}$.

(b) $\{w \mid w$ is any string that doesn't contain exactly two $\mathtt{a}$'s$\}$.

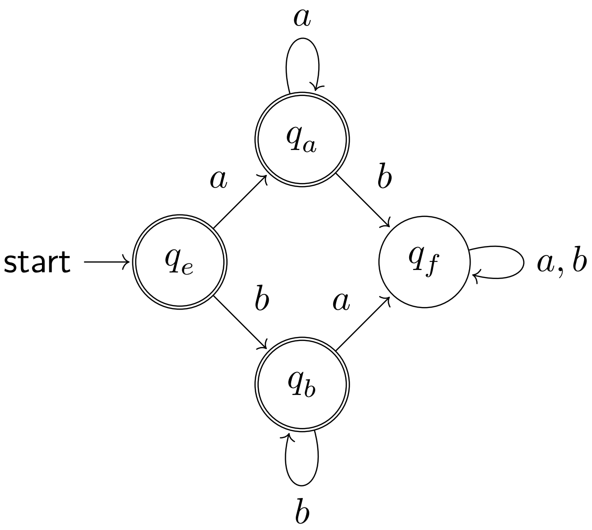
Hw2 Problem3 (a)

Simpler language: $\{w \mid w \text{ contains the substring } ab \text{ or } ba\}$.



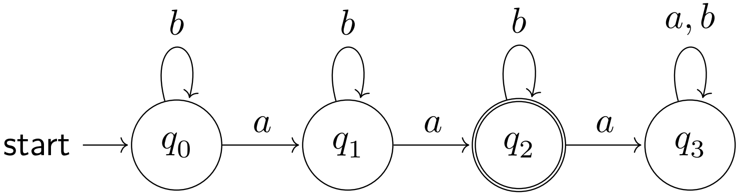
Hw2 Problem3 (a)

Language: $\{w \mid w \text{ contains neither the substring } ab \text{ nor } ba\}$.



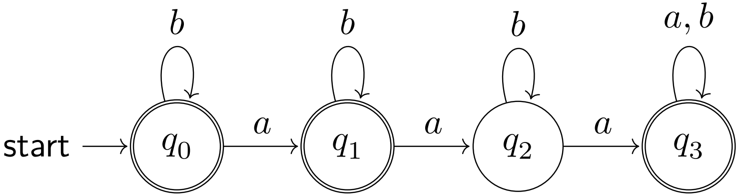
Hw2 Problem3 (b)

Simpler language:
$\{w \mid w$ is any string that contains exactly two a's$\}$.



Hw2 Problem3 (b)

Simpler language:
$\{w \mid w$ is any string that doesn't contain exactly two a's$\}$.



Hw2 Problem4

(Problem 1.36; 10 points) For any string $w = w_1 w_2 \cdots w_n$, the *reverse* of w, written w^R, is the string w in reverse order, $w_n \cdots w_2 w_1$. For any language A, let $A^R = \{w^R \mid w \in A\}$. Show that if A is regular, so is A^R.

Hw2 Problem4

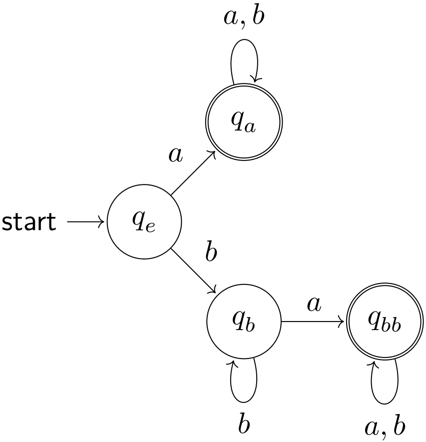
Let DFA M recognizes the language A, we can construct an DFA M^R that recognizes A^R according to the following:

- M^R's states and alphabet are as same as M.
- Reverse all the translations of M.
  e.q. $\delta(q_1, a) = q_2 \rightarrow \delta(q_2, a) = q_1$.
- Turn M's initial state into accepting state.
- Turn M's accepting state into initial state. But we will obtain more than one initial states here, so we start from each initial state and find all the combinations of simple paths and cycles starting from this initial state to the accepting state.
- Because the combinations are limited, we can draw a DFA for each combination with limited numbers. Finally, unionize all DFA and we'll get M^R.

Because M^R recognizes A^R, A^R is regular.

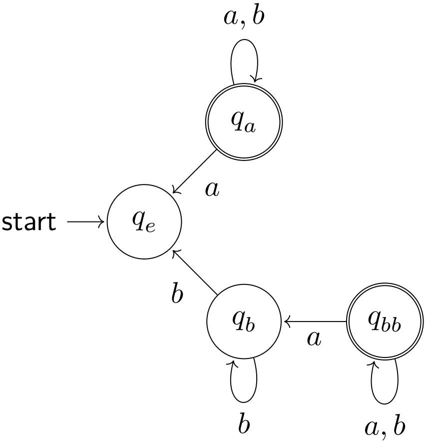
Hw2 Problem4

e.q. : M



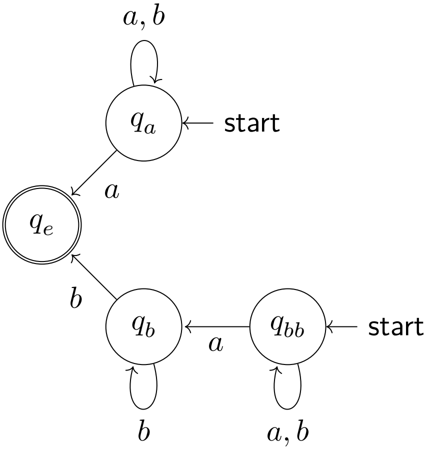
Hw2 Problem4

Reverse all the translations:



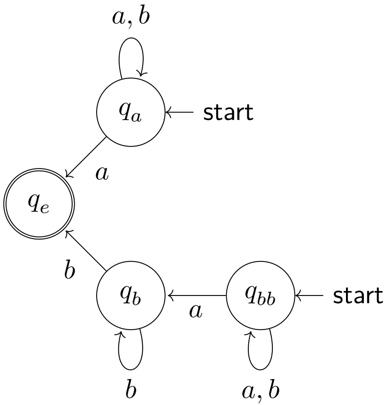
Hw2 Problem4

Exchange the initial state and accepting states:



Hw2 Problem4

Find all the combinations of simple paths and cycles:



Start from q_a: $(a|b)^*a$
Start from q_{bb}: $(a|b)^*ab^*b$
Draw DFA for them and unionize them to obtain M^R.

Hw2 Problem4

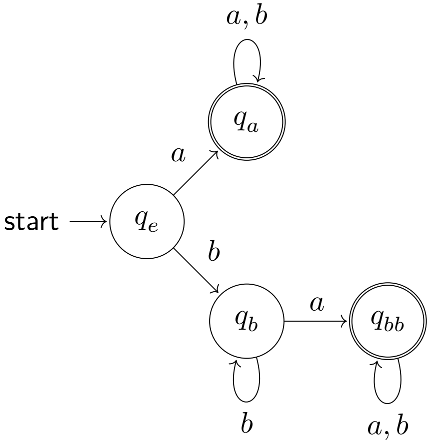
Let NFA M recognizes the language A, we can construct an NFA M^R that recognizes A^R according to the following:

- M^R's states and alphabet are as same as M.
- Reverse all the translations of M as the translations of M^R. e.q. $\delta(q_1, a) = q_2 \to \delta(q_2, a) = q_1$.
- The accepting state of M^R is M's initial state.
- Add an additional initial state q_0 to M^R. Construct the translations from q_0 to all the accepting states of M with the label ϵ.

Because M^R recognizes A^R, A^R is regular.

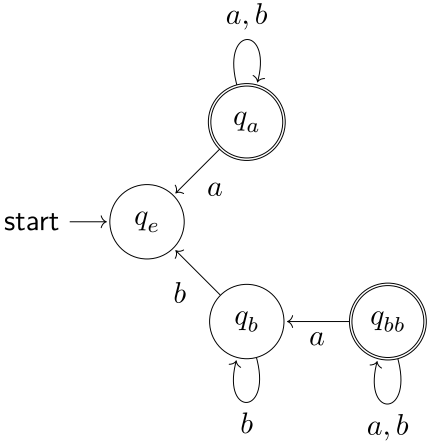
Hw2 Problem4

e.q. : M



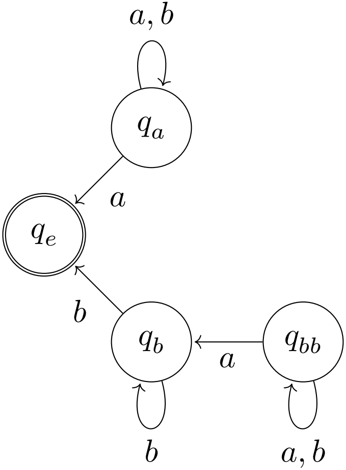
Hw2 Problem4

Reverse all the translations:



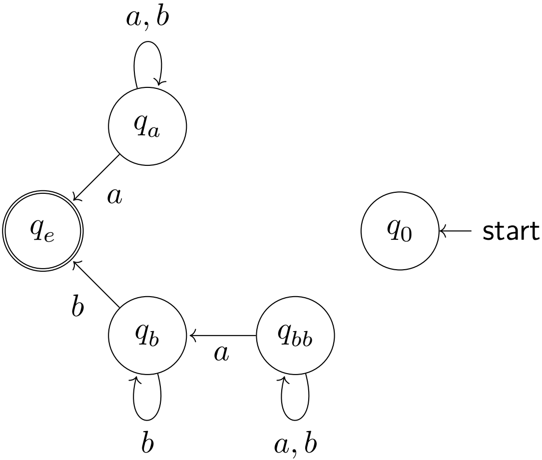
Hw2 Problem4

Change the initial state into accepting state:



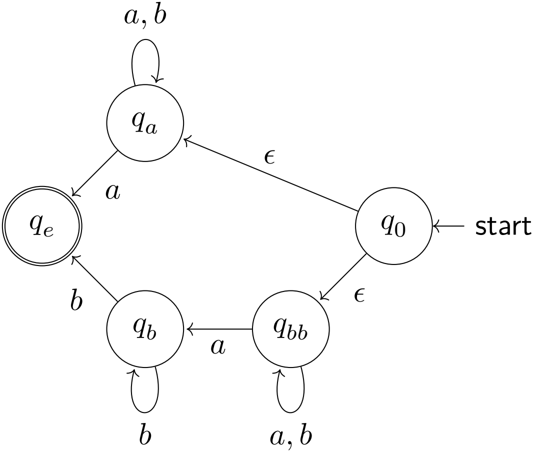
Hw2 Problem4

Add an additional initial state q_0:



Hw2 Problem4

Construct the translations from q_0 to all the accepting states of M with the label ϵ, then we can get the NFA M^R:



Hw2 Problem5

(Problem 1.37; 20 points) Let

$$\Sigma_3 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right\}.$$

Σ_3 contains all size 3 columns of 0s and 1s. A string of symbols in Σ_3 gives three rows of 0s and 1s. Consider each row to be a binary number and let

$$B = \{w \in \Sigma_3^* \mid \text{the bottom row of } w \text{ is the sum of the top two rows}\}.$$

For example,

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \in B, \text{but} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \notin B.$$

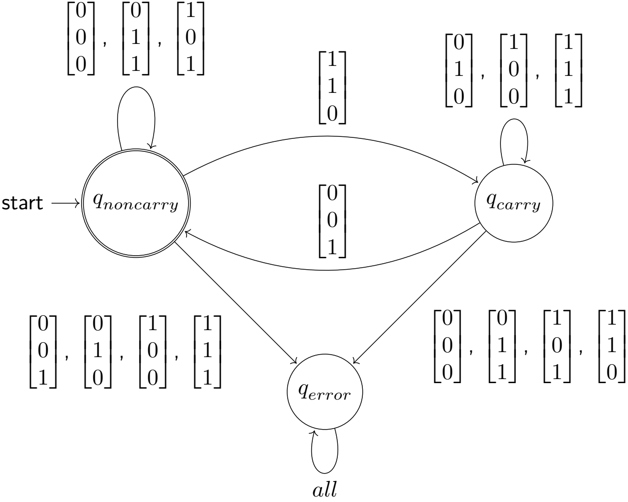
Show that B is regular. (Hint: working with B^R is easier. You may assume the result claimed in the previous problem (Problem 1.36).)

Hw2 Problem5

Consider the situation of carry, starting from the tail of B is easier than starting from the head. So we first show that B^R is regular. We can construct a DFA that recognizes B^R when considering the carry and the correctness of calculation.

Hw2 Problem5

The DFA that recognizes B^R:



Hw2 Problem5

Because there is a DFA that recognizes B^R, B^R is regular. According to the result claimed in problem 4 (if A is regular, so is A^R), we can say that $(B^R)^R = B$ is regular.

Hw2 Problem6

(20 points) Generalize the proof of Theorem 1.25 of [Sipser 2006, 2013] (Pages 45–47) to handle A_1 and A_2 with different alphabets.

Hw2 Problem6

Suppose $M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$ recognizes A_1 and $M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$ recognizes A_2.
Construct $M = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1 \cup A_2$:

- $Q = \{(r_1, r_2) | r_1 \in (Q_1 \cup \{q_f\})$ and $r_2 \in (Q_2 \cup \{q_f\})\}$.
- $\Sigma = \Sigma_1 \cup \Sigma_2$.
- $\delta((r_1, r_2), a) =$
$$\begin{cases} (\delta_1(r_1, a), \delta_2(r_2, a)) & \text{if } r_1, r_2 \neq q_f \wedge a \in (\Sigma_1 \cap \Sigma_2) \\ (\delta_1(r_1, a), q_f) & \text{if } r_1 \neq q_f \wedge a \in \Sigma_1 \wedge (r_2 = q_f \vee a \notin \Sigma_2) \\ (q_f, \delta_2(r_2, a)) & \text{if } r_2 \neq q_f \wedge a \in \Sigma_2 \wedge (r_1 = q_f \vee a \notin \Sigma_1) \\ (q_f, q_f) & \text{if } (r_1 = q_f \vee a \notin \Sigma_1) \wedge (r_2 = q_f \vee a \notin \Sigma_2) \end{cases}$$
- $q_0 = (q_1, q_2)$.
- $F = \{(r_1, r_2) | r_1 \in F_1$ or $r_2 \in F_2\}$.

Hw2 Problem6

Why we need q_f?

Because when we read a character a that in Σ_1 but not in Σ_2, A_2 cannot recognize a so M_2 must fail and never accept. If there's no q_f, M cannot find out this situation.

Hw3 Problem1

(Exercise 1.7; 10 points) For each of the following languages, give the state diagram of an NFA, with the specified number of states, that recognizes the language. In all parts, the alphabet is $\{0, 1\}$.

(a) The language $\{w \mid w$ contains the substring 1101, i.e., $w = x1101y$ for some x and $y\}$ with five states

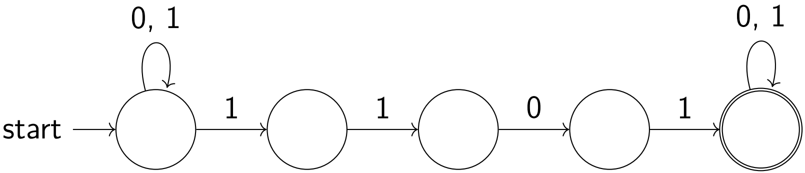
(b) The language $1^+0^*1^*$ with three states

Hw3 Problem1

請用題目要求的狀態數量畫出指定的 NFA

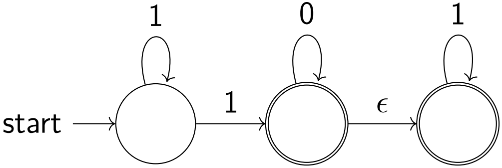
Hw3 Problem1

第一小題：一個包含子字串 1101 的字串，5 個狀態



Hw3 Problem1

第二小題：$1^+0^*1^*$，3 個狀態



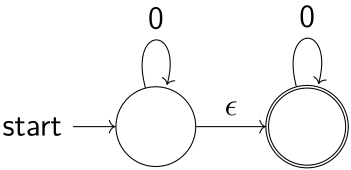
Hw3 Problem2

(Exercise 1.14; 10 points) Show by giving an example that, if M is an NFA that recognizes language C, swapping the accept and nonaccept states in M doesn't necessarily yield a new NFA that recognizes the complement of C. Is the class of languages recognized by NFAs closed under complement? Explain you answer.

Hw3 Problem2

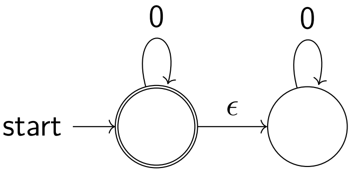
這題有兩個部分
給出一個例子說明把 NFA 的 accepting 調換過來不會產生 complement
以及 NFA 辨識的語言的 complement 是否能被另外一個 NFA 辨識
也就是「能被 NFA 辨識」這件事 closed under complement

Hw3 Problem2



接受 0^*

Hw3 Problem2



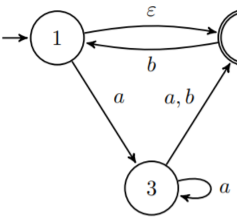
還是接受 0^*

Hw3 Problem2

NFA 可以轉換成等價的 DFA
已知 DFA 的 accepting 調換之後就能辨識其語言的 complement
又 DFA 就是一種 NFA，所以是 closed under complement

Hw3 Problem3

(Exercise 1.16; 20 points) Use the construction given in Theorem 1.39 (every NFA has an equivalent DFA) to convert the following NFA into an equivalent DFA.

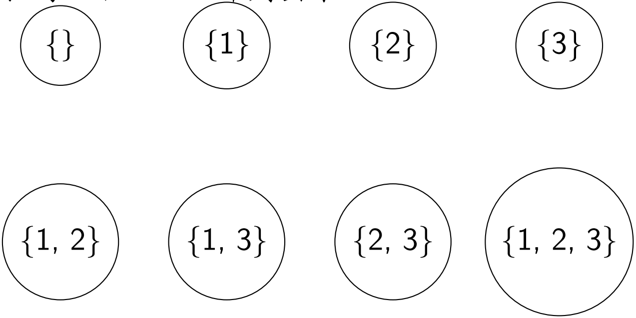


Hw3 Problem3

利用 Th 1.39 的方法（subset construction）建構出等價的 DFA

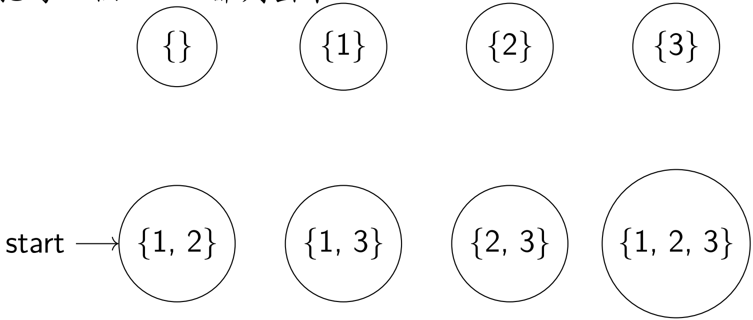
Hw3 Problem3

把每一個 state 都列出來



Hw3 Problem3

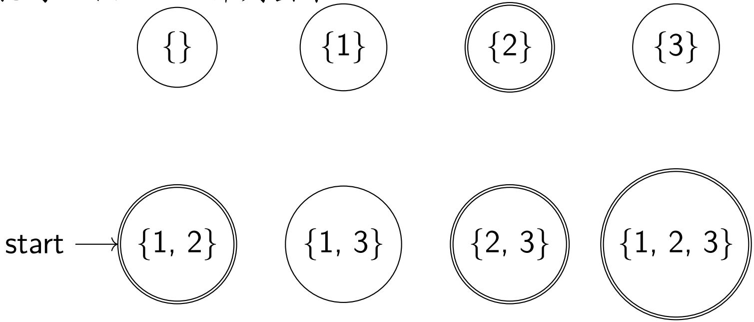
把每一個 state 都列出來



$q'_0 = E(\{1\}) = \{1, 2\}$

Hw3 Problem3

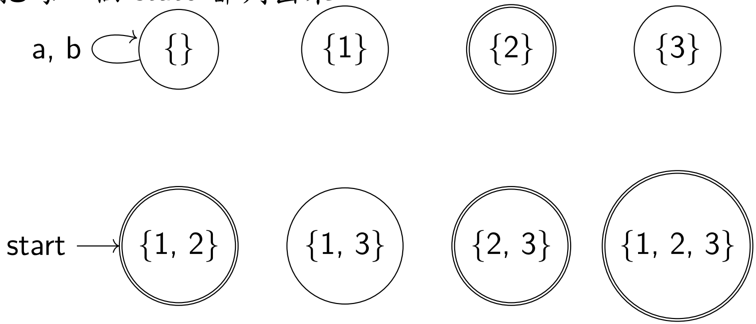
把每一個 state 都列出來



$F' = \{\{2\}, \{1, 2\}, \{2, 3\}, \{1, 2, 3\}\}$

Hw3 Problem3

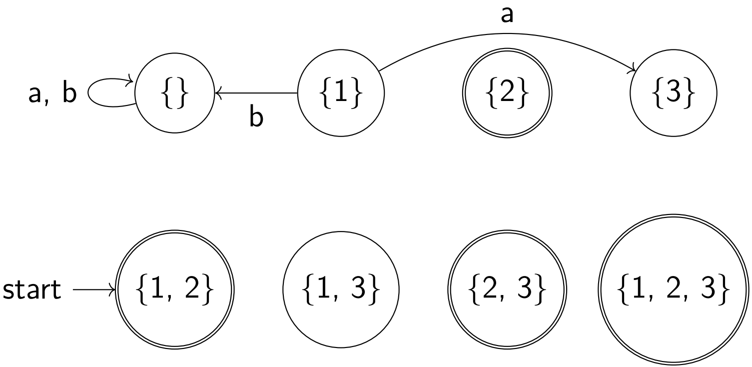
把每一個 state 都列出來



$\delta'(\{\}, a) = \{\}$
$\delta'(\{\}, b) = \{\}$

Hw3 Problem3

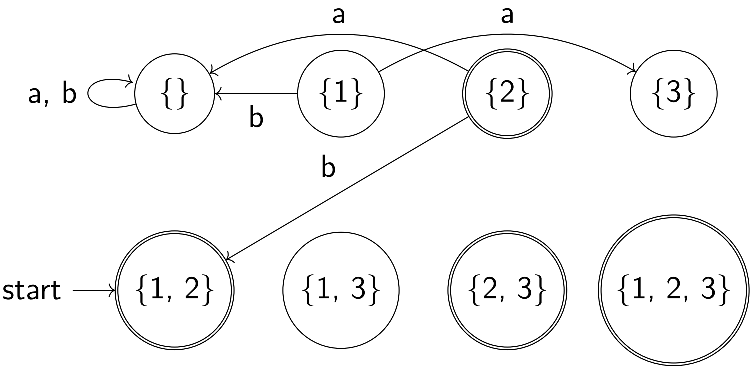
把每一個 state 都列出來



$\delta'(\{1\}, a) = \{3\}$
$\delta'(\{1\}, b) = \{\}$
注意我們並沒有把 {1} 偷偷展開成 {1, 2}

Hw3 Problem3

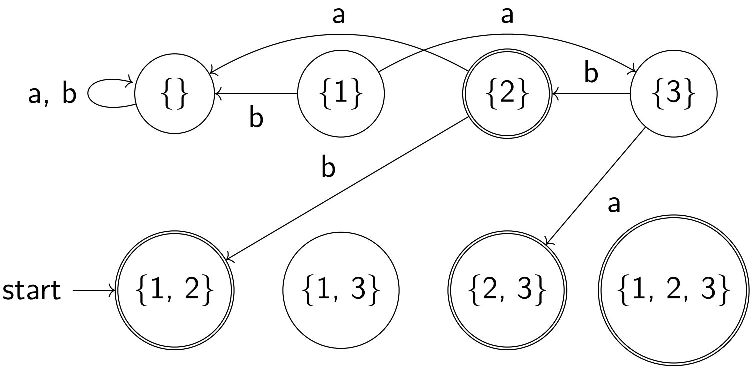
把每一個 state 都列出來



$\delta'(\{2\}, a) = \{\}$
$\delta'(\{2\}, b) = \{1, 2\}$

Hw3 Problem3

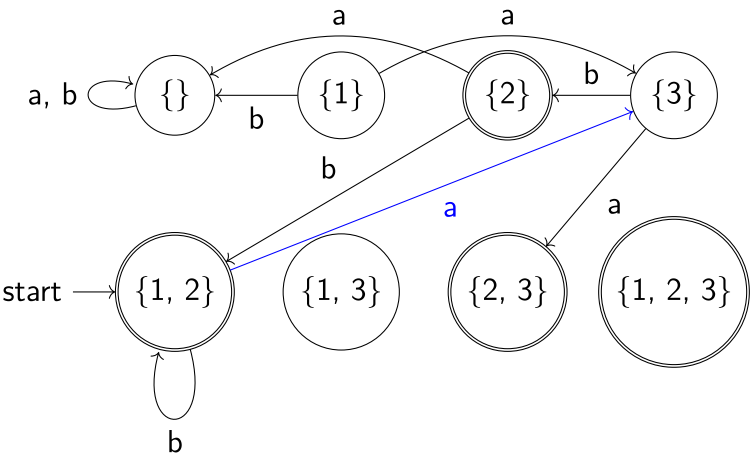
把每一個 state 都列出來



$\delta'(\{3\}, a) = \{2, 3\}$
$\delta'(\{3\}, b) = \{2\}$

Hw3 Problem3

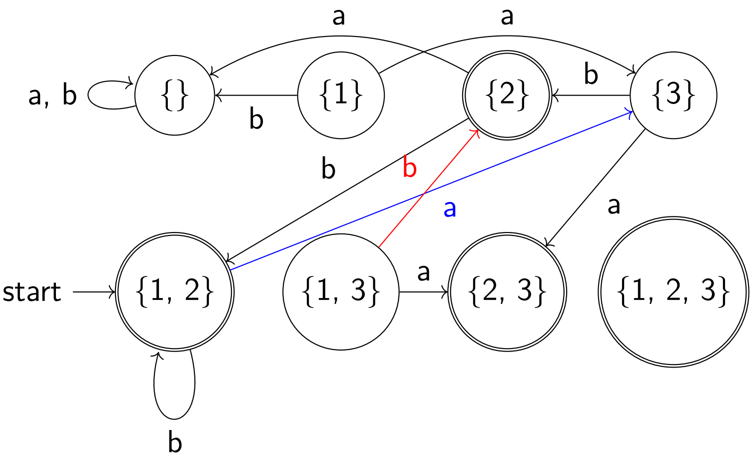
把每一個 state 都列出來



$\delta'(\{1,2\},a) = \{3\}$
$\delta'(\{1,2\},b) = \{1,2\}$

Hw3 Problem3

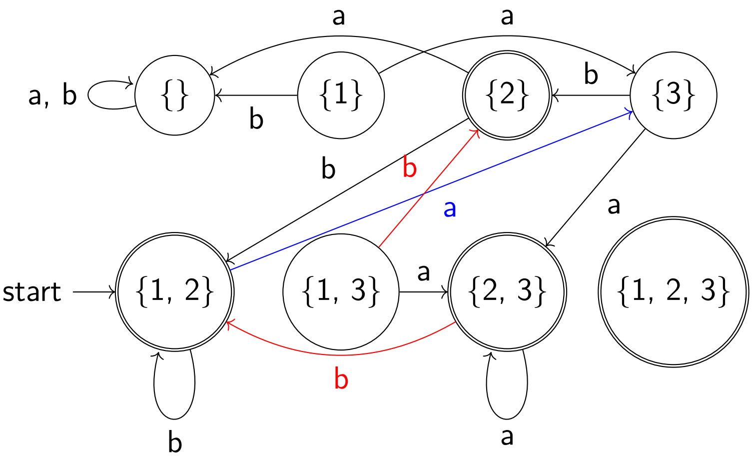
把每一個 state 都列出來



$\delta'(\{1,3\},a)=\{2,3\}$
$\delta'(\{1,3\},b)=\{2\}$

Hw3 Problem3

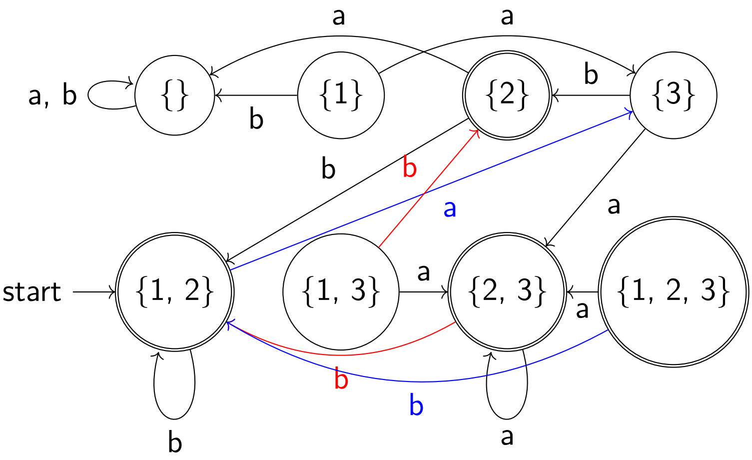
把每一個 state 都列出來



$\delta'(\{2,3\},a) = \{2,3\}$
$\delta'(\{2,3\},b) = \{1,2\}$

Hw3 Problem3

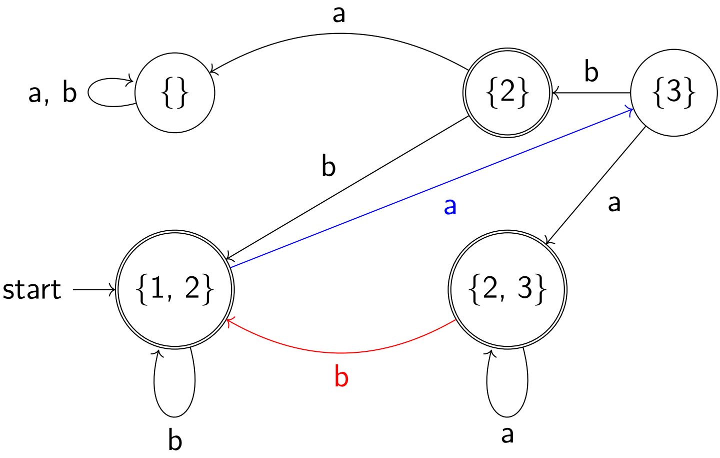
把每一個 state 都列出來



$\delta'(\{1,2,3\},a) = \{2,3\}$
$\delta'(\{1,2,3\},b) = \{1,2\}$

Hw3 Problem3

把每一個 state 都列出來



刪去無法到達的狀態

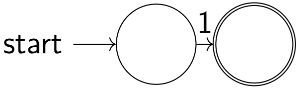
$\{1\}$，$\{1, 3\}$ 和 $\{1, 2, 3\}$ 這三個節點一定無法到達

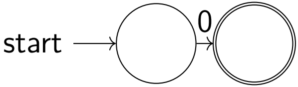
Hw3 Problem4

(Exercise 1.18; 10 points) Use the procedure described in Lemma 1.55 to convert the regular expression $(0 \cup 1)^{+}011(0 \cup 1)^{*}$ into an NFA.

Hw3 Problem4

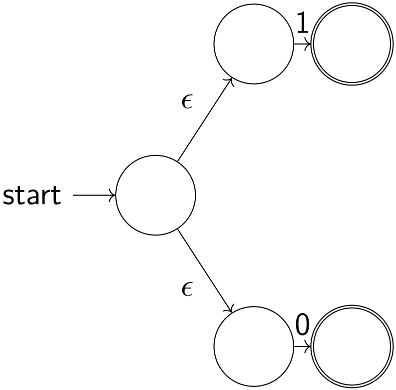
$0 \quad 1$





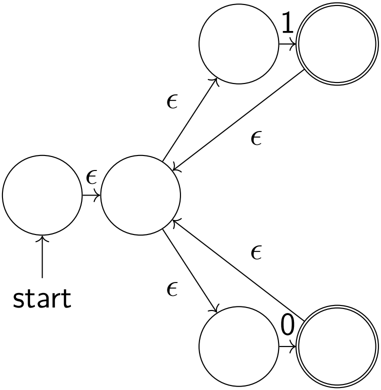
Hw3 Problem4

$0 \cup 1$



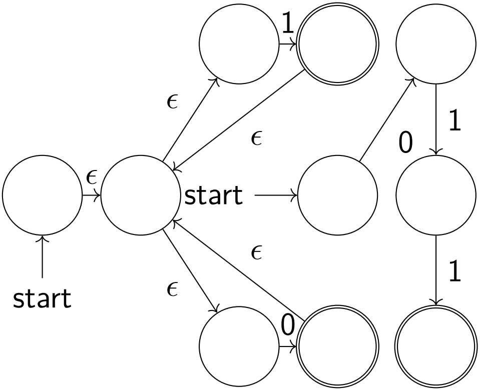
Hw3 Problem4

$(0 \cup 1)^+$



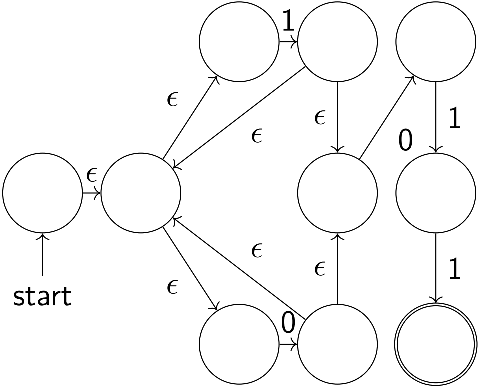
Hw3 Problem4

$(0 \cup 1)^{+}011$



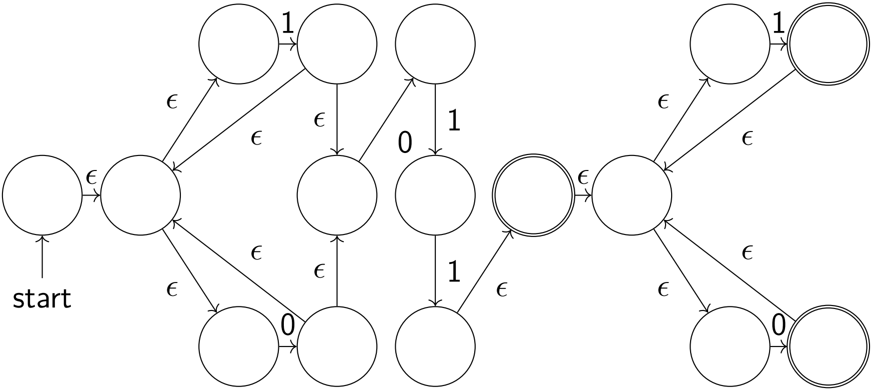
Hw3 Problem4

$(0 \cup 1)^+011$



Hw3 Problem4

$(0 \cup 1)^{+}011(0 \cup 1)^{*}$



Hw3 Problem5

(Exercise 1.20; 10 points) Give regular expressions generating the following languages:

(a) $\{w \mid w$ contains the substring $\texttt{0101}$, i.e., $w = x\texttt{0101}y$ for some x and $y\}$

(b) $\{w \mid w$ doesn't contain the substring $\texttt{011}\}$

Hw3 Problem5

第一小題：包含子字串：0101
把 0101 明確表達即可，前後可加可不加
$(0 \cup 1)^*0101(0 \cup 1)^*$

Hw3 Problem5

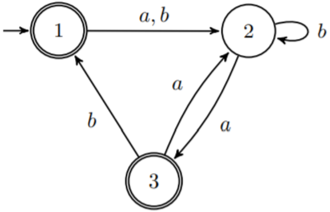
第二小題：字串不包含子字串 011
只要出現 0，後面就不能出現 2 個以上的 1
暗示第一次遇見 0 之前可以有一堆 1（開頭為 1^*）
接下來我們開始考慮一旦遇到 0 該怎麼辦
因為此題設計的初衷是希望可以直觀的思考解題過程
所以不考慮轉換 DFA accepting states 找補集的作法
而是考慮把0 後面怎麼接給表達出來

Hw3 Problem5

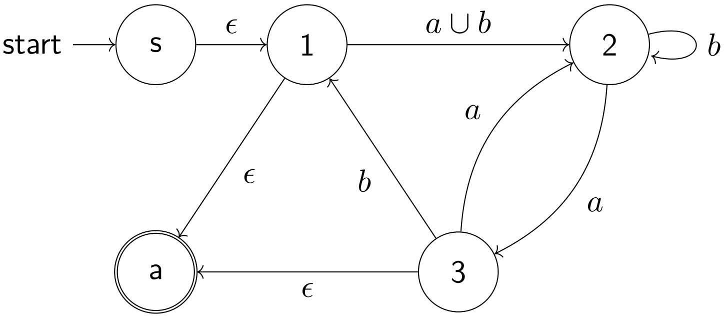
我們這裡使用看起來很暴力的方法
0 後面除了 11 之外可以接上：0 1 00 01 10 或是什麼都不接
所以可以寫成 $0(\epsilon \cup 0 \cup 1 \cup 00 \cup \cup 01 \cup 10)$
結合上前面提到的 1^*，這題的答案變成：
$1^*(0(\epsilon \cup 0 \cup 1 \cup 00 \cup \cup 01 \cup 10))^*$
可以再進行化簡，但題目沒有硬性規定

Hw3 Problem6

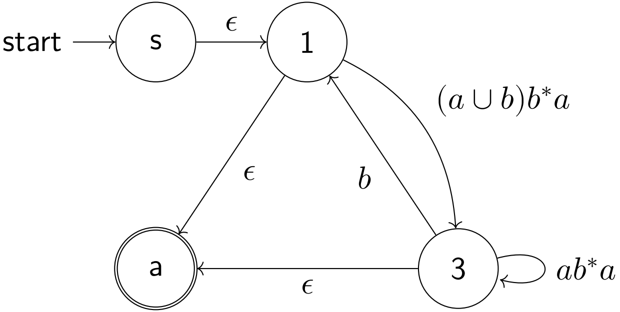
(Exercise 1.21; 20 points) Use the procedure described in Lemma 1.60 to convert the following finite automaton into a regular expression.



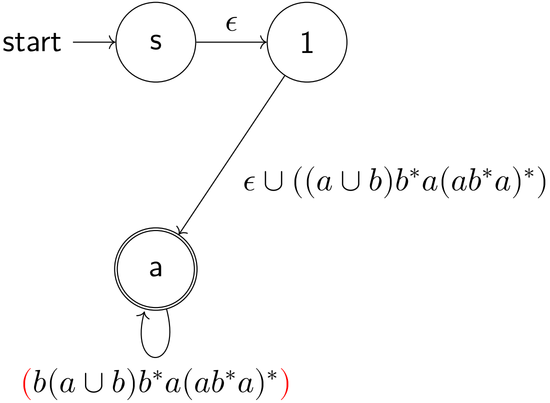
Hw3 Problem6



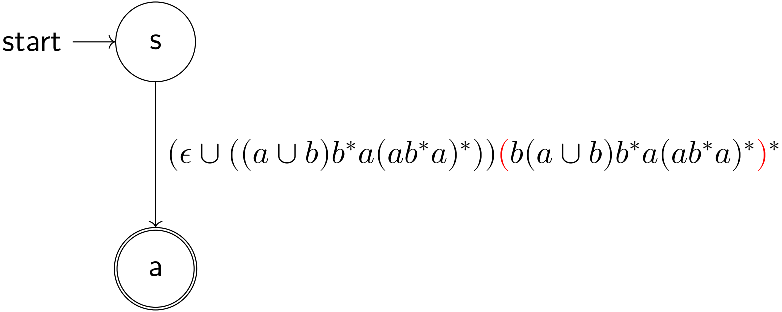
Hw3 Problem6



Hw3 Problem6



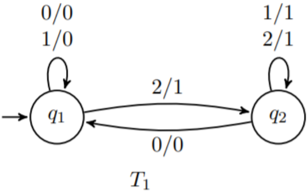
Hw3 Problem6

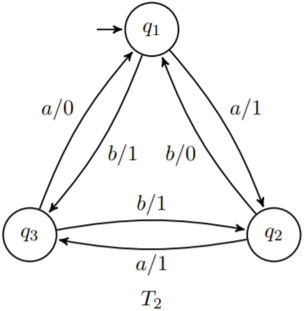


本題根據不同的刪除 state 過程，而有不同的答案！

Hw3 Problem7

(Exercise 1.24; 10 points) A *finite-state transducer* (FST) is a type of deterministic finite automaton whose output is a string rather than *accept* or *reject*. The following are state diagrams of finite state transducers T_1 and T_2.





Hw3 Problem7

Each transition of an FST is labeled with two symbols, one designating the input symbol for that transition and the other designating the output symbol. The two symbols are written with a slash, /, separating them. In T_1, the transition from q_1 to q_2 has input symbol 2 and output symbol 1. Some conditions may have multiple input-output pairs, such as the transition in T_1 from q_1 to itself. When an FST computes on an input string w, it takes the input symbols $w_1 \cdots w_n$ one by one and, starting from the start state, follows the transitions by matching the input labels with the sequence of symbols $w_1 \cdots w_n = w$. Every time it goes along a transition, it outputs the corresponding output symbol. For example, on input 2212011, machine T_1 enters the sequence of states $q_1, q_2, q_2, q_2, q_2, q_1, q_1, q_1$ and produces output 1111000. On input `abbb`, T_2 outputs 1011. Give the sequence of states entered and the output produced in each of the following parts.

(a) T_1 on input 122021

(b) T_2 on input `baaabb`

Hw3 Problem7

q_1
q_1 吃 1 輸出 0 跑到 q_1
q_1 吃 2 輸出 1 跑到 q_2
q_2 吃 2 輸出 1 跑到 q_2
q_2 吃 0 輸出 0 跑到 q_1
q_1 吃 2 輸出 1 跑到 q_2
q_2 吃 1 輸出 1 跑到 q_2
輸出 011011

Hw3 Problem7

q_1
q_1 吃 b 輸出 1 跑到 q_3
q_3 吃 a 輸出 0 跑到 q_1
q_1 吃 a 輸出 1 跑到 q_2
q_2 吃 a 輸出 1 跑到 q_3
q_3 吃 b 輸出 1 跑到 q_2
q_2 吃 b 輸出 0 跑到 q_1
輸出 101101

Hw3 Problem8

(Exercise 1.25; 10 points) Read the informal definition of the finite state transducer given in Exercise 1.24. Give a formal definition of this model, following the patterns in Definition 1.5 (Page 35 in Sipser's book or Page 7 of the slides). Assume that an FST has an input alphabet Σ and an output alphabet Γ but not a set of accept states. Include a formal definition of the computation of an FST. (Hint: an FST is a 5-tuple. Its transition function is of the form $\delta : Q \times \Sigma \longrightarrow Q \times \Gamma$.)

Hw3 Problem8

An FST T is a 5-tuple $(Q, \Sigma, \Gamma, \delta, q_0)$
Q is a finite set of states
Σ is a finite set of input symbols
Γ is a finite set of output symbols
$\delta : Q \times \Sigma \to Q \times \Gamma$ is the transition function
$q_0 \in Q$ is the start state

Let $w = w_1 w_2 ... w_n$ be a string over Σ and $x = x_1 x_2 ... x_n$ a string over Γ
We say T produces output x on input w with the sequence of states $r_0, r_1, ..., r_n$ when

- $r_0 = q_0$
- $\delta(r_i, w_{i+1}) = (r_{i+1}, x_{i+1})$ for $i = 0, 1, ..., n-1$

Hw3 Problem8

最容易錯的點：
沒注意到題目有說要寫出 FST 的運作過程

Hw4 Problem1

(Problem 1.43; 10 points) An *all*-NFA M is a 5-tuple $(Q, \Sigma, \delta, q, F)$ that accepts $x \in \Sigma^*$ if every possible state that M could be after reading input x is a state from F. Note, in contrast, that an ordinary NFA accepts a string if *some* state among these possible states is an accept state. Prove that all-NFAs recognize the class of regular languages.

Hw4 Problem1

We need to prove the following two assumptions:

- All regular languages can be recognized by an all-NFA.
- All languages all-NFAs recognize are regular.

Assumptions: All regular languages can be recognized by an all-NFA.

Proof: All regular languages are recognized by a DFA, and DFA is also an all-NFA because all the accepting runs terminate at the accepting states.

Hw4 Problem1

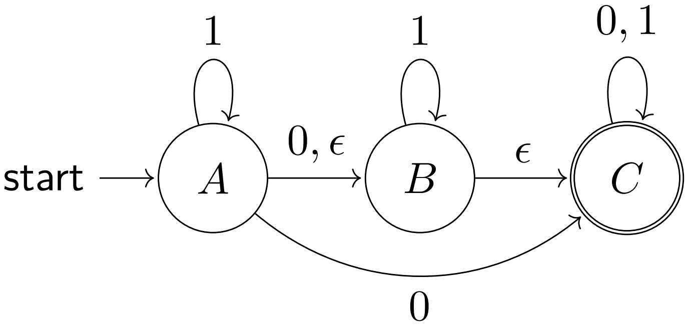
Assumptions: All languages all-NFAs recognize are regular.

Proof: Suppose that A is the language that an all-NFA $N = (Q, \Sigma, \delta, q, F)$ recognizes. Now we can construct a DFA $M = (Q', \Sigma, \delta', q', F')$ that recognizes A as follows:

- $Q' = P(Q)$ (the power set of Q).
- δ' is the ϵ-closure of transitions from the elements of the state-set.
- $q' = \{q\}$.
- $F' = P(F)$.

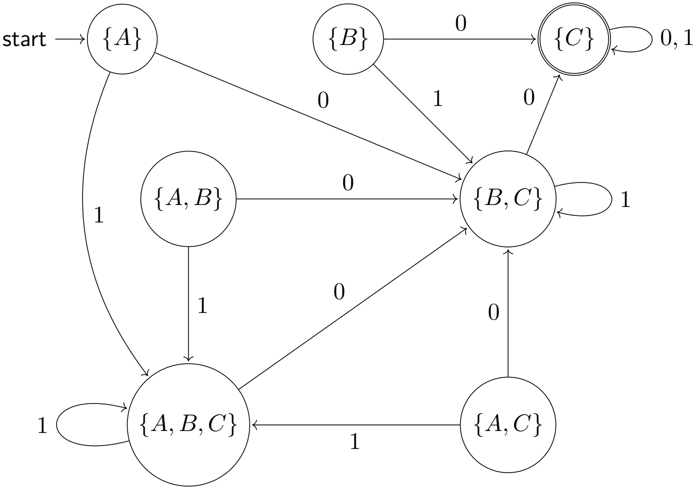
Hw4 Problem1

For example: all-NFA N:



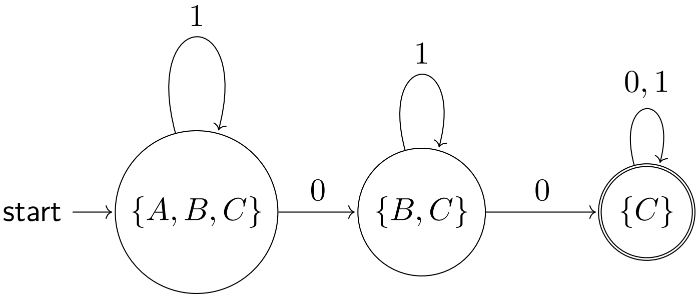
Hw4 Problem1

For example: DFA M:



Hw4 Problem1

Simplify M:



Hw4 Problem2

(Problem 1.31; 20 points) For languages A and B, let the *perfect shuffle* of A and B be the language $\{w \mid w = a_1b_1 \cdots a_kb_k,$ where $a_1 \cdots a_k \in A$ and $b_1 \cdots b_k \in B$, each $a_i, b_i \in \Sigma\}$. Show that the class of regular languages is closed under perfect shuffle.

Hw4 Problem2

Let $M_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$ and $M_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ be two DFAs that recognize two regular languages A and B, respectively. Now we can construct a DFA $D = (Q, \Sigma, \delta, q, F)$ that recognizes the perfect shuffle of A and B as follows:

- $Q = Q_A \times Q_B \times \{A, B\}$.
- $q = \{q_A, q_B, A\}$.
- $\delta((x, y, A), a) = (\delta_A(x, a), y, B)$ and $\delta((x, y, B), a) = (x, \delta_B(y, a), A)$.
- $F = F_A \times F_B \times \{A\}$.

Hw4 Problem3

(Problem 1.38; 20 points) Let

$$\Sigma_2 = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}.$$

Here, Σ_2 contains all columns of 0s and 1s of length two. A string of symbols in Σ_2 gives two rows of 0s and 1s. Consider each row to be a binary number and let

$$C = \{w \in \Sigma_2^* \mid \text{the bottom row of } w \text{ is three times the top row}\}.$$

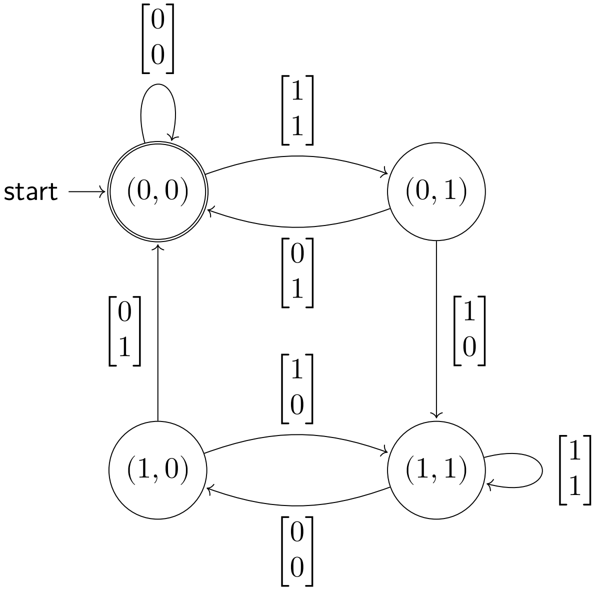
For example, $\begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \in C$, but $\begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \notin C$. Show that C is regular. (You may assume the result claimed in Problem 5 of HW#2.)

Hw4 Problem3

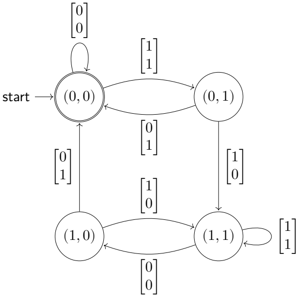
Use the process solving Problem 5 of HW#2: construct a DFA that recognizes C^R.

Because 3 times means 1 + 2 times, and 2 times in binary system means shifting 1 bit left, we need to consider not only carry but also the digit shifted from the right.

Hw4 Problem3



Hw4 Problem3



The pair (c, r) in each state means:

- $c = 1$ if there is a carry and $c = 0$ if not.
- $r = 0$ if the digit shifted from the right is 0 and $r = 1$ if 1.

So if we move from state (c_s, r_s) with transition $\begin{bmatrix} b_{top} \\ b_{bot} \end{bmatrix}$, we will reach the state $((b_{top} + c_s + r_s)/2, b_{top})$ and $b_{bot} = (b_{top} + c_s + r_s)\%2$.

Hw4 Problem4

(Problem 1.40; 20 points) Let Σ_2 be the same as in Problem 3. Consider the top and bottom rows to be strings of 0s and 1s and let

$$E = \{w \in \Sigma_2^* \mid \text{the bottom row of } w \text{ is the reverse of the top row of } w\}.$$

Show that E is not regular.

Hw4 Problem4

Use the pumping lemma:

Let s be $\begin{bmatrix}0\\1\end{bmatrix}^p\begin{bmatrix}1\\0\end{bmatrix}^p$, where p is the pumping length for E.

When dividing s as xyz, because $|xy| < p$, y must consist of $\begin{bmatrix}0\\1\end{bmatrix}$s.

And obviously, $xy^2z \notin E$ (the number of 0 is different between the top and the bottom rows).

Hw4 Problem5

(Problem 1.51; 10 points) Prove that the language $\{w \in \{0,1\}^* \mid w \text{ is not a palindrome}\}$ is not regular. You may use the pumping lemma and the closedness of the class of regular languages under union, intersection, and complement. (Note: a *palindrome* is a string that reads the same forward and backward.)

Hw4 Problem5

Let $\bar{A} = \{w \in \{0, 1\}* \mid w$ is not a palindrome$\}$.

Because the class of regular languages is closed under complement, if A is regular, $\bar{A}$ must be regular. On the other hand, if A is not regular, $\bar{A}$ must not be regular.

Hw4 Problem5

Prove that $A = \{w \in \{0,1\}* \mid w$ is a palindrome$\}$ is not regular.

Use the pumping lemma:

Let s be 0^p10^p, where p is the pumping length for A.
When dividing s as xyz, because $|xy| < p$, y must consist of 0s.
And obviously, $xy^2z \notin A$ (the number of 0 is different on both sides of 1).

Hw4 Problem6

(Problem 1.66; 20 points) Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA and let h be a state of M called its "home". A *synchronizing sequence* for M and h is a string $s \in \Sigma^*$ where $\delta(q, s) = h$ for every $q \in Q$. Say that M is *synchronizable* if it has a synchronizing sequence for some state h. Prove that, if M is a k-state synchronizable DFA, then it has a synchronizing sequence of length at most k^3. (Note: $\delta(q, s)$ equals the state where M ends up, when M starts from state q and reads input s.)

Hw4 Problem6

We first start from two states q_A and q_B of Q.

Let s_{AB} be a string that leads q_A and q_B into the same state g.

The length of s_{AB} is at most $k*(k-1)$. Because the pairs of different two states in Q are at most $k*(k-1)$, if the length of s_{AB} is $k*(k-1)+1$, there must be two repeated pairs, which means that the substring between them could be removed.

For example: if s_{AB} can be divided as $s_1s_2s_3$ such that

$$(q_A, q_B) \xrightarrow{s_1} (q'_A, q'_B) \xrightarrow{s_2} (q'_A, q'_B) \xrightarrow{s_3} (g, g)$$

Then s_2 can be removed.

Hw4 Problem6

Now we have k states in Q. We can first run s_{AB} with the length at most $k*(k-1)$ so that q_A and q_B will transfer to the same state. Then, we can similarly run s_{BC} to make q_B and q_C transfer to the same state, which means that q_A, q_B and q_C are in the same state.

By repeating the steps above $k-1$ times, all k states will be transferred to the same state, which is h. And we can obtain our synchronizing sequence s with the length at most $k*(k-1)^2 \leq k^3$.

Hw5 Problem1

(Exercise 2.1; 20 points) Consider the following CFG discussed in class, where for convenience the variables have been renamed with single letters.

$$
\begin{aligned}
E &\rightarrow E + T \mid T \\
T &\rightarrow T \times F \mid F \\
F &\rightarrow (E) \mid a
\end{aligned}
$$

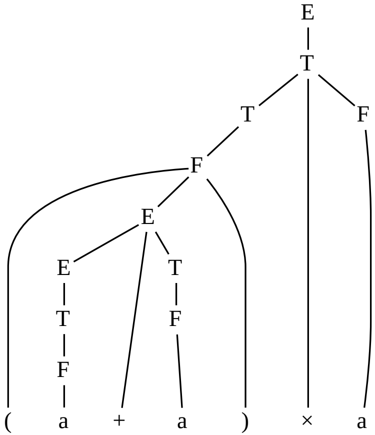
Give (leftmost) derivations and the corresponding parse trees for the following strings.

(a) $(a + a) \times a$

(b) $((a) + a)$

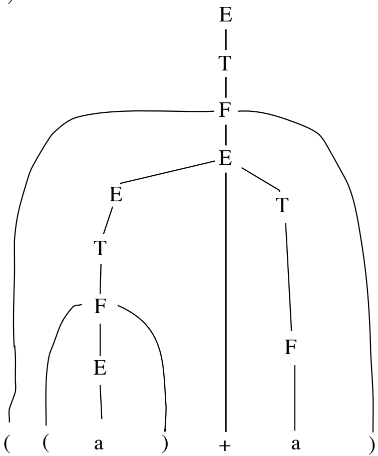
Hw5 Problem1

第一小題：$(a + a) \times a$



Hw5 Problem1

第二小題：$((a) + a)$



Hw5 Problem2

(Exercise 2.4; 20 points) Give context-free grammars that generate the following languages. In all parts the alphabet Σ is $\{0, 1\}$.

(a) $\{w \mid \text{the length of } w \text{ is a multiple of } 3\}$

(b) $\{w \mid w = w^R, \text{ that is, } w \text{ is a palindrome}\}$

Hw5 Problem2

第一小題
$S \to A$
$A \to CCCA \mid \epsilon$
$C \to 0 \mid 1$

C 生成長度為 1 的字串，A 生成長度為三的倍數的字串，而初始符號 S 則生成 A

Hw5 Problem2

第二小題

$S \to 0S0 \mid 1S1 \mid C \mid \epsilon$

$C \to 0 \mid 1$

為何不直接 CSC 呢？因為兩個 C 可能會生出不同字，就不會是回文了

Hw5 Problem3

(Exercise 2.6b; 10 points) Give a context-free grammar that generates the complement of the language $\{a^n b^n \mid n \geq 0\}$.

Hw5 Problem3

造出一個 CFG 生成 $\{a^n b^n \mid n \geq 0\}$ 的補集

我們可以分成兩種情況來討論
第一種情況為字串中有 ba 出現
如此一來不論 a 和 b 的數量是否相同，其都不可能符合 $\{a^n b^n \mid n \geq 0\}$ 第二種情況為 a 和 b 的數量不一樣明顯不符合 $\{a^n b^n \mid n \geq 0\}$

Hw5 Problem3

$S \to S_1 \mid S_2$

$S_1 \to ba \mid XS_1 \mid S_1X$
$X \to a \mid b$

$S_2 \to AC \mid CB$
$A \to aA \mid a$
$B \to bB \mid b$
$C \to ab \mid aCb$

Hw5 Problem4

(Exercise 2.8; 10 points) Show that the string "`a girl touches the boy with a flower`" has two different leftmost derivations in the following CFG.

$$
\begin{array}{rcl}
\langle\text{SENTENCE}\rangle & \to & \langle\text{NOUN-PHRASE}\rangle\langle\text{VERB-PHRASE}\rangle \\
\langle\text{NOUN-PHRASE}\rangle & \to & \langle\text{CMPLX-NOUN}\rangle \mid \\
 & & \langle\text{CMPLX-NOUN}\rangle\langle\text{PREP-PHRASE}\rangle \\
\langle\text{VERB-PHRASE}\rangle & \to & \langle\text{CMPLX-VERB}\rangle \mid \\
 & & \langle\text{CMPLX-VERB}\rangle\langle\text{PREP-PHRASE}\rangle \\
\langle\text{PREP-PHRASE}\rangle & \to & \langle\text{PREP}\rangle\langle\text{CMPLX-NOUN}\rangle \\
\langle\text{CMPLX-NOUN}\rangle & \to & \langle\text{ARTICLE}\rangle\langle\text{NOUN}\rangle \\
\langle\text{CMPLX-VERB}\rangle & \to & \langle\text{VERB}\rangle \mid \langle\text{VERB}\rangle\langle\text{NOUN-PHRASE}\rangle \\
\langle\text{ARTICLE}\rangle & \to & \texttt{a} \mid \texttt{the} \\
\langle\text{NOUN}\rangle & \to & \texttt{boy} \mid \texttt{girl} \mid \texttt{flower} \\
\langle\text{VERB}\rangle & \to & \texttt{touches} \mid \texttt{likes} \mid \texttt{sees} \\
\langle\text{PREP}\rangle & \to & \texttt{with}
\end{array}
$$

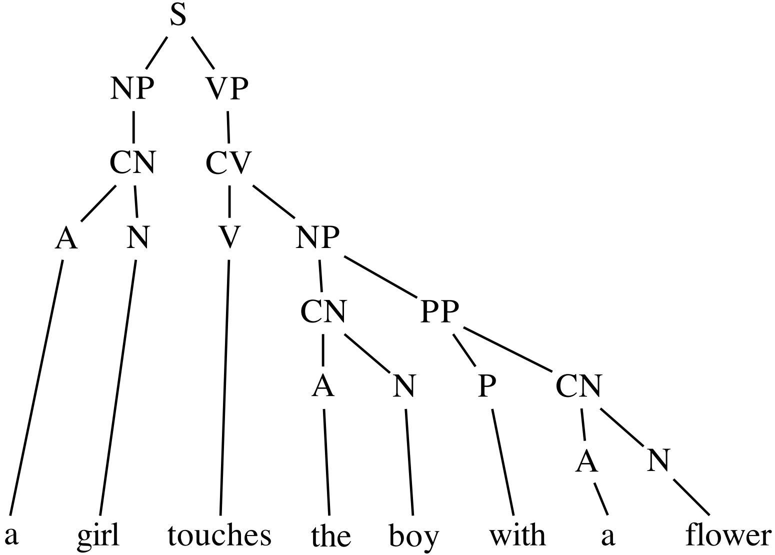
Hw5 Problem4

第一種：女孩碰有花的男孩

$\mathrm{S} \Rightarrow \mathrm{NP}\,\mathrm{VP} \Rightarrow \mathrm{CN}\,\mathrm{VP} \Rightarrow \mathrm{A}\,\mathrm{N}\,\mathrm{VP} \Rightarrow$ a $\mathrm{N}\,\mathrm{VP} \Rightarrow$ a girl $\mathrm{VP} \Rightarrow$
a girl $\mathrm{CV} \Rightarrow$ a girl $\mathrm{V}\,\mathrm{NP} \Rightarrow$ a girl touches $\mathrm{NP} \Rightarrow$
a girl touches $\mathrm{CN}\,\mathrm{PP} \Rightarrow$ a girl touches $\mathrm{A}\,\mathrm{N}\,\mathrm{PP} \Rightarrow$
a girl touches the $\mathrm{N}\,\mathrm{PP} \Rightarrow$ a girl touches the boy $\mathrm{PP} \Rightarrow$
a girl touches the boy $\mathrm{P}\,\mathrm{CN} \Rightarrow$ a girl touches the boy with $\mathrm{CN} \Rightarrow$
a girl touches the boy with $\mathrm{A}\,\mathrm{N} \Rightarrow$ a girl touches the boy with a $\mathrm{N} \Rightarrow$
a girl touches the boy with a flower

Hw5 Problem4

第一種：女孩碰有花的男孩



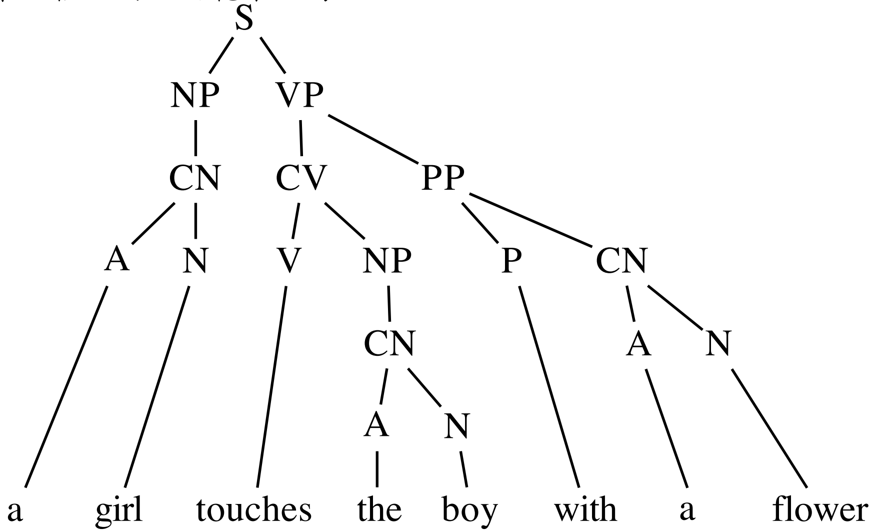
Hw5 Problem4

第二種：女孩用花來碰男孩

S $\Rightarrow$ NP VP $\Rightarrow$ CN VP $\Rightarrow$ A N VP $\Rightarrow$ a N VP $\Rightarrow$ a girl VP $\Rightarrow$
a girl CV PP $\Rightarrow$ a girl V NP PP $\Rightarrow$ a girl touches NP PP $\Rightarrow$
a girl touches CN PP $\Rightarrow$ a girl touches A N PP $\Rightarrow$
a girl touches the N PP $\Rightarrow$ a girl touches the boy PP $\Rightarrow$
a girl touches the boy P CN $\Rightarrow$ a girl touches the boy with CN $\Rightarrow$
a girl touches the boy with A N $\Rightarrow$ a girl touches the boy with a N $\Rightarrow$
a girl touches the boy with a flower

Hw5 Problem4

第二種：女孩用花來碰男孩



Hw5 Problem5

(Exercise 2.9; 20 points) Give a context-free grammar that generates the language

$$A = \{a^i b^j c^k \mid i = j \text{ or } j = k \text{ where } i, j, k \geq 0\}.$$

Is your grammar ambiguous? Why or why not?

Hw5 Problem5

設計一個 CFG 生成 $a^ib^jc^k$ 其中 $i = j \vee j = k$
我們可以走兩條路線：$i = j$ 路線或 $j = k$ 路線
對 $i = j$ 路線而言，左半邊要有等量生成的 a 與 b，右邊的 c 則是任意生成
$j = k$ 路線也是以此類推

Hw5 Problem5

$S \to UC \mid AV$
$U \to aUb \mid \epsilon$
$V \to bVc \mid \epsilon$
$A \to aA \mid \epsilon$
$C \to cC \mid \epsilon$

解釋這個 CFG 是否為 ambiguous
考慮字串 abc，可以有兩條路線：
$S \Rightarrow UC \Rightarrow aUbC \Rightarrow abC \Rightarrow abcC \Rightarrow abc$
$S \Rightarrow AV \Rightarrow aAV \Rightarrow aV \Rightarrow abVc \Rightarrow abc$

Hw5 Problem6

(Exercise 2.14; 20 points) Convert the following CFG (where A is the start variable) into an equivalent CFG in Chomsky normal form, using the procedure given in Theorem 2.9.

$$\begin{aligned} A &\rightarrow BAB \mid B \mid \varepsilon \\ B &\rightarrow 0B1 \mid \varepsilon \end{aligned}$$

Hw5 Problem6

$$A \to BAB \mid B \mid \epsilon$$
$$B \to 0B1 \mid \epsilon$$

Hw5 Problem6

第一程序：增加新的 start symbol

加上 $S_0 \rightarrow A \quad S_0 \rightarrow A$

$A \rightarrow BAB \mid B \mid \epsilon$

$B \rightarrow 0B1 \mid \epsilon$

Hw5 Problem6

第二程序：去除 ϵ

去除 $B \to \epsilon \quad S_0 \to A$

$A \to BAB \mid B \mid \epsilon \mid BA \mid AB \mid A$

$B \to 0B1$

Hw5 Problem6

第二程序：去除 ϵ
去除 $A \to \epsilon \quad S_0 \to A \mid \epsilon$
$A \to BAB \mid B \mid BA \mid AB \mid A \mid BB$
$B \to 0B1$

Hw5 Problem6

第三程序：去除 unit rule

去除 $A \rightarrow A \quad S_0 \rightarrow A \mid \epsilon$

$A \rightarrow BAB \mid B \mid BA \mid AB \mid BB$

$B \rightarrow 0B1$

Hw5 Problem6

第三程序：去除 unit rule

去除 $A \to B \quad S_0 \to A \mid \epsilon$

$A \to BAB \mid BA \mid AB \mid BB \mid 0B1$

$B \to 0B1$

Hw5 Problem6

第三程序：去除 unit rule

去除 $S \rightarrow A \quad S_0 \rightarrow BAB \mid BA \mid AB \mid BB \mid 0B1 \mid \epsilon$

$A \rightarrow BAB \mid BA \mid AB \mid BB \mid 0B1$

$B \rightarrow 0B1$

Hw5 Problem6

第四程序：分割其它 rule

去除 $S_0 \to BAB$ 與 $A \to BAB$

$S_0 \to BC_1 \mid BA \mid AB \mid BB \mid 0B1 \mid \epsilon$

$A \to BC_2 \mid BA \mid AB \mid BB \mid 0B1$

$B \to 0B1 \quad C_1 \to AB$

$C_2 \to AB$

Hw5 Problem6

第四程序：分割其它 rule

去除 $S \to 0B1$、$A \to 0B1$ 與 $B \to 0B1$

$S_0 \to BC_1 \mid BA \mid AB \mid BB \mid C_3 1 \mid \epsilon$

$A \to BC_2 \mid BA \mid AB \mid BB \mid C_4 1$

$B \to C_5 1 \quad C_1 \to AB$

$C_2 \to AB \quad C_3 \to 0B$

$C_4 \to 0B$

$C_5 \to 0B$

Hw5 Problem6

$$
\begin{aligned}
&S_0 \to BC_1 \mid BA \mid AB \mid BB \mid C_3I_1 \mid \epsilon \\
&A \to BC_2 \mid BA \mid AB \mid BB \mid C_4I_2 \\
&B \to C_5I_3 \quad C_1 \to AB \\
&C_2 \to AB \quad C_3 \to O_1B \\
&C_4 \to O_2B \\
&C_5 \to O_3B \\
&I_1 \to 1 \\
&I_2 \to 1 \\
&I_3 \to 1 \\
&O_1 \to 0 \\
&O_2 \to 0 \\
&O_3 \to 0
\end{aligned}
$$