1. Consider the following relation instance r of the relational schema $R(A, B, C, D)$.

R

A	B	C	D
8	6	1	7
0	4	1	9
8	6	1	7
8	5	2	7

List all the functional dependencies of the form $\alpha \rightarrow \beta$ where $\alpha \subseteq R$[1] and $\beta \in R$ that *definitely* do not hold on R. In fact, this can be symbolised as $\alpha \nrightarrow \beta$.

Solution:

- $A \nrightarrow B$
- $A \nrightarrow C$
- $C \nrightarrow A$
- $C \nrightarrow B$
- $C \nrightarrow D$
- $D \nrightarrow B$
- $D \nrightarrow C$
- $AD \nrightarrow B$
- $AD \nrightarrow C$

2. Consider the relational schema R and let $a, b, c, d \subseteq R$. Use only Armstrong's axioms to prove the soundness of the following two inference rules:

   (a) *Pseudo Transitivity*: If $a \rightarrow b$ and $bc \rightarrow d$, then $ac \rightarrow d$.

   (b) *Composition Rule*: If $a \rightarrow b$ and $c \rightarrow d$, then $ac \rightarrow bd$.

You can try writing the proofs move systematically at https://www.comp.nus.edu.sg/~adi-yoga/CS2102/FD/.

Solution:

(a) Pseudo Transitivity

(1) $a \rightarrow b$ [Given]

(2) $ac \rightarrow bc$ [Augmentation of (1) with c]

[1] Basically, α is a *set of attributes* and β is a *single attribute*.

(3) $bc \to d$ [Given]

(4) $ac \to d$ [Transitivity of (2) and (3)]

```
{a}   -> {b}    [Given]
{a,c} -> {b,c}  [Augmentation (1) with {c}]
{b,c} -> {d}    [Given]
{a,c} -> {d}    [Transitivity (2) and (3)]
```

(b) Composition Rule

(1) $a \to b$ [Given]

(2) $c \to d$ [Given]

(3) $ac \to bc$ [Augmentation of (1) with c]

(4) $bc \to bd$ [Augmentation of (2) with b]

(5) $ac \to bd$ [Transitivity of (3) and (4)]

```
{a}   -> {b}    [Given]
{c}   -> {d}    [Given]
{a,c} -> {b,c}  [Augmentation (1) with {c}]
{b,c} -> {b,d}  [Augmentation (2) with {b}]
{a,c} -> {b,d}  [Transitivity (3) and (4)]
```

3. Consider $R(A, B, C, D, E, G)$[2] with set of functional dependencies $F = \{ABC \to E, BD \to A, CG \to B\}$.

   (a) Use Armstrong's axioms to show that F implies $CDG \to E$.

   (b) Compute $\{CDG\}^+$.

   (c) Find all the keys of R.

   You can try writing the proofs move systematically at https://www.comp.nus.edu.sg/~adi-yoga/CS2102/FD/.

Solution:

(a) Proof:

(1) $ABC \to E$ [Given]

(2) $BD \to A$ [Given]

(3) $CG \to B$ [Given]

(4) $CDG \to BD$ [Augmentation of (3) with D]

(5) $CDG \to A$ [Transitivity of (4) and (2)]

(6) $CDG \to CG$ [Reflexivity]

[2] Why G and not F? We reserve F for the set of functional dependencies.

(7) $CDG \rightarrow B$ [Transitivity of (6) and (3)]

(8) $CDG \rightarrow BCDG$ [Augmentation of (7) with CDG]

(9) $BCDG \rightarrow AB$ [Augmentation of (5) with B]

(10) $CDG \rightarrow AB$ [Transitivity of (8) and (9)]

(11) $CDG \rightarrow ABC$ [Augmentation of (10) with C]

(12) $CDG \rightarrow E$ [Transitivity of (11) and (1)]

(b) $\{CDG\}^+ = \{ABCDEG\}$.
This can be computed via closure algorithm.

(c) Observe that C, D and G do not appear in the right hand side of any functional dependencies. Therefore, every key of R must contain CDG. Meanwhile $\{CDG\}^+ = \{ABCDEG\}$ from (b), which indicates that CDG is the only key.

```
{a,b,c}   -> {e}         [Given]
{b,d}     -> {a}         [Given]
{c,g}     -> {b}         [Given]
{c,d,g}   -> {b,d}       [Augmentation (3) with {d}]
{c,d,g}   -> {a}         [Transitivity (4) and (2)]
{c,d,g}   -> {c,g}       [Reflexivity]
{c,d,g}   -> {b}         [Transitivity (6) and (3)]
{c,d,g}   -> {b,c,d,g}   [Augmentation (7) with {c,d,g}]
{b,c,d,g} -> {a,b}       [Augmentation (5) with {b}]
{c,d,g}   -> {a,b}       [Transitivity (8) and (9)]
{c,d,g}   -> {a,b,c}     [Augmentation (10) with {c}]
{c,d,g}   -> {e}         [Transitivity (11) and (1)]
{c,d,g}   -> {e}         [QED]
compute closure of {c,d,g}
compute keys of {a,b,c,d,e,g}
```

4. Consider $R(A,B,C,D,E)$ with set of functional dependencies $F = \{AB \rightarrow CDE, AC \rightarrow BDE, B \rightarrow C, C \rightarrow B, C \rightarrow D, B \rightarrow E\}$.

Find all the keys of R.

Solution: Observe that A does not appear in the right hand side of any functional dependencies. Therefore, every key of R must contain A. Let's consider all sets of attributes that contain A.

- $\{A\}^+ = \{A\}$ — A is not a key
- $\{AB\}^+ = \{ABCDE\}$ — AB is a key; any $S \supset \{AB\}$ can't be key
- $\{AC\}^+ = \{ABCDE\}$ — AC is a key; any $S \supset \{AB\}$ can't be key
- $\{AD\}^+ = \{AD\}$ — AD is not a key
- $\{AE\}^+ = \{AE\}$ — AE is not a key

- $\{ADE\}^+ = \{ADE\}$ — ADE is not a key
- All other supersets of $\{A\}$ can't be keys since they must be *proper superset* of either AB or AC. Therefore, there are only two keys: AB and AC.

```
{a,b} -> {c,d,e}          [Given]
{a,c} -> {b,d,e}          [Given]
{b}   -> {c}              [Given]
{c}   -> {b}              [Given]
{c}   -> {d}              [Given]
{b}   -> {e}              [Given]
compute closure of {a}
compute closure of {a,b}
compute closure of {a,c}
compute closure of {a,d}
compute closure of {a,e}
compute closure of {a,d,e}
compute keys of {a,b,c,d,e}
```