Homework 8

Exercise 1

Maximize xy such that p_xx + p_yy =w

$\mathcal{L}$ = xy + λ(p_xx + p_yy - w)

First order condtions

$\frac{\partial \mathcal{L}}{\partial x}$ = y + $\lambda\, p_x$ = 0

$\frac{\partial \mathcal{L}}{\partial y}$ = x + $\lambda\, p_y$ = 0

$\frac{\partial \mathcal{L}}{\partial \lambda}$ = p_x x + p_y y - w = 0

y + $\lambda\, p_x$ = 0

y = -$\lambda\, p_x$

$\lambda = \frac{-y}{p_x}$

(1) x - $\frac{p_y}{p_x}$ y = 0

(2) p_x x + p_y y = w

We can interpret (1) as the quantity of x consumed must equal the ratio of the relative price $\frac{p_y}{p_x}$ times the the quantity of y consumed.

We can interpret 2 as the total amount spent on x and y must equal wages.

$$\begin{pmatrix} 1 & \frac{-p_y}{p_x} \\ p_x & p_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ w \end{pmatrix}$$

$$\beta = \begin{pmatrix} 1 & \frac{-p_y}{p_x} \\ p_x & p_y \end{pmatrix}$$

$$c = \begin{pmatrix} 0 \\ w \end{pmatrix}$$

$\Delta = 2\, p_y$

$$\beta^{-1} = \frac{1}{2\, p_y} \begin{pmatrix} p_y & \frac{p_y}{p_x} \\ -p_x & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2\, p_x} \\ \frac{-1}{2\, p_x\, p_y} & \frac{1}{2\, p_y} \end{pmatrix}$$

$$\beta^{-1}\, c = \begin{pmatrix} \frac{1}{2} & \frac{1}{2\, p_x} \\ \frac{-1}{2\, p_x\, p_y} & \frac{1}{2\, p_y} \end{pmatrix} * \begin{pmatrix} 0 \\ w \end{pmatrix} = \begin{pmatrix} \frac{w}{2\, p_x} \\ \frac{w}{2\, p_y} \end{pmatrix}$$

$x = \frac{w}{2\,p_x}$ $y = \frac{w}{2\,p_y}$

$\epsilon_{x,p_x} = \frac{\partial x}{\partial p_x}\,\frac{p_x}{x} = \frac{-w}{2\,p_x^2}\,\frac{p_x}{x} = \frac{-w}{2\,p_x}$

$\epsilon_{x,p_y} = \frac{\partial x}{\partial p_y}\,\frac{p_y}{y} = \frac{-w}{2\,p_y^2}\,\frac{p_y}{y} = \frac{-w}{2\,p_y}$

$\epsilon_{x,p_y} = \frac{\partial x}{\partial p_y}\,\frac{p_y}{x} = 0$

$\epsilon_{y\,p_x} = \frac{\partial y}{\partial p_x}\,\frac{p_x}{y} = 0$

```
l = x * y + λ (px * x + py * y - w)
foc1 = D[l, x] == 0
foc2 = D[l, y] == 0
foc3 = D[l, λ] == 0
reducedForm = Eliminate[{foc1, foc2, foc3}, λ]
Solve[reducedForm, {x, y}]
```

```
x y + (- w + px x + py y) λ

y + px λ == 0

x + py λ == 0

- w + px x + py y == 0

w == 2 py y && px x == py y
```

$\left\{\left\{x \to \frac{w}{2\,px},\ y \to \frac{w}{2\,py}\right\}\right\}$

Exercise 2

Reduced Form

dQ = Q_w dw + Q_{P_s} dP_s

dP = P_w dw + P_s dP_s

where

$F_x = \frac{\partial F}{\partial x}$

Structural Form

dQ = S_P dP + S_w dw

dQ = D_P dP + D_{P_s} dP_s

The signs of the partial derivatives are as follows:

S_P > 0 (upward sloping supply curve)

S_w < 0 (due to increasing labor costs)

D_P< 0 (downward sloping demand curve)

D_{P_s} > 0 (goods are substitute

We can rewrite the above system as follows

1 dQ - Q_w dw - Q_{P_S} dP_s + 0 dP = 0

0 dQ - P_w dw - P_s dP_s + 1 dP = 0

1 dQ - S_w dw - 0 dP_s - S_PdP = 0

1 dQ - 0 dw - D_{P_s} dP_s - D_p dP = 0

$$\begin{pmatrix} 1 & -Q_w & -Q_{P_s} & 0 \\ 0 & -P_w & -P_s & 1 \\ 1 & -S_w & 0 & -S_P \\ 1 & 0 & D_{P_s} & -D_P \end{pmatrix} \begin{pmatrix} \mathrm{dQ} \\ \mathrm{dw} \\ \mathrm{dP}_s \\ \mathrm{dP} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

We can invert the coefficient matrix thusly:

```
In[6]:= b = {{1, -Qw, -QPs, 0}, {0, -Pw, -Ps, 1}, {1, -Sw, 0, -SP}, {1, 0, DPs, -DP}}
bInv = Inverse[b]
bInv // MatrixForm
c = {{0}, {0}, {0}, {0}}
```

```
Out[6]= {{1, -Qw, -QPs, 0}, {0, -Pw, -Ps, 1}, {1, -Sw, 0, -SP}, {1, 0, DPs, -DP}}
```

Out[7]=

$$
\begin{aligned}
\{\{&(-D_{P_s} P_w S_P - D_{P_s} S_w + D_P P_s S_w) / \\
&\quad (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w), \\
&(D_{P_s} Q_w S_P - D_P Q_{P_s} S_w) / (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - \\
&\qquad P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w),\ (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s}) / \\
&\quad (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w), \\
&(P_s Q_w S_P - P_w Q_{P_s} S_P - Q_{P_s} S_w) / \\
&\quad (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w)\}, \\
\{&(-D_{P_s} + D_P P_s - P_s S_P) / (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - \\
&\qquad P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w),\ (-D_P Q_{P_s} + D_{P_s} S_P + Q_{P_s} S_P) / \\
&\quad (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w), \\
&(D_{P_s} - D_P P_s + Q_{P_s}) / (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - \\
&\qquad P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w),\ (-Q_{P_s} + P_s S_P) / \\
&\quad (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w)\}, \\
\{&(-D_P P_w + P_w S_P + S_w) / (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - \\
&\qquad P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w),\ (D_P Q_w - Q_w S_P - D_P S_w) / \\
&\quad (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w), \\
&(D_P P_w - Q_w) / (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - \\
&\qquad P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w),\ (Q_w - P_w S_P - S_w) / \\
&\quad (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w)\}, \\
\{&(-D_{P_s} P_w + P_s S_w) / (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - \\
&\qquad P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w),\ (D_{P_s} Q_w - D_{P_s} S_w - Q_{P_s} S_w) / \\
&\quad (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w), \\
&(D_{P_s} P_w - P_s Q_w + P_w Q_{P_s}) / (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - \\
&\qquad P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w),\ (P_s Q_w - P_w Q_{P_s} - P_s S_w) / \\
&\quad (D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w)\}\}
\end{aligned}
$$

Out[8]//MatrixForm=

$$
\begin{pmatrix}
\frac{-D_{P_s} P_w S_P - D_{P_s} S_w + D_P P_s S_w}{D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w} & \frac{D_{P_s} Q_w S_P - D_P Q_{P_s} S_w}{D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D} \\
\frac{-D_{P_s} + D_P P_s - P_s S_P}{D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w} & \frac{-D_P Q_{P_s} + D_{P_s} S_P + Q_{P_s} S_P}{D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D} \\
\frac{-D_P P_w + P_w S_P + S_w}{D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w} & \frac{D_P Q_w - Q_w S_P - D_P S_w}{D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D} \\
\frac{-D_{P_s} P_w + P_s S_w}{D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D_P P_s S_w - Q_{P_s} S_w} & \frac{D_{P_s} Q_w - D_{P_s} S_w - Q_{P_s} S_w}{D_{P_s} Q_w - D_P P_s Q_w + D_P P_w Q_{P_s} - D_{P_s} P_w S_P + P_s Q_w S_P - P_w Q_{P_s} S_P - D_{P_s} S_w + D}
\end{pmatrix}
$$

Out[9]= {{0}, {0}, {0}, {0}}

In[10]:= **bInv.c**

Out[10]= {{0}, {0}, {0}, {0}}

Exercise 3

1.

$\forall_{x \in S}$ (x, x) ∈ R (by reflexivity)

$\forall_{x, y, z \in S}$ (x, y) ∈ R ∧ (y, z) ∈ R ⇒ (x, z) ∈ R_t (by definition of transitivity)

∴ (x, x) ∈ R ∧ (x, x) ∈ R ⇒ (x, x) ∈ R_t

2.
$(x, y) \in R \wedge (y, x) \in R \Rightarrow (x, y) \in R_t \; (y, x) \in R_t$ (by defintion, the original relations must be in the transitive closure)
$\therefore R_t$ is symmetric

3.

Exercise 4

For the symmetric component, we have the subset of $I \subseteq R$ such that
$\{(x, x') \in I\} \Leftrightarrow \{(x', x) \in I\}$. But by transitvity of R we know that $(x, x') \in R \wedge (x', x'') \in R \Rightarrow (x, x'') \in R$.
By symmetry we have $(x', x'') \in I \Leftrightarrow (x'', x') \in I$. Therefore it must be the case that $(x', x') \in I \wedge (x', x'') \in I \Rightarrow (x, x'') \in I$.

We know the original relation R is transitive, meaning
$\forall_{x,y,z \in X}(\{x, y) \in R\} \wedge (\{y, z) \in R\} \Leftrightarrow \{(x, z) \in R\}$. For the asymettric component, we have the subset of R which is $\{(x, x') \in P\} \Rightarrow \{(x', x) \notin P\}$. Replacing y with x', and noting the transitive nature of the original we obtain that $\forall_{x,y \in X}(\{x, x') \in R\} \wedge (\{x', z) \in R\} \Leftrightarrow \{(x, z) \in R\}$

By symmetry and transitivity we can replace y in the relationship
$(x \; P \; y) \wedge (y \; I \; z)$ to obtain $(x \; P \; z)$.

Similarly, by symmetry and transitivity, we can replace the second y in $(x \; I \; y) \wedge (y \; P \; z)$ to obtain $(x \; P \; z)$.

Exercise 5

$\forall_{x,y,z \in S} \neg(x, y) \in R \wedge \neg(y, z) \in R \Leftrightarrow \neg(x, z)$
An example of such a relationship is the “=” relationship over $\mathbb{R}$.
$\neg (x = y) \wedge \neg (y = z) \Rightarrow \neg (x = z)$
Yes. Since consumer preferences are both transitive and complete, they must also be negatively transitive.

Exercise 6

AAC means that if $\forall_{x,y \in y, x \neq y} \; x \; P \; y \Rightarrow \neg \; y \; P \; x$. That is, if x is preferred to y then y is not preferred to x.This guarantees the existence of maximal elements. Completeness means that every pair of (x, y) can compared. That means that maximal elements can be compared to each other to find the element that is “bigger” than all maximal elements. So a best element must exist.

Exercise 7

A cover C of a set S is a collection {Cn} of sets such that S ⊆ ∪C_n. That is, the set S is included in the union of the sets Cn, which may be infinite in number. An open cover of S is a cover of S containing only open sets. If C and C0 each cover S, but C in smaller in the sense that C ⊂ C0, we call C a sub-cover from C0.
A set S ⊆ X is compact iff every open cover {Cn} of S has a finite subcover $\{Cnk\}_{k=1}^{K}$ (such that S ⊆ ∪C_{n_k}).

This is to say that there is a finite subcover of S.

(0, 1) is not compact. Consider [0, 1] as the smallest cover of (0, 1). Consioder finite subsets of [0, 1] such as Cn = [0, 0.99], [0, 0.999] etc. No finite unions ∪ Cn of these sets will give the back the set [0,1]. Thus, (0, 1) is not compact.

Consider any finite set S={x, y, z...}. By definition of cover, at least one set in C_n is S. This set is a subcover of some other set. Therefore it is compact.

Computational Exercise 1

The matrix representation **G** of a binary relation G on set S is formed by a matrix whose j^{th} row and k^{th} column is 1 iff the (j, k) ∈ G and 0 otherwise.

```
In[104]:= symmetricClosure[x_] := Module[{result = x}, Do[
    If[result[[i]][[j]] == 1, result[[j]][[i]] = 1, ],
    {i, 1, Length[result]}, {j, 1, Length[result]}];
  Return[result]]
```

```
In[12]:= reflexiveClosure[x_] := Module[{result = x},
  Do[
   result[[i]][[i]] = 1,
   {i, 1, Length[x]}];
  Return[result]]
```

```
transitiveCLosure[x_] := Module[{result = x}, Return[result . result]]
```

The transitive closure is formed by squaring the original binary matrix of the relation. The reflexive closure is formed by setting all diagonal elements to 1. The symmetricClosure is formed by checking whether (a, b) is in the oriignal relation and then adding (b, a) to the new relationship.

In[18]:=
```
a = {{1, 1, 0}, {0, 0, 1}, {0, 0, 1}};
b = {{0, 0, 1}, {0, 1, 0}, {1, 0, 0}};
c = {{0, 0, 0}, {1, 0, 0}, {1, 1, 0}};
a // MatrixForm
b // MatrixForm
c // MatrixForm
```

Out[21]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Out[22]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

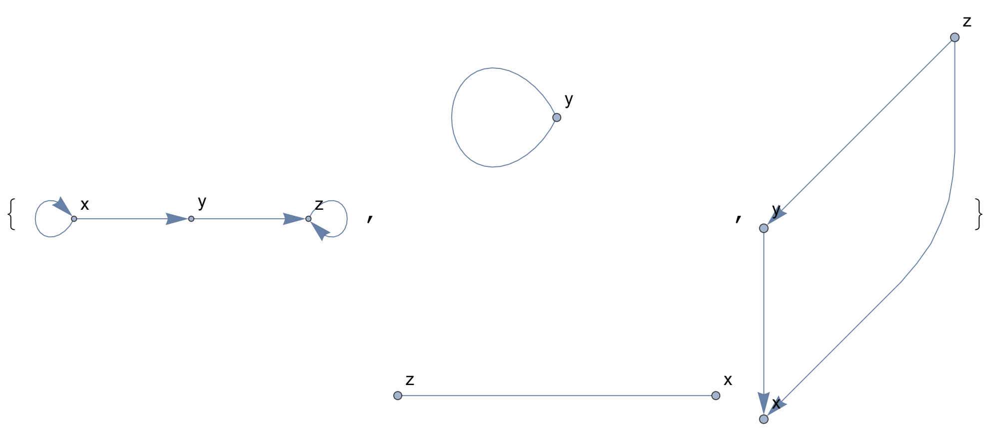
Out[23]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

xAx, xAy, yAz, zAz
xBz, yBy, zBx
yCx, zCx, zCy

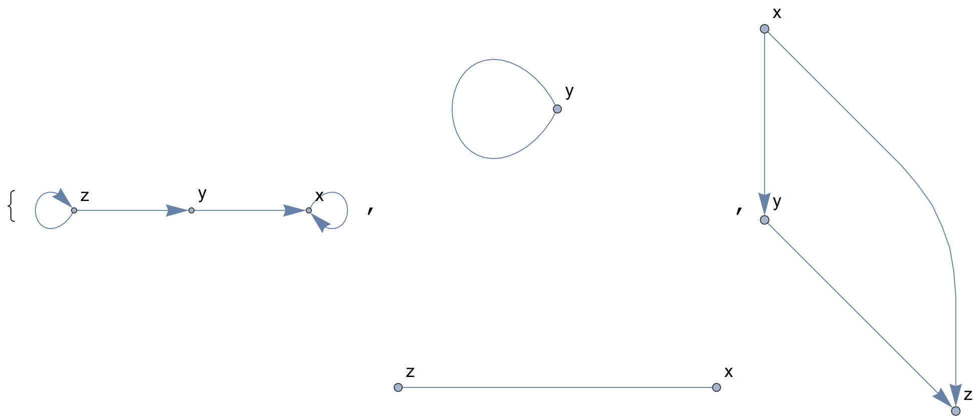
In[55]:=
```
plotNicely =
  Function[x, AdjacencyGraph[x, VertexLabels → {1 → "x", 2 → "y", 3 → "z"}]];
Map[plotNicely, {a, b, c}]
```

Out[56]=



In[57]:=
```
invA = Transpose[a ];
invB = Transpose[b];
invC = Transpose[c];
Map[plotNicely, {invA, invB, invC}]
```

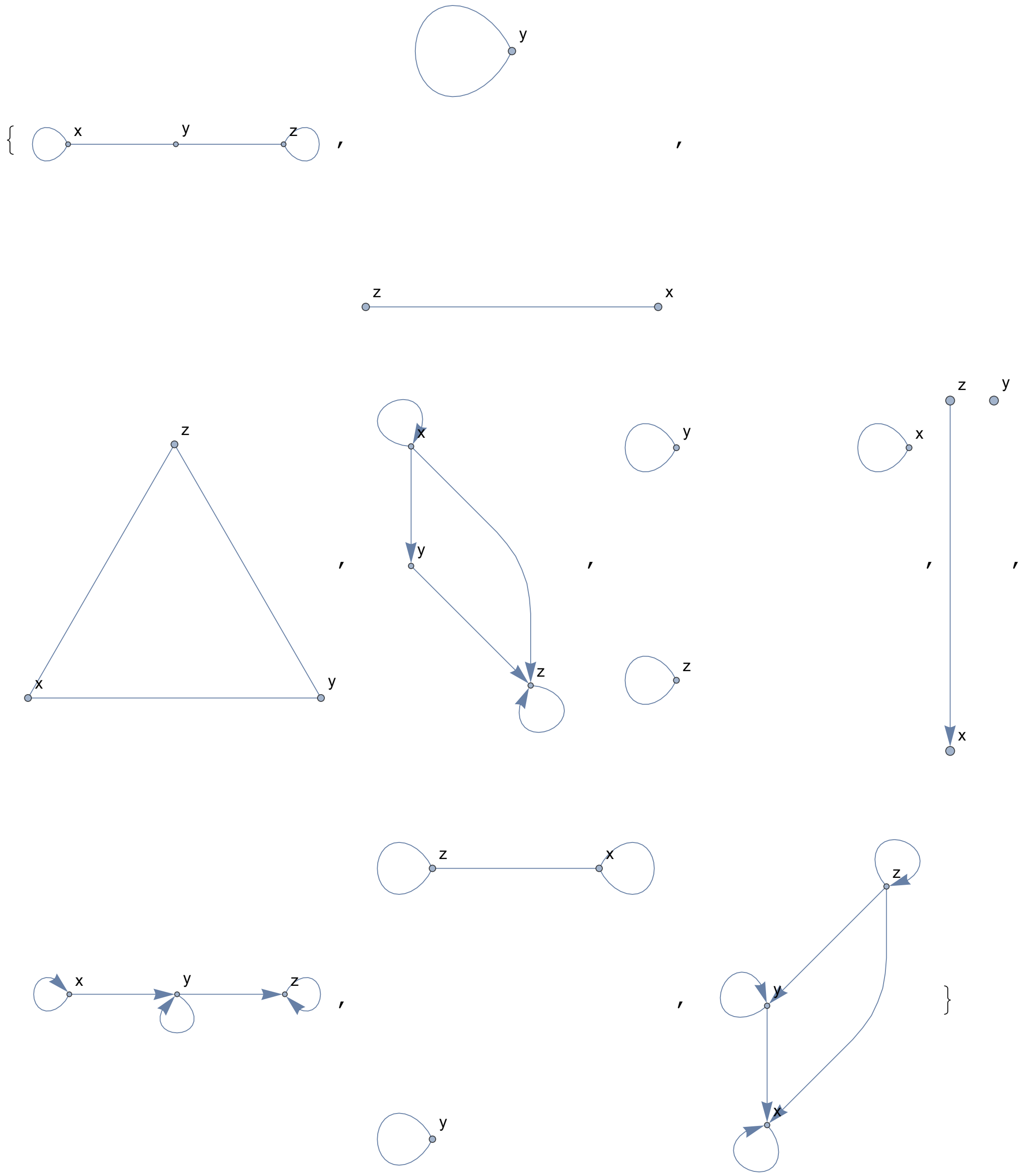
Out[60]=



In[151]:=
```
symA = symmetricClosure[a];
symB = symmetricClosure[b];
symC = symmetricClosure[c];
refA = reflexiveClosure[a];
refB = reflexiveClosure[b];
refC = reflexiveClosure[c];
tranA = transitiveCLosure[a];
tranB = transitiveCLosure[b];
tranC = transitiveCLosure[c];
collection = {symA, symB, symC, tranA, tranB, tranC, refA, refB, refC};
MatrixForm /@ collection
Map[plotNicely, collection]
```

Out[161]=
$$\left\{\begin{pmatrix}1&1&0\\1&0&1\\0&1&1\end{pmatrix}, \begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}, \begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix}, \begin{pmatrix}1&1&1\\0&0&1\\0&0&1\end{pmatrix}, \begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}, \begin{pmatrix}0&0&0\\0&0&0\\1&0&0\end{pmatrix}, \begin{pmatrix}1&1&0\\0&1&1\\0&0&1\end{pmatrix}, \begin{pmatrix}1&0&1\\0&1&0\\1&0&1\end{pmatrix}, \begin{pmatrix}1&0&0\\1&1&0\\1&1&1\end{pmatrix}\right\}$$

Out[162]=



Computational Exercise 2

In[166]:=
```
data = Table[{x, y}, {x, 1, 10}, {y, 1, 10}]
```

Out[166]=
```
{{{1, 1}, {1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}, {1, 7}, {1, 8}, {1, 9}, {1, 10}},
 {{2, 1}, {2, 2}, {2, 3}, {2, 4}, {2, 5}, {2, 6}, {2, 7}, {2, 8}, {2, 9}, {2, 10}},
 {{3, 1}, {3, 2}, {3, 3}, {3, 4}, {3, 5}, {3, 6}, {3, 7}, {3, 8}, {3, 9}, {3, 10}},
 {{4, 1}, {4, 2}, {4, 3}, {4, 4}, {4, 5}, {4, 6}, {4, 7}, {4, 8}, {4, 9}, {4, 10}},
 {{5, 1}, {5, 2}, {5, 3}, {5, 4}, {5, 5}, {5, 6}, {5, 7}, {5, 8}, {5, 9}, {5, 10}},
 {{6, 1}, {6, 2}, {6, 3}, {6, 4}, {6, 5}, {6, 6}, {6, 7}, {6, 8}, {6, 9}, {6, 10}},
 {{7, 1}, {7, 2}, {7, 3}, {7, 4}, {7, 5}, {7, 6}, {7, 7}, {7, 8}, {7, 9}, {7, 10}},
 {{8, 1}, {8, 2}, {8, 3}, {8, 4}, {8, 5}, {8, 6}, {8, 7}, {8, 8}, {8, 9}, {8, 10}},
 {{9, 1}, {9, 2}, {9, 3}, {9, 4}, {9, 5}, {9, 6}, {9, 7}, {9, 8}, {9, 9}, {9, 10}},
 {{10, 1}, {10, 2}, {10, 3}, {10, 4}, {10, 5},
  {10, 6}, {10, 7}, {10, 8}, {10, 9}, {10, 10}}}
```

In[167]:=
```
MatrixForm[data]
```

Out[167]//MatrixForm=

$$\begin{pmatrix}
\binom{1}{1} & \binom{1}{2} & \binom{1}{3} & \binom{1}{4} & \binom{1}{5} & \binom{1}{6} & \binom{1}{7} & \binom{1}{8} & \binom{1}{9} & \binom{1}{10} \\
\binom{2}{1} & \binom{2}{2} & \binom{2}{3} & \binom{2}{4} & \binom{2}{5} & \binom{2}{6} & \binom{2}{7} & \binom{2}{8} & \binom{2}{9} & \binom{2}{10} \\
\binom{3}{1} & \binom{3}{2} & \binom{3}{3} & \binom{3}{4} & \binom{3}{5} & \binom{3}{6} & \binom{3}{7} & \binom{3}{8} & \binom{3}{9} & \binom{3}{10} \\
\binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4} & \binom{4}{5} & \binom{4}{6} & \binom{4}{7} & \binom{4}{8} & \binom{4}{9} & \binom{4}{10} \\
\binom{5}{1} & \binom{5}{2} & \binom{5}{3} & \binom{5}{4} & \binom{5}{5} & \binom{5}{6} & \binom{5}{7} & \binom{5}{8} & \binom{5}{9} & \binom{5}{10} \\
\binom{6}{1} & \binom{6}{2} & \binom{6}{3} & \binom{6}{4} & \binom{6}{5} & \binom{6}{6} & \binom{6}{7} & \binom{6}{8} & \binom{6}{9} & \binom{6}{10} \\
\binom{7}{1} & \binom{7}{2} & \binom{7}{3} & \binom{7}{4} & \binom{7}{5} & \binom{7}{6} & \binom{7}{7} & \binom{7}{8} & \binom{7}{9} & \binom{7}{10} \\
\binom{8}{1} & \binom{8}{2} & \binom{8}{3} & \binom{8}{4} & \binom{8}{5} & \binom{8}{6} & \binom{8}{7} & \binom{8}{8} & \binom{8}{9} & \binom{8}{10} \\
\binom{9}{1} & \binom{9}{2} & \binom{9}{3} & \binom{9}{4} & \binom{9}{5} & \binom{9}{6} & \binom{9}{7} & \binom{9}{8} & \binom{9}{9} & \binom{9}{10} \\
\binom{10}{1} & \binom{10}{2} & \binom{10}{3} & \binom{10}{4} & \binom{10}{5} & \binom{10}{6} & \binom{10}{7} & \binom{10}{8} & \binom{10}{9} & \binom{10}{10}
\end{pmatrix}$$

In[168]:=
```
u[{x_, y_}] := Return[x * y]
```

In[173]:=
```
util = Table[u[{x, y}], {x, 1, 10}, {y, 1, 10}]
```

Out[173]=
```
{{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, {2, 4, 6, 8, 10, 12, 14, 16, 18, 20},
 {3, 6, 9, 12, 15, 18, 21, 24, 27, 30}, {4, 8, 12, 16, 20, 24, 28, 32, 36, 40},
 {5, 10, 15, 20, 25, 30, 35, 40, 45, 50}, {6, 12, 18, 24, 30, 36, 42, 48, 54, 60},
 {7, 14, 21, 28, 35, 42, 49, 56, 63, 70}, {8, 16, 24, 32, 40, 48, 56, 64, 72, 80},
 {9, 18, 27, 36, 45, 54, 63, 72, 81, 90}, {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}}
```

In[174]:=
```
util // MatrixForm
```

Out[174]//MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 & 30 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 \\ 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 & 50 \\ 6 & 12 & 18 & 24 & 30 & 36 & 42 & 48 & 54 & 60 \\ 7 & 14 & 21 & 28 & 35 & 42 & 49 & 56 & 63 & 70 \\ 8 & 16 & 24 & 32 & 40 & 48 & 56 & 64 & 72 & 80 \\ 9 & 18 & 27 & 36 & 45 & 54 & 63 & 72 & 81 & 90 \\ 10 & 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 & 100 \end{pmatrix}$$

In[178]:=
```
equivalentClass[{j_, k_}] := Position[util, j*k]
```

In[179]:=
```
equivalentClass[{6, 5}]
```

Out[179]=
```
{{3, 10}, {5, 6}, {6, 5}, {10, 3}}
```

In[181]:=
```
getLower[{j_, k_}] := Position[util, _? (# < j*k) &]
```