Mathematics (3)

LaTeX

S. Moghari

Shahrood University of Technology
Mathematical Sciences Department

Fall 2016

Outlines

1 Definitions

2 New Environments

Theorem, Lemma, Proposition, Proof , …

3 Algorithm Environment

4 Commutative Diagram

5 Graph

Definitions

```
\def\fta{$ M=(\Sigma, Q, \Gamma, \delta, \rho, \beta) $}

\def\dfta{$ M=(\Sigma, Q_d, \Gamma_d, \delta_d, \rho_d, \beta_d) $}

\def\lfta{$ M=(\Sigma, Q, \Gamma, \delta, \mathcal{L}, \rho,\beta) $}
```

New Environments

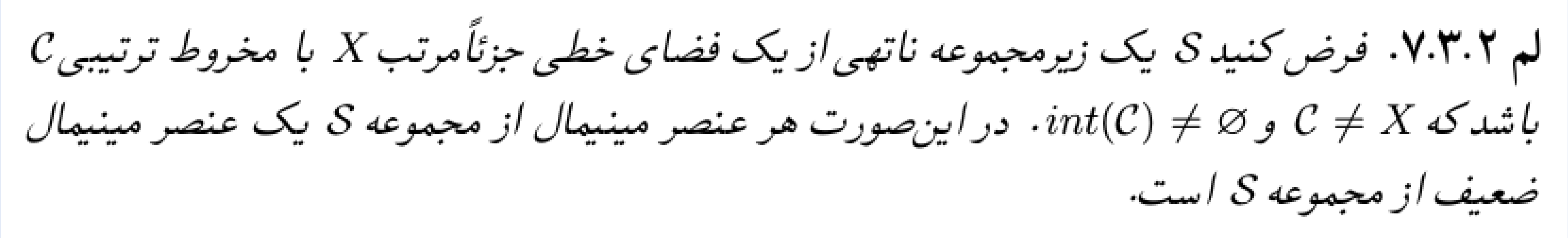
```
\newtheorem{preproof}{{\bf برهان}\hspace{-.15cm}}

\renewcommand{\thepreproof}{}

\newenvironment{proof}{\begin{preproof}{\rm}}

{\hfill{$\blacksquare$}\end{preproof}}
```

New Environments

```
\begin{theorem}{قضیه}[section]
\begin{lemma}[theorem]{لم}
\begin{proposotion}[theorem]{گزاره}
\begin{conjunction}[theorem]{حدس}
\begin{corollary}[theorem]{نتیجه}
\begin{question}[theorem]{پرسش}
\begin{remark}[theorem]{یادآوری}
\begin{definition}[theorem]{تعریف}
\begin{example}[theorem]{مثال}
```

Example

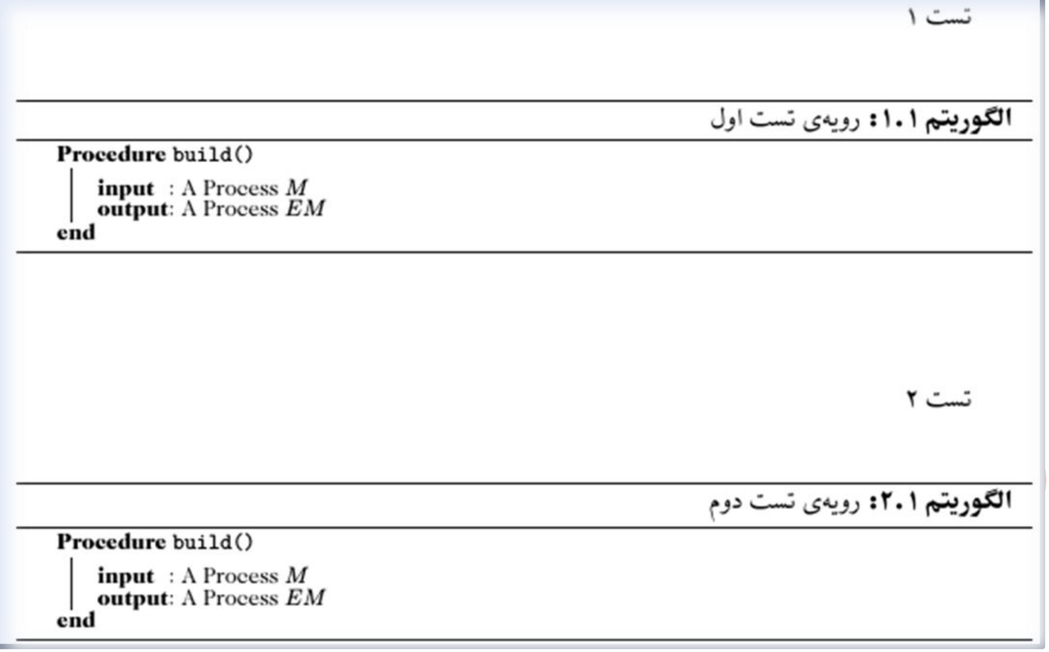
```
\begin{lemma}
\label{lmm1}
فرض کنید  $\cS$یک زیرمجموعه ناتهی از یک فضای خطی جزئاًمرتب  $X$ با مخروط ترتیبی  $\cC$
باشد که  $\cC \neq X$  و $int(\cC) \neq \varnothing$. در این‌صورت هر عنصر
مینیمال از مجموعه  $\cS$یک عنصر مینیمال ضعیف  از مجموعه  $\cS$است.
\end{lemma}
```



لم ۲.۳.۷. فرض کنید $\mathcal{S}$ یک زیرمجموعه ناتهی از یک فضای خطی جزئاًمرتب X با مخروط ترتیبی $\mathcal{C}$ باشد که $\mathcal{C} \neq X$ و $int(\mathcal{C}) \neq \varnothing$. در این‌صورت هر عنصر مینیمال از مجموعه $\mathcal{S}$ یک عنصر مینیمال ضعیف از مجموعه $\mathcal{S}$ است.

Algorithm Environments

```
\usepackage[ruled,algochapter]{algorithm2e}
\usepackage[noend]{algpseudocode}

\newcommand{\listofalgorithmes}{\tocfile{\listalgorithmcfname}{loa}}
\renewcommand*{\listalgorithmcfname}{فهرست الگوریتم‌ها}
\renewcommand*{\algorithmcfname}{الگوریتم}
\renewcommand*{\algorithmautorefname}{الگوریتم}
```



تست ۱

الگوریتم ۱.۱: رویه‌ی تست اول

```
Procedure build()
    input  : A Process M
    output: A Process EM
end
```

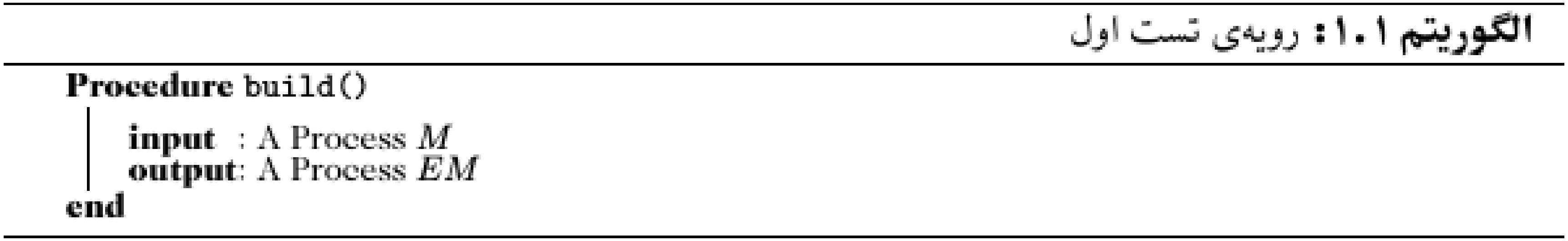
تست ۲

الگوریتم ۲.۱: رویه‌ی تست دوم

```
Procedure build()
    input  : A Process M
    output: A Process EM
end
```

Algorithm Environments

```
\begin{LTR}
\begin{algorithm}[H]
\label{proc:build1}
\setstretch{0.8}
\begin{RTL}
\caption{رویه‌ی تست اول}
\end{RTL}
\SetKwFunction{proc}{build}
\SetKwProg{myproc}{Procedure}{}{end}
\SetKwInOut{Input}{input}
\SetKwInOut{Output}{output}

\myproc{\proc{}}{
\BlankLine
        \Input{A Process $M$}
        \Output{A Process $EM$}
}
\end{algorithm}
\end{LTR}
```



الگوریتم ۱.۱: رویه‌ی تست اول

```
Procedure build()
    input  : A Process $M$
    output: A Process $EM$
end
```

Commutative Diagrams

```
\usepackage{amsmath, amscd}
```

```
$
\begin{CD}
   A        @>a>>       B\\
  @VVbV          @VVcV\\
   C       @>d>>       D
\end{CD}
$
```

$$\begin{tikzcd}
A \arrow[r, "a"] \arrow[d, "b"] & B \arrow[d, "c"] \\
C \arrow[r, "d"] & D
\end{tikzcd}$$

Commutative Diagrams

@<<<	leftarrow
@>>>	rightarrow
@AAA	uparrow
@=	horizontalequals
@VVV	downarrow
@|	verticalequals
@.	emptyarrow

Commutative Diagrams

```
$
\begin{CD}
   A     @<<<     B     @>>>     C\\
   @.             @|             @AAA\\
   @.             D     @=       E
\end{CD}
$
```

$$\begin{tikzcd}
A & B \arrow[l] \arrow[r] \arrow[d, equal] & C \\
 & D \arrow[r, equal] & E \arrow[u]
\end{tikzcd}$$

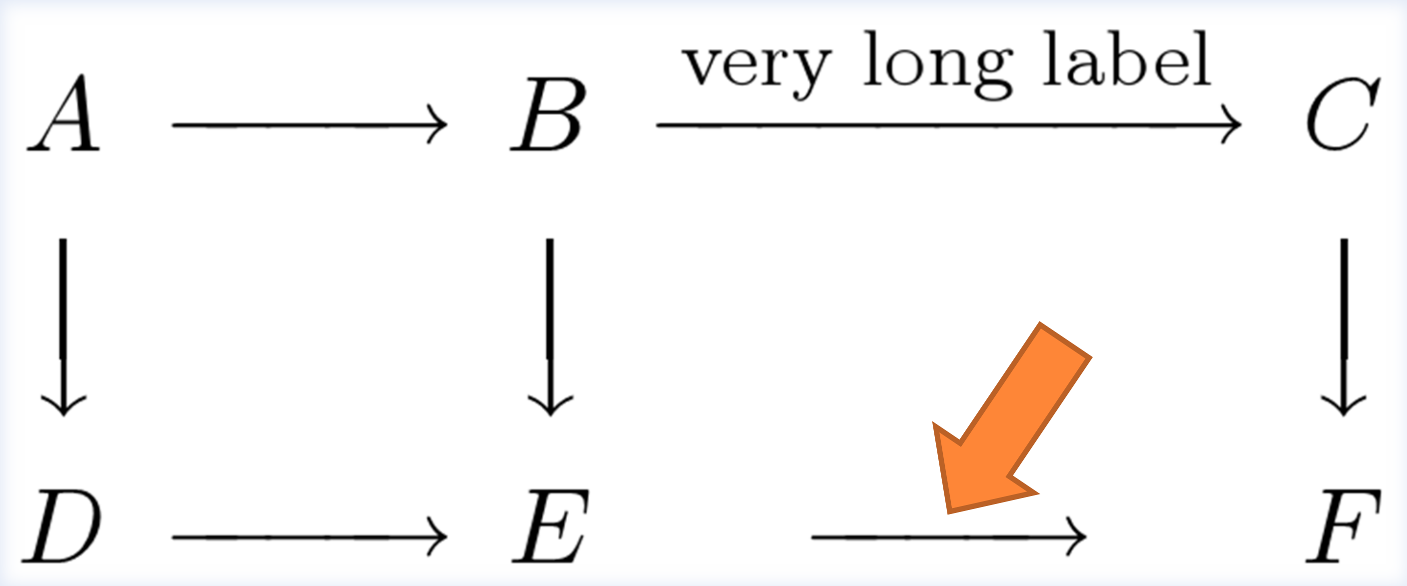
Commutative Diagrams

```
$
\begin{CD}
   A        @>a>b>       B\\
  @VlVrV          @VlVrV\\
   C        @>d>b>       D
\end{CD}
$
```

$$\begin{tikzcd}
A \arrow[r, "a", "b"'] \arrow[d, "l"', "r"] & B \\
C & D \arrow[l, "a"', "b"] \arrow[u, "l", "r"']
\end{tikzcd}$$

Commutative Diagrams

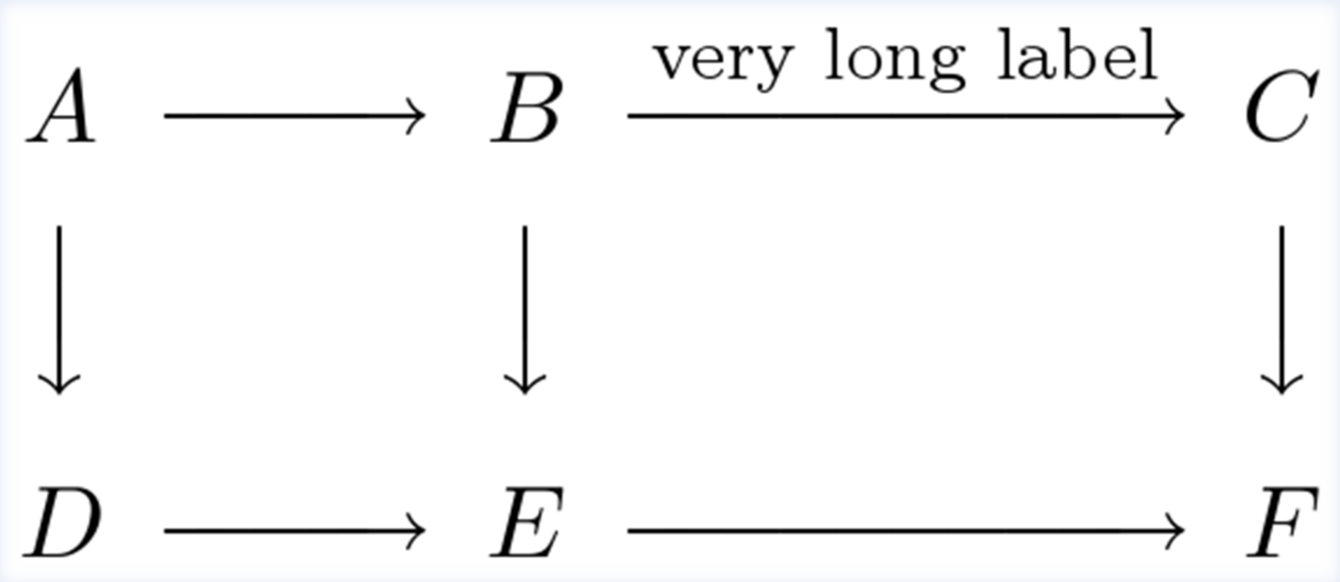
```
$
\begin{CD}
   A     @>>>     B     @>\text{very long label}>>     C\\
   @VVV        @VVV              @VVV\\
   D     @>>>    E     @>>>     F
\end{CD}
$
```



$$\begin{tikzcd}
A \arrow[r] \arrow[d] & B \arrow[r, "\text{very long label}"] \arrow[d] & C \arrow[d] \\
D \arrow[r] & E \arrow[r] & F
\end{tikzcd}$$

Commutative Diagrams

```
$
\begin{CD}
   A     @>>>     B     @>\text{very long label}>>     C\\
   @VVV        @VVV              @VVV\\
   D     @>>>    E     @>{\phantom{\text{very long label}}}>>   F
\end{CD}
$
```



$$\begin{tikzcd}
A \arrow[r] \arrow[d] & B \arrow[r, "\text{very long label}"] \arrow[d] & C \arrow[d] \\
D \arrow[r] & E \arrow[r] & F
\end{tikzcd}$$

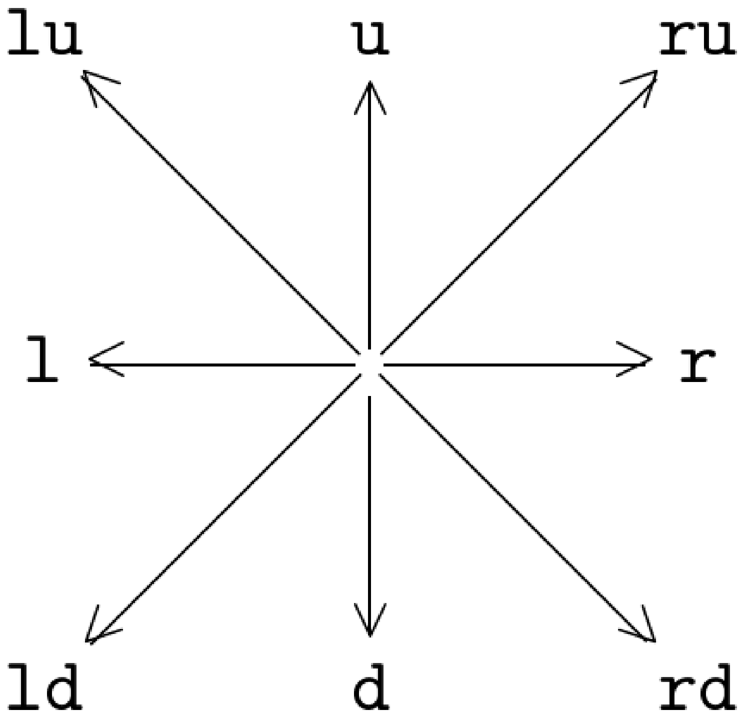
Commutative Diagrams

```
\usepackage[small,nohug,heads=littlevee]{diagrams}
\diagramstyle[labelstyle=\scriptstyle]
```

```
\begin{diagram}
   A            &\rTo^{a}    &B\\
   \dTo_{b}     &            &\dTo_{c} \\
   C            &\rTo^{d}    &D
\end{diagram}
```

$$\begin{tikzcd}
A \arrow[r, "a"] \arrow[d, "b"] & B \arrow[d, "c"] \\
C \arrow[r, "d"] & D
\end{tikzcd}$$

Commutative Diagrams



$\longrightarrow$	To
$\longmapsto$	Mapsto
——	Line
$\hookrightarrow$	Into
$\twoheadrightarrow$	Onto
········>	Dotsto
$\dashrightarrow$	Dashto
$\Longrightarrow$	Implies

Commutative Diagrams

```
\begin{diagram}
   A   &\rInto      &B              &\lInto     &C \\
       &\rdOnto   &\dDotsto    & \ldOnto        \\
       &               &D
\end{diagram}
```

$$\begin{tikzcd}
A \arrow[r, hook] \arrow[dr, two heads] & B \arrow[d, dotted, two heads] & C \arrow[l, hook'] \arrow[dl, two heads] \\
 & D &
\end{tikzcd}$$

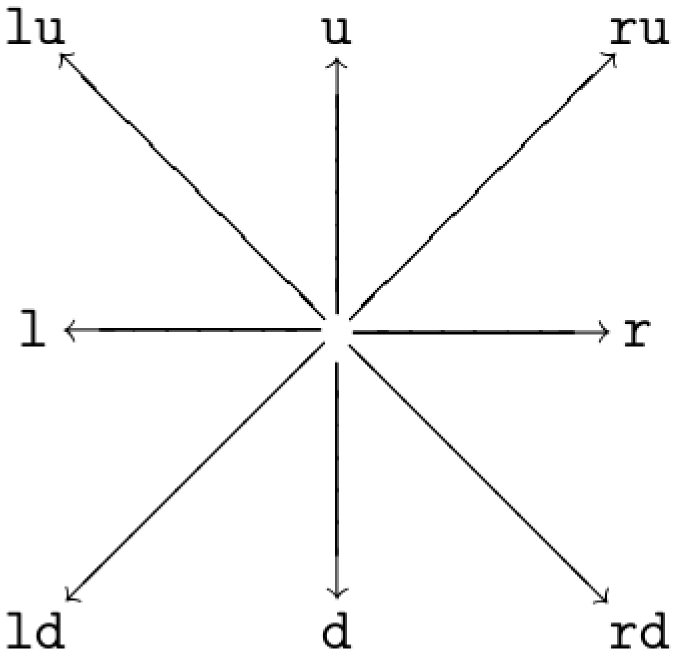
Commutative Diagrams

```
\usepackage[arrsy]{kuvio}
```

```
\Diagram
   A            &\rTo^{a}    &B\\
   \dTo_{b}     &            &\dTo_{c} \\
   C            &\rTo^{d}   &D
\endDiagram
```

$$\begin{tikzcd}
A \arrow[r, "a"] \arrow[d, "b"'] & B \arrow[d, "c"'] \\
C \arrow[r, "d"] & D
\end{tikzcd}$$

Commutative Diagrams



→	To		Nul
	Mapsto	⇒	Two
↪	Into	=	Eq
↠	Epi		Dots
↔	Bij	—	Line

Commutative Diagrams

```
\Diagram
   A   &\rInto       &B                &\lInto     &C \\
       &\rdEpi     &\dDots        & \ldEpi       \\
       &               &D
\endDiagram
```

$$\begin{tikzcd}
A \arrow[r, hook] \arrow[dr, two heads] & B \arrow[d, dotted, no head] & C \arrow[l, hook'] \arrow[dl, two heads] \\
 & D &
\end{tikzcd}$$

Commutative Diagrams

\usepackage{xymatrix}

```
\[
\xymatrix{
   T \ar@/_/[ddr]_y \ar@/^/[drr]^x \ar@{.>}[dr]|-{(x,y)}\\
    &X\times_ZY  \ar[d]^q  \ar[r]_p&X  \ar[d]_f\\
    &Y \ar[r]^g    &Z
}
\]
```

$$\begin{tikzcd}
T \arrow[dr, dotted, "{(x,y)}" description] \arrow[drr, bend left, "x"'] \arrow[ddr, bend right, "y"'] & & \\
& X \times_Z Y \arrow[r, "p"'] \arrow[d, "q"] & X \arrow[d, "f"'] \\
& Y \arrow[r, "g"] & Z
\end{tikzcd}$$

Commutative Diagrams

```
\usepackage{xypic}
\usepackage{xy}
```

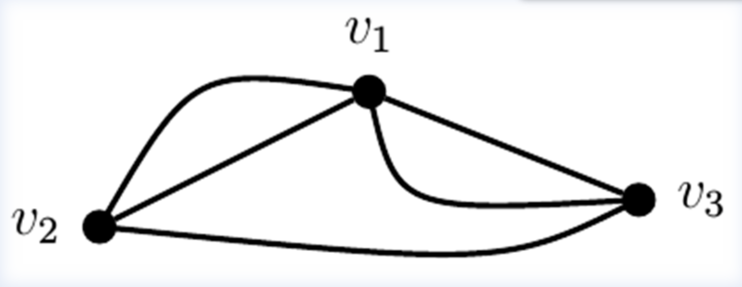
```
\xymatrix{
        A_1 \ar[dr]\ar[drr]\ar[drrr]\ar[drrrr]\ar[drrrrr] &
        A_2\ar[dl] &  A_3 &  A_4  \ar[d]\\
        B_0 & B_1\ar[u]   \ar[ur] & B_2\ar[ul] & B_3 &
        B_4\ar[ul] &  B_5
}
```

$$\begin{tikzcd}
A_1 \arrow[dr] \arrow[drr] \arrow[drrr] \arrow[drrrr] \arrow[drrrrr] & A_2 \arrow[dl] & A_3 & A_4 \arrow[d] & & \\
B_0 & B_1 \arrow[u] \arrow[ur] & B_2 \arrow[ul] & B_3 & B_4 \arrow[ul] & B_5
\end{tikzcd}$$

Graph

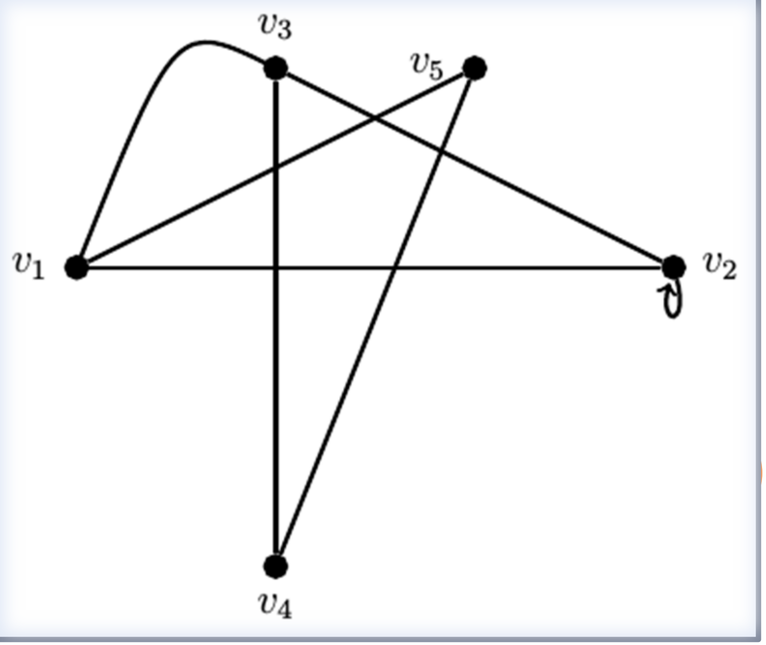
```
\begin{tikzpicture} [scale=1,very thick]
\draw (0,0) .. controls (0.7,1.2) .. (2,1);
\draw (2,1) .. controls (2.2,0.1) .. (4,0.2);
\draw (0,0) .. controls (3,-0.3) .. (4,0.2);
\vertex[fill] (v2) at (0,0) [label=left:$v_{2}$] {};
\vertex[fill] (v1) at (2,1) [label=above:$v_{1}$] {};
\vertex[fill] (v3) at (4,0.2) [label=right:$v_{3}$] {};
\path
(v1) edge (v2)
(v1) edge (v3);
\end{tikzpicture}
```

```
\usepackage{tikz}
\tikzstyle{vertex}=[circle, draw, inner
sep=0pt, minimum size=6pt]
\newcommand{\vertex}{\node[vertex]}
```



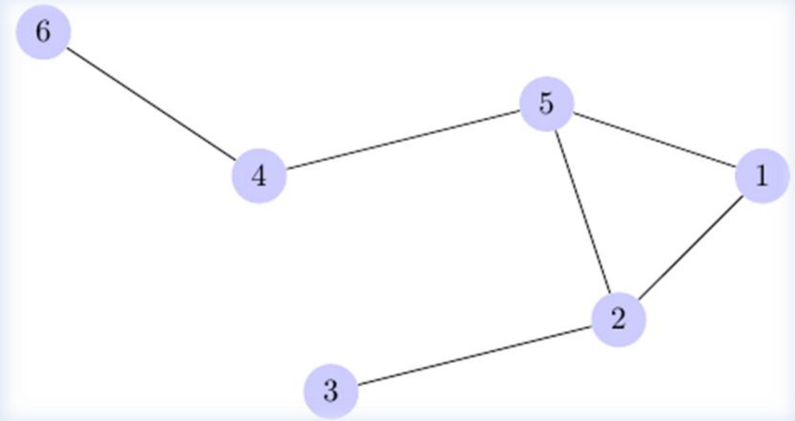
Graph

```
\begin{tikzpicture} [scale=1,very thick]
\draw (0,0) .. controls (1,2.5) .. (2,2);
\vertex[fill] (v1) at (0,0) [label=left:$v_{1}$] {};
\vertex[fill] (v2) at (6,0) [label=right:$v_{2}$] {};
\vertex[fill] (v3) at (2,2) [label=above:$v_{3}$] {};
\vertex[fill] (v4) at (2,-3) [label=below:$v_{4}$] {};
\vertex[fill] (v5) at (4,2) [label=left:$v_{5}$] {};
\path
(v1) edge (v2)
(v1) edge (v5)
(v2) edge (v3)
(v2) edge [loop below]  (v2)
(v4) edge (v5)
(v4) edge (v3);
\end{tikzpicture}
```



Graph

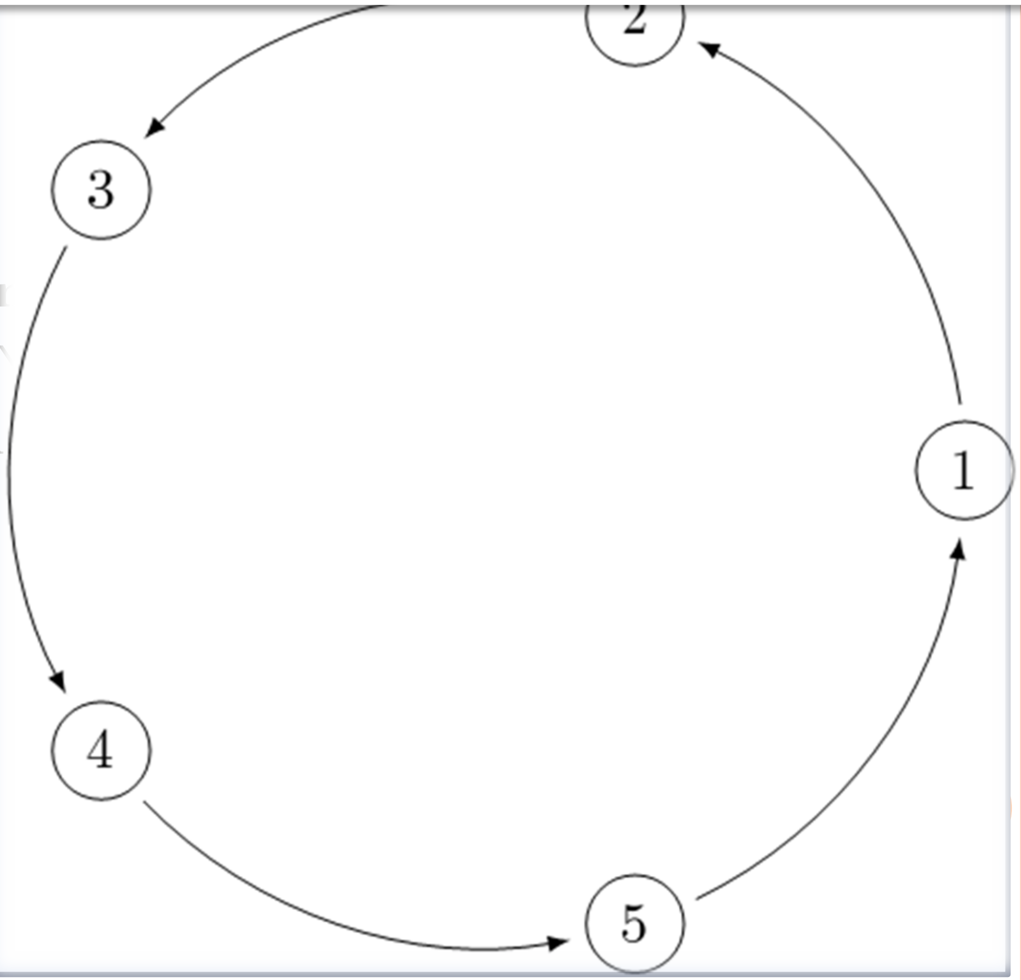
```
\begin{tikzpicture}
  [scale=.8,auto=left,every node/.style={circle,fill=blue!20}]
  \node (n6) at (1,10) {6};
  \node (n4) at (4,8)  {4};
  \node (n5) at (8,9)  {5};
  \node (n1) at (11,8) {1};
  \node (n2) at (9,6)  {2};
  \node (n3) at (5,5)  {3};
  \foreach \from/\to in {n6/n4,n4/n5,n5/n1,n1/n2,n2/n5, n2/n3}
    \draw (\from) -- (\to);
\end{tikzpicture}
```



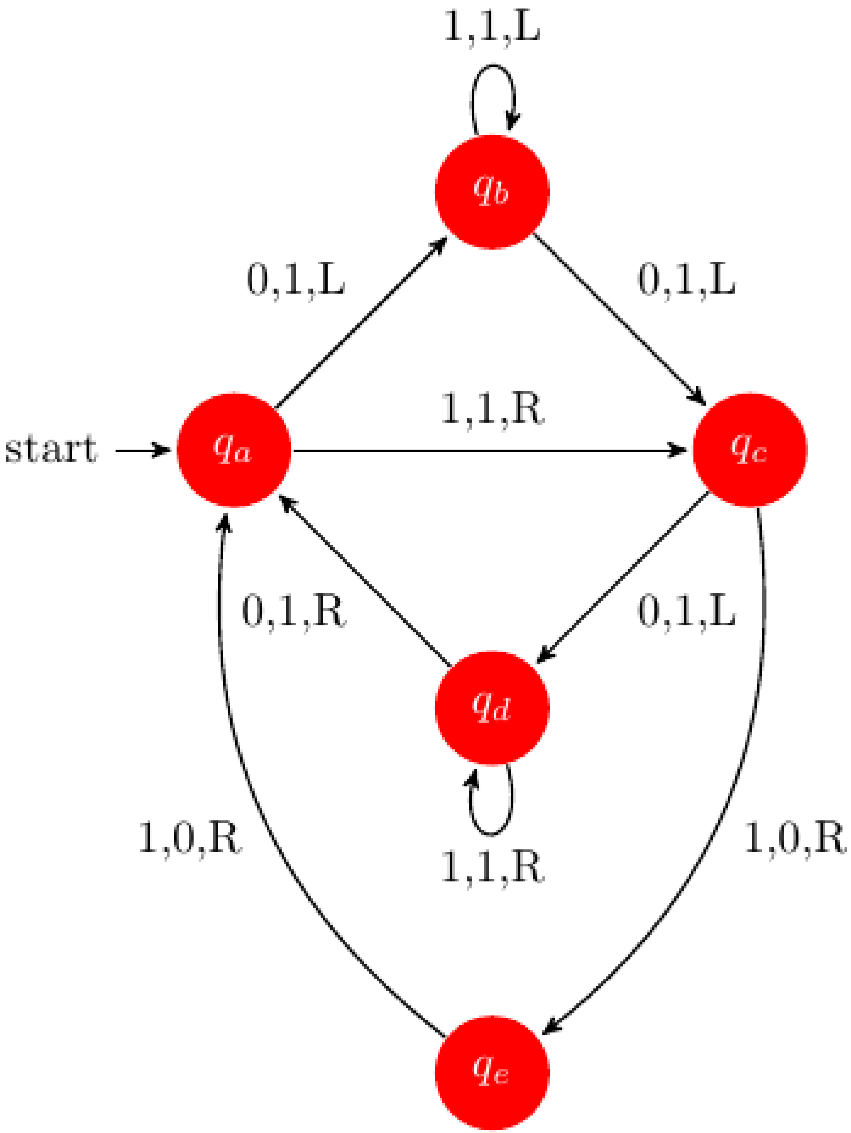
Graph

```
\usepackage{tikz}
\usepackage{verbatim}
\usepackage[active,tightpage]{preview}
\PreviewEnvironment{tikzpicture}
\setlength\PreviewBorder{5pt}
```

```
\begin{tikzpicture}
\def \n {5}
\def \radius {3cm}
\def \margin {8}
\foreach \s in {1,...,\n}
{
 \node[draw, circle] at ({360/\
 \draw[->, >=latex] ({360/\n * (
   arc ({360/\n * (\s - 1)+\margi
}
\end{tikzpicture}
```



Graph (Automata)



```
\usepackage{tikz}
\usetikzlibrary{arrows,automata}
```

Graph

```
\begin{tikzpicture}[->,>=stealth',shorten >=1pt,auto,
                                    node distance=2.8cm, semithick]
 \tikzstyle{every state}=[fill=red,draw=none,text=white]
 \node[initial,state] (A)                    {$q_a$};
 \node[state]         (B) [above right of=A] {$q_b$};
 \node[state]         (D) [below right of=A] {$q_d$};
 \node[state]         (C) [below right of=B] {$q_c$};
 \node[state]         (E) [below of=D]       {$q_e$};
 \path
   (A) edge node {0,1,L} (B)   edge  node {1,1,R} (C)
   (B) edge [loop above] node {1,1,L} (B) edge  node {0,1,L} (C)
   (C) edge node {0,1,L} (D)  edge [bend left]  node {1,0,R} (E)
   (D) edge [loop below] node {1,1,R} (D)  edge  node {0,1,R} (A)
   (E) edge [bend left]  node {1,0,R} (A);
\end{tikzpicture}
```

The END

Thank you!