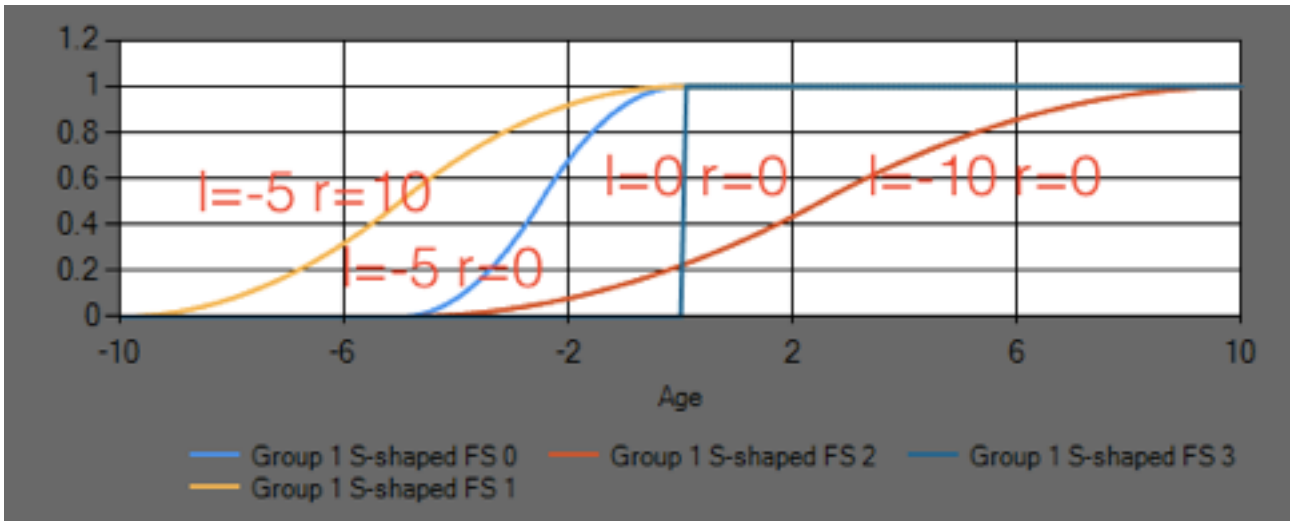
6.(a)

```
    static int count = 0;
    public SshapeFuzzySet(Universe u) : base(u)
    {
        parameters = new double[2];
        parameters[0] = -5.0; // l
        parameters[1]  =.0; //r
        parNames = new string[2];
        parNames[0] = "l";
        parNames[1] = "r";
        type =   " S-shaped FS " + count.ToString();
        NameTitle();
        count++;
        updateDataPoints();
    }
    public override double GetMembershipDegree(double x)
    {
        double y = 0.0;
        if (x <= parameters[0])  //  x <=l
            y = 0.0;
        else  if(x <= ((parameters[0]+parameters[1])/2))  //x <=(l+r)/2
            y =2* (x - parameters[0]) *(x-parameters[0])/  (parameters[1]  - parameters[0])/
(parameters[1]  - parameters[0]); //2*((x-l)/(r-l))^2
        else if(x <= parameters[1])  //x <=r
            y =1- 2*(parameters[1] - x)*  (parameters[1] - x)/ (parameters[1] - parameters[0])/
(parameters[1] - parameters[0]); //1-2*((r-x)/(r-l))^2
        else //x >r
            y= 1;  //y=l
        return y;
    }
}
```

(b)



8.(a)

```
  class PishapeFuzzySet : FuzzySet
{
    static int count = 0;
```

```
        public PishapeFuzzySet(Universe u) : base(u)
        {
            parameters = new double[2];
            parameters[0] = -5.0; // a
            parameters[1] = 0.0; //c
            parNames = new string[2];
            parNames[0] = "a";
            parNames[1] = "c";
            type = " Pi-shaped FS " + count.ToString();
            NameTitle();
            count++;
            updateDataPoints();
        }
        public override double GetMembershipDegree(double x)
        {
            double y = 0.0;
            if (x <= parameters[1]) //s-shaped
            {
                double l = parameters[1] - parameters[0]; //c-a
                double r = parameters[1]; //c
                if (x <= l)
                    y = 0.0;
                else if (x <=( (l+r)/ 2))
                    y = 2 * (x - l) * (x - l) / (r - l) / (r - l);
                else  if (x <= r)
                    y = 1 - 2 * (r - x) * (r- x) / (r - l) / (r - l);
                else
                    y = 1.0;
            }
            else if (x > parameters[1]) //z-shaped
            {
                double l = parameters[1]; //c
                double r = parameters[1] + parameters[0]; //c+a
                if (x <= l)
                    y = 1;
                else if (x <= ((l + r)) / 2)
                    y = 1-2 * (x - l) * (x - l) / (r - l) / (r - l);
                else if (x <= r)
                    y = 2 * (r - x) * (r - x) / (r - l) / (r - l);
                else
                    y = 0.0;
            }
            return y;
        }
    }
```

(b)