

Chapter 3

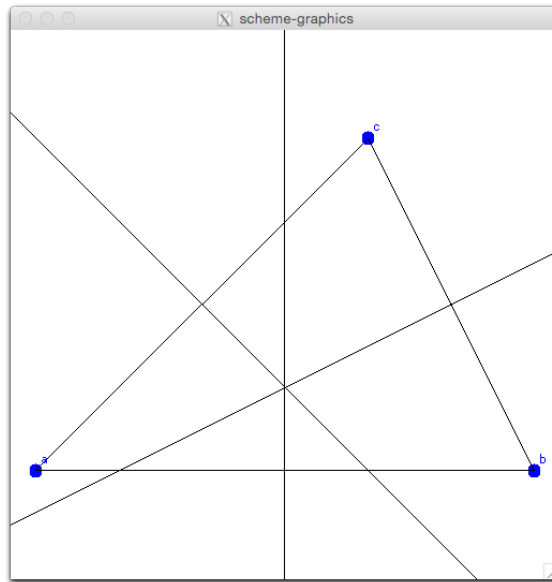
Demonstration

My system uses this idea of manipulating diagrams “in the mind’s eye” to explore and discover geometry theorems. Before discussing some of the internal representations and modules, I will briefly describe the goals of the system to provide direction and context to understand the components.

3.1 Imperative Figure Construction

```
1 (define (triangle-with-pep-bisectors)
2   (let-geo* ((a (make-point 0 0))
3             (b (make-point 1.5 0))
4             (c (make-point 1 1))
5             (t (polygon-from-points a b c))
6             (pb1 (perpendicular-bisector (make-segment a b)))
7             (pb2 (perpendicular-bisector (make-segment b c)))
8             (pb3 (perpendicular-bisector (make-segment c a))))
9   (figure t pb1 pb2 pb3)))
```

```
=> (show-figure (triangle-with-perp-bisectors))
```



```
=> (show-figure (triangle-with-perp-bisectors))
```

```
((concurrent #[line 22] #[line 20] #[line 18])  
 (perpendicular #[line 22] #[segment 21])  
 (perpendicular #[line 20] #[segment 19])  
 (perpendicular #[line 18] #[segment 17]))
```

3.2 Declarative Constraint Solving

Listing 3.1: Getting labels

```
1 (define (arbitrary-triangle)  
2   (m:mechanism  
3     (m:establish-polygon-topology 'a 'b 'c)))
```

Listing 3.2: Constraint Solving for Isocles Triangle

```
1 (define (isocles-triangle)  
2   (m:mechanism
```

```

3 (m:establish-polygon-topology 'a 'b 'c)
4 (m:c-length-equal (m:bar 'a 'b)
5                   (m:bar 'b 'c))))

```

Listing 3.3: Constraint Solving for Isoceles Triangle

```

1 (define (parallelogram-by-angles)
2 (m:mechanism
3 (m:establish-polygon-topology 'a 'b 'c 'd)
4 (m:c-angle-equal (m:joint 'a)
5                 (m:joint 'c))
6 (m:c-angle-equal (m:joint 'b)
7                 (m:joint 'd))))

```