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

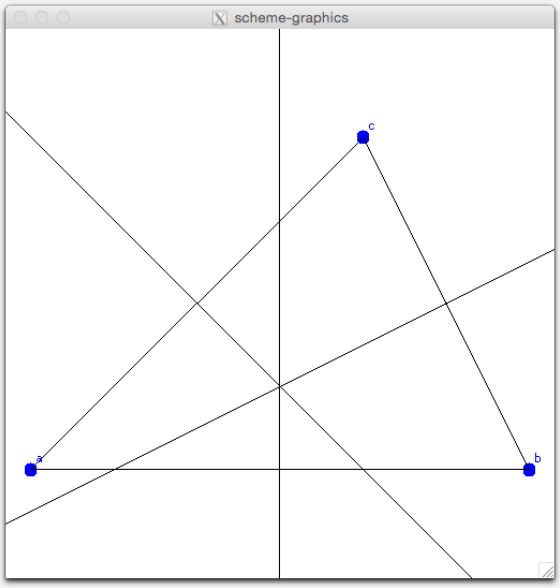
Code Example 3.1: Basic Example

```
(define (triangle-with-pep-bisectors)
  (let-geo* ((a (make-point 0 0))
             (b (make-point 1.5 0))
             (c (make-point 1 1))
             (t (polygon-from-points a b c))
             (pb1 (perpendicular-bisector (make-segment a b)))
             (pb2 (perpendicular-bisector (make-segment b c)))
             (pb3 (perpendicular-bisector (make-segment c a))))
    (figure t pb1 pb2 pb3)))

(define (triangle-with-pep-bisectors)
  (let-geo* ((a (make-point 0 0))
             (b (make-point 1.5 0))
             (c (make-point 1 1))
             (t (polygon-from-points a b c))
             (pb1 (perpendicular-bisector (make-segment a b)))
             (pb2 (perpendicular-bisector (make-segment b c)))
             (pb3 (perpendicular-bisector (make-segment c a))))
    (figure t pb1 pb2 pb3)))
```

Graphics Example 3.2: Basic Example Image

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



Interaction Example 3.3: Simple Analysis

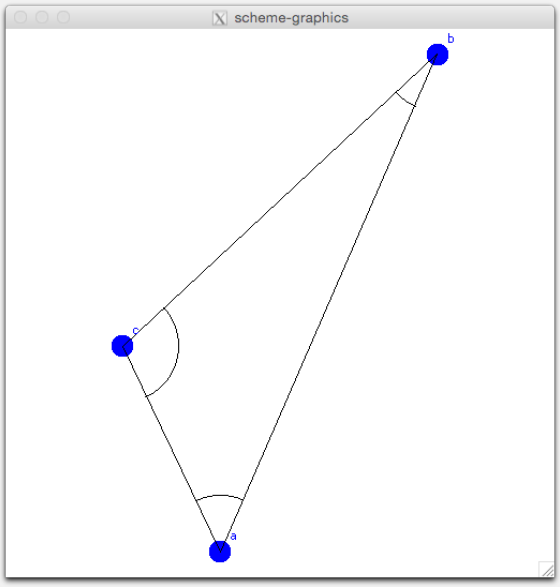
```
=> (show-figure (triangle-with-perp-bisectors))

((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

Graphics Example 3.4: Arbitrary Triangle

```
Text Goes Here
```


scheme-graphics
b
c
a