lib/basic/sets_unittest.ath

```
load "sets"
load "testing"

open Set
open Testing

(Testing   "(eval 23 in [1 5 23 98])"
  lambda () (eval 23 in [1 5 23 98])
  [expecting
   true
   ])

(Testing   "(eval 23 in [1 5 98])"
  lambda () (eval 23 in [1 5 98])
  [expecting
   false
   ])

(Testing   "(eval 5 in [])"
  lambda () (eval 5 in [])
  [expecting
   false
   ])

(Testing   "(eval 5 in [5])"
  lambda () (eval 5 in [5])
  [expecting
   true
])

(Testing   "(eval [1 2] subset [3 2 4 1 5])"
  lambda () (eval [1 2] subset [3 2 4 1 5])
  [expecting
   true
   ])

(Testing   "(eval [1 2] subset [3 2])"
  lambda () (eval [1 2] subset [3 2])
  [expecting
   false
   ])

(Testing   "(eval [] subset [])"
  lambda () (eval [] subset [])
  [expecting
   true
   ])

(Testing   "(eval 1 ++ 2 ++ [] = 2 ++ 1 ++ [])"
  lambda () (eval 1 ++ 2 ++ [] = 2 ++ 1 ++ [])
  [expecting
   true
   ])

(Testing   "(eval 1 ++ 2 ++ 3 ++ 4 ++ [] = 3 ++ 2 ++ 1 ++ [])"
  lambda () (eval 1 ++ 2 ++ 3 ++ 4 ++ [] = 3 ++ 2 ++ 1 ++ [])
  [expecting
   false
   ])

(Testing   "(eval [1 2] proper-subset [2 3 1])"
  lambda () (eval [1 2] proper-subset [2 3 1])
  [expecting
   true
   ])

(Testing   "(eval [1 2] proper-subset [2 1])"
  lambda () (eval [1 2] proper-subset [2 1])
```

```
69    [expecting
70     false
71     ])
72
73  (Testing   "(eval [1 2 3 2 5] - 2)"
74    lambda () (eval [1 2 3 2 5] - 2)
75    [expecting
76     [1 3 5]
77     ])
78
79  (Testing   "(eval [1 2 3] \\/ [4 5 6])"
80    lambda () (eval [1 2 3] \/ [4 5 6])
81    [expecting
82     [1 2 3 4 5 6]
83     ])
84
85  (Testing   "(eval [1 2] \\/ [1 2])"
86    lambda () (eval [1 2] \/ [1 2])
87    [expecting
88     [1 2]
89     ])
90
91  (Testing   "(eval [1 2 1] /\\ [5 1 3])"
92    lambda () (eval [1 2 1] /\ [5 1 3])
93    [expecting
94     [1]
95     ])
96
97  (Testing   "(eval [1 2 1] /\\ [5])"
98    lambda () (eval [1 2 1] /\ [5])
99    [expecting
100    []
101    ])
102
103 (Testing   "(eval 3 paired-with [2 8])"
104   lambda () (eval 3 paired-with [2 8])
105   [expecting
106    [(pair 3 2) (pair 3 8)]
107    ])
108
109 (Testing   "(eval [1 2] X ['foo 'bar 'car])"
110   lambda () (eval [1 2] X ['foo 'bar 'car])
111   [expecting
112    [(pair 1 'foo) (pair 1 'bar) (pair 1 'car)
113     (pair 2 'foo) (pair 2 'bar) (pair 2 'car)]
114    ])
115
116 (Testing   "(eval dom [('a @ 1) ('b @ 2) ('c @ 98)])"
117   lambda () (eval dom [('a @ 1) ('b @ 2) ('c @ 98)])
118   [expecting
119    ['a 'b 'c]
120    ])
121
122 (Testing   "(eval range [('a @ 1) ('b @ 2) ('c @ 98)])"
123   lambda () (eval range [('a @ 1) ('b @ 2) ('c @ 98)])
124   [expecting
125    [1 2 98]
126    ])
127
128 (Testing   "(eval 1 @ 2 composed-with [(2 @ 5) (7 @ 8) (2 @ 3)])"
129   lambda () (eval 1 @ 2 composed-with [(2 @ 5) (7 @ 8) (2 @ 3)])
130   [expecting
131    [(pair 1 5) (pair 1 3)]
132    ])
133
134 (Testing   "(eval 1 @ 2 composed-with [(7 @ 8) (9 @ 10)])"
135   lambda () (eval 1 @ 2 composed-with [(7 @ 8) (9 @ 10)])
136   [expecting
137    []
138    ])
```

```
139
140 (Testing   "(eval 1 @ 2 composed-with [])"
141   lambda () (eval 1 @ 2 composed-with [])
142   [expecting
143    []
144    ])
145
146 (Testing   "(eval [('nyc @ 'boston) ('houston @ 'dallas) ('austin @ 'dc)] o
147                  [('boston @ 'montreal) ('dallas @ 'chicago) ('dc @ 'nyc)] o
148                  [('chicago @ 'seattle) ('montreal @ 'london)])"
149   lambda () (eval [('nyc @ 'boston) ('houston @ 'dallas) ('austin @ 'dc)] o
150                  [('boston @ 'montreal) ('dallas @ 'chicago) ('dc @ 'nyc)] o
151                  [('chicago @ 'seattle) ('montreal @ 'london)])
152   [expecting
153    [(pair 'nyc 'london) (pair 'houston 'seattle)]
154    ])
155
156 (Testing   "let {R1 := [('nyc @ 'boston) ('austin @ 'dc)];
157                 R2 := [('boston @ 'montreal) ('dc @ 'chicago) ('chicago @ 'seattle)]}
158               (eval R1 o R2)"
159   lambda () let {R1 := [('nyc @ 'boston) ('austin @ 'dc)];
160                 R2 := [('boston @ 'montreal) ('dc @ 'chicago) ('chicago @ 'seattle)]}
161               (eval R1 o R2)
162   [expecting
163    [(pair 'nyc 'montreal) (pair 'austin 'chicago)]
164    ])
165
166 (Testing   "(eval [(1 @ 'foo) (2 @ 'b) (1 @ 'bar)] restrict1 1)"
167   lambda () (eval [(1 @ 'foo) (2 @ 'b) (1 @ 'bar)] restrict1 1)
168   [expecting
169    [(pair 1 'foo) (pair 1 'bar)]
170    ])
171
172 (Testing   "(eval [(1 @ 'foo) (2 @ 'b) (3 @ 'c) (4 @ 'd) (1 @ 'bar)] ^ [1 2])"
173   lambda () (eval [(1 @ 'foo) (2 @ 'b) (3 @ 'c) (4 @ 'd) (1 @ 'bar)] ^ [1 2])
174   [expecting
175    [(pair 1 'foo) (pair 1 'bar) (pair 2 'b)]
176    ])
177
178 (Testing   "(eval [(1 @ 'a) (2 @ 'b) (3 @ 'c)] ** [1 3])"
179   lambda () (eval [(1 @ 'a) (2 @ 'b) (3 @ 'c)] ** [1 3])
180   [expecting
181    ['a 'c]
182    ])
183
184 (Testing   "(eval card [1 2 3] \\/ [4 7 8])"
185   lambda () (eval card [1 2 3] \/ [4 7 8])
186   [expecting
187    6
188    ])
```