

```

1 #lang racket
2
3 (struct empty-set () #:transparent)
4 (struct singleton (v) #:transparent)
5 (struct union (s1 s2) #:transparent)
6
7 (define empty (empty-set))
8 (define just1 (singleton 1))
9
10 (define (member? s a)
11   (match s
12     ((empty-set) #f)
13     ((singleton v) (if (equal? v a) #t #f))
14     ((union x y) (or (member? x a) (member? y
14 a))))))
15
16 (member? just1 2)
17
18 (define abc (union (singleton "a") (union
18 (singleton "b") (singleton "c"))))
19 abc
20
21 (define (make-union s1 s2)
22   (match s1
23     ((empty-set) s2)
24     ((singleton v) (if (member? s2 v) s2 (union
24 s1 s2))))
25     ((union (singleton x) y) (if (member? s2 x)
26                                     (make-union y
26 s2)
27                                     (union
27 (singleton x) (make-union y s2))))))
28
29 (make-union abc abc)
30

```

```
31 | (define (make-set args)
32 |   (match args
33 |     ((list) (empty-set))
34 |     ((list v) (singleton v))
35 |     ((list s1 s2) (make-union s1 s2))))
36 |
```