DINING PHILOSOPHERS IN ANST ACTOR LANGUAGE

B_{phil} = rec(λb.λl.λr.λself. λacks. λm.

if (eq?(m, self),

if (eq?(acks, 0),

become (b(l, r, self, acks + 1)),

seq(send(l, mkrelease(self)),

send(r, mkrelease(self)),

become(b(l, r, self, 0)),

send(l, mkpickup(self)),

send(r, mkpickup(self)))),

become (b(l, r, self, acks))))

DINING PHILOSOPHERS IN AMST (2)

```
Bfork = rec (λb. λh. λw. λm.
            if (pickup?(m),
                if (eq?(h, nil),
                    seq(send(phil(m), phil(m)),
                        become(b(phil(m), nil))),
                    become(b(h, phil(m)))),
                if (release?(m),
                    if (eq?(w, nil),
                        become(b(nil, nil)),
                        seq(send(w, w),
                            become(b(w, nil)))),
                    become(b(h, w)))))
```

DINING PHILOSOPHERS IN AMST (3)

Using the definitions to set up a 2-phils dining table:


```
letactor { F1 := Bfork(nil, nil),
           F2 := Bfork(nil, nil),
           P1 := Bphil(F1, F2, P1, ∅),
           P2 := Bphil(F2, F1, P2, ∅) } e
```

where e is defined as:

```
e = seq(send(F1, mkpickup(P1)),
        send(F2, mkpickup(P1)),
        send(F1, mkpickup(P2)),
        send(F2, mkpickup(P2)))
```

DINING PHILOSOPHERS IN AMST (4)

Auxiliary definitions:

```
mkPickup = λp. pr("pickup", p)
mkrelease = λp. pr("release", p)
pickup? = λm. if (ispr?(m),
                  eq?(1st(m), "pickup"),
                  nil)
release? = λm. if (ispr?(m),
                  eq?(1st(m), "release"),
                  nil)
phil = λm. if(pickup?(m),
              2nd(m),
              nil)
```

Reference Cell in Actor Language

$$B_{cell} = rec(\lambda b.\lambda c.\lambda m.$$
$$\quad if\ (get?(m),$$
$$\quad\quad seq\ (become\ (b(c)),$$
$$\quad\quad\quad send\ (cust(m), c))$$
$$\quad\quad if\ (set?(m),$$
$$\quad\quad\quad become\ (b(contents(m))),$$
$$\quad\quad\quad become\ (b(c))))$$

Using the cell:

$$letactor\ \{a := B_{cell}(0)\}\ e$$

$$e = seq(send(a, mkset(3)),$$
$$\quad send(a, mkset(5)),$$
$$\quad send(a, mkget(e)))$$