Theory Assignment 2

Caden Keese

Methods:

- inc - increment
- dec - decrement
- rd - read if positive, negative or zero

States:

- Initial State $n = 0, n \in \mathbb{Z}$,

Rules:

inc:

- $n \xrightarrow{inc(i)} n + 1$

dec:

- $n \xrightarrow{dec(i)} n - 1$

rd:

- $n \xrightarrow{rd(p)} n \text{ if } n > 0$
- $n \xrightarrow{rd(n)} n \text{ if } n < 0$
- $n \xrightarrow{rd(z)} n \text{ if } n = 0$

Pseudo Code:

Basic Structure

```
R[n] = [Register_1, ..., Register_n];
```

inc(i):

i is the current thread number

```
inc(i):
 R[i] = R[i] + 1
```

inc(i):

i is the current thread number

```
dec(i):
  R[i] = R[i] - 1
```

rd():

```
rd():
1:  values[i] = {0,...,0}
2:  for (i:=0; i < n; i++)
3:      values = R[i]
4:  sum := 0
5:  for (i := 0; i < n; i++)
6:      sum = sum + values[i]
7:  if sum > 0
8:      return p
9:  if sum < 0
10:   return n
11: else
12:     return z
```

The linearization policy for `inc(i)` and `dec(i)` are that the linearization is at the point of the modification of `R[i]` . The linearization of `rd()` occurs at the point the registers are read, lines 2 & 3.
this works as each register is atomic, so once a modification occurs (the linearization of `inc(i)` and `dec(i)`) it is not possible to read the previous value and therefore have a conflict.