The Process Network Domain

Thomas M. Parks

March 10, 1995

University of California, Berkeley

Ptolemy Conference

parks@eecs.berkeley.edu

http://www.eecs.berkeley.edu/∼parks

1 Class DataFlowProcess

```
void DataFlowProcess::run()
{
    // Configure the star for dynamic execution.
    star.setDynamicExecution(TRUE);
    // Fire the Star ad infinitum.
    do
    {
        if (star.waitPort())
            star.waitPort()->receiveData();
    } while(star.run());
}
```


2 Class PNGeodesic

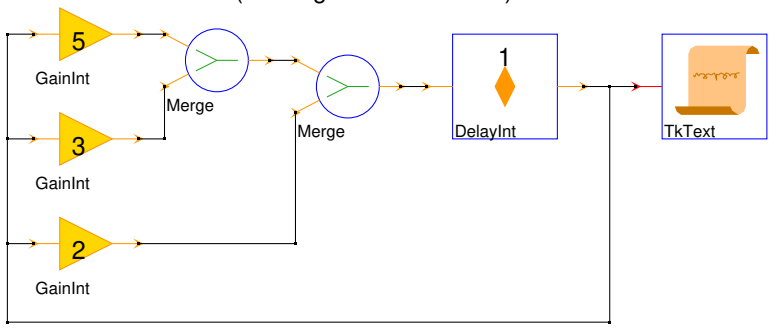
```
// Block until not empty.
Particle* PNGeodesic::slowGet()
{
    if (sz < 1 && notEmpty) notEmpty->wait();
    if (sz > 0)
    {
        sz--; return pstack.get();
    }
    else return 0;
}
```



```
// Notify when not empty.
void PNGeodesic::slowPut(Particle* p)
{
    pstack.putTail(p); sz++;
    if (notEmpty) notEmpty->notifyAll();
}
```

3 $2^A3^B5^C$

Produce a sequence of integers
of the form 2^a 3^b 5^c
(from figure 4 in Kahn77)

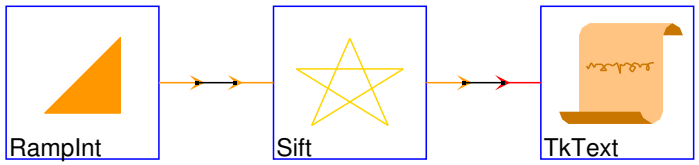


3.1 Merge

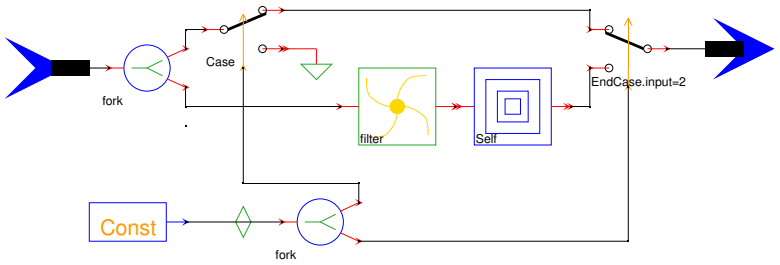
```
go
{
    a.receiveData();
    b.receiveData();
    for (;;) // Do forever.
    {
        if (a%0 < b%0)
        {
            output%0 = a%0;
            a.receiveData();
        }
        else if (a%0 > b%0)
        {
            output%0 = b%0;
            b.receiveData();
        }
        else        // Remove duplicates.
        {
            output%0 = a%0;
            a.receiveData();
            b.receiveData();
        }
        output.sendData();
    }
}
```


4 The Sieve of Eratosthenes

Sieve of Eratosthenes
(from figure 3 in Kahn77,
with optimized Sift process)



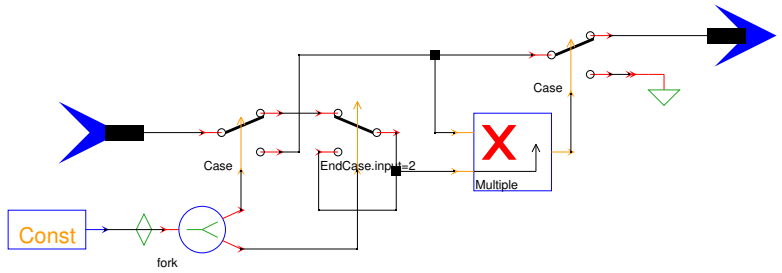
4.1 Sift



```
go
{
    StringList prime((input%0).print());
    output%0 = input%0; // Emit a discovered prime.

    DataFlowStar* star =
        splice("Filter","input","output",&input);
    star->setState("prime", prime);
    star->initState();
    newProcess(star);
}
```


4.2 Filter




```
go
{
    int N = input%0;
    int P = prime;

    if (N%P)
    {
        output.reset();
        output%0 = input%0;
        output.sendData();
    }
}
```


5 Future Work

- Clustering of SDF and BDF subsystems.
- Compiler techniques for threaded stars (Reactive C).
- Port to POSIX threads for multi-processor workstations.