嵌入式系统理论作业

曹广杰

15352015 数据科学与计算机

授课教师：黄凯

2017/10/27

1.1.1

Algorithm 1

```
if L1[X], L2[Y] then
del(X), del(Y), out[X, Y]
else if L1[X], L2[φ] then
del(X), out[X]
else if L1[φ], L2[Y] then
del(Y), out[Y]
else if L1[φ], L2[φ] then
no operation
end if
```

Algorithm 2

```
if L1[X] = L2[Y] then
del(X), del(Y), out[X, Y]
else if L1[X] < L2[Y] then
del(X), out[X]
else if L1[X] > L2[Y] then
del(Y), out[Y]
end if
```

1. Examine the determinacy and the fairness of these two algorithms. Prove and disprove your conclusion.

   算法A不确定，但是公平。

   因为：

   - 算法A的输出序列会受到输入序列顺序的影响
   - 算法A的输出结果始终遵循先到先服务（FCFS）的原则

   设若已有两个输入序列：

$$X_1 = ([x_1, x_2])$$
$$X_2 = ([x_1, x_2], [y_1])$$

   对应的输出序列：

$$F[X_1] = ([x_1, x_2])$$
$$F[X_2] = ([x_1, x_2, y_1])$$

算法B确定，但是不公平：

算法B对于任意两个输入序列，输出都是升序排列的——“`know that each channel produces increasing sequences of serial numbers (with unknown gaps though)`”，因此确定。

算法B总是会输出长度较短的序列，这就意味着，最后我们会剩下一个序列不能处理。

2.1.2

Draw a Kahn process network that can generate the sequence of quadratic numbers n(n+1)/2. Use basic processes that add two numbers, multiply two numbers, or duplicate a number. You can also use initialization processes that generate a constant and then simply forward their input. Finally, you can use a sink process.

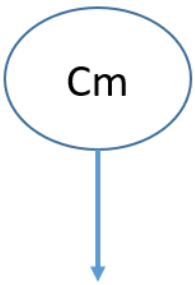
因为没有除法，在处理这种问题的时候，只能使用把等差数列累加的函数转化为累加的形式：

$$
\begin{aligned}
f(n) &= n(n+1)/2 \\
&= \sum_{i=0}^{n} i
\end{aligned}
$$

首先实现一个递增的数列，即n的序列，没有n就无法进行之后的乘除运算。对于该系统：

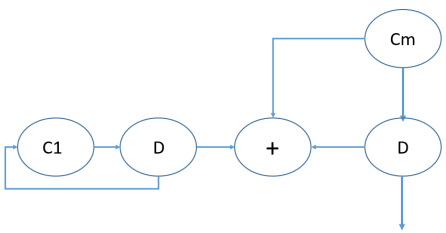
1. 不断地输出数字，每一个输出都比上一次的输出多一个
2. 输出从1开始，上不封顶

为了输出从1开始而且递增，我们需要使用constant module，但是由于每次输出的数据受限于输入的数据，所以该module不能设置为一个常数输出，准确地说，应该是一个输出相当于输入的模块。



因为有初始化的权限，所以先静这个模块初始化为1，则其输出的时候，就会是1。此后，需要输出在原有的基础上多一个的数字。

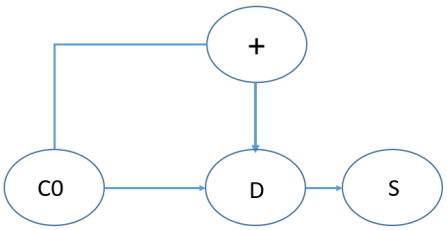
- 是的，多1。为此，需要再一次使用constant模块，以实现对于常数值1的参数设置。
- 由于需要保留每一次的输出信息，所以需要duplicate模块——一份输出，一份加一



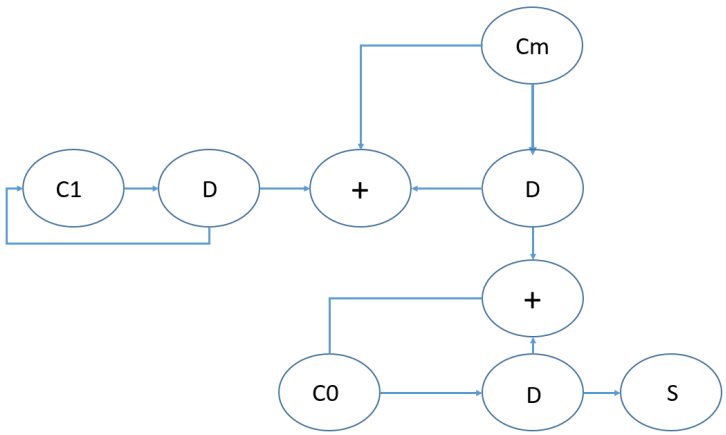
如此，我们就已经实现了源源不断地输出自然数序列的模块。接下来就是实现累加以避免使用除法：

$$f(n) = n + f(n-1)$$

上文已经实现了**n**，接下来就是要把之前输出的数据累加并保存起来



每次获取一个数字，并使用寄存器将其储存起来，一个数字两种用途——将以上两个模块拼接即可得到我们需要的累加器：



2.2 Model of Computation - Synchronous Data Flow

2.2.1 Given the SDF graph in Figure.4

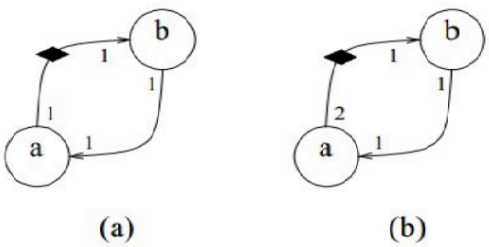


Figure 4:

- Determine the topological matrix of these two SDF graphs
- Are these two graphs consistent?
- If yes, determine the number of firings of each node, which leads for a periodic execution

1. 分析(a)图：

   由图中的入度出度可以得到以下方程组：

   $$\begin{cases} a - b &= 0 \\ b - a &= 0 \end{cases}$$

   对应的，拓扑矩阵应为：

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

2. 分析(b)图：

由图中的入度出度可以得到以下方程组：

$$\begin{cases} 2a - b = & 0 \\ b - a = & 0 \end{cases}$$

对应的，拓扑矩阵应为：

$$\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

左图是consistent（一致的）：

- 因为矩阵的rank的值为1，该矩阵线性独立的纵列的极大值(即化简之后的行数)，恰好是进程数目减一；

右图是inconsistent（不一致的）：

- 因为矩阵的rank为2，是进程的数目，不能求得唯一解；

左图激活节点的最少数量为a或者b，只要有一个即可。

2.2.2 Given SDF graph in Firgure.5

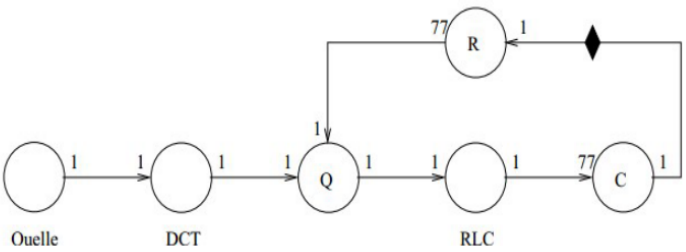


Figure 5

确定拓扑矩阵：

$$\begin{cases} Ouelle - DCT = & 0 \\ DCT - Q = & 0 \\ Q - RLC = & 0 \\ RLC - C = & 0 \\ C - R = & 0 \\ 77R - Q = & 0 \end{cases}$$

综上，可以确定构成的拓扑矩阵维度为6：

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -77 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 0 & 0 & -77 \end{bmatrix}$$

将该矩阵化简为下三角矩阵得：

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -77 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

化简之后的行数为5，而进程数量为6——这很显然，该系统是consistent。若要使得该系统可以运作，激活的节点：

$$\begin{aligned} uelle &=> 1 \\ DCT &=> 1 \\ Q &=> 1 \\ RLC &=> 1 \\ C &=> 1 \\ R &=> 77 \end{aligned}$$