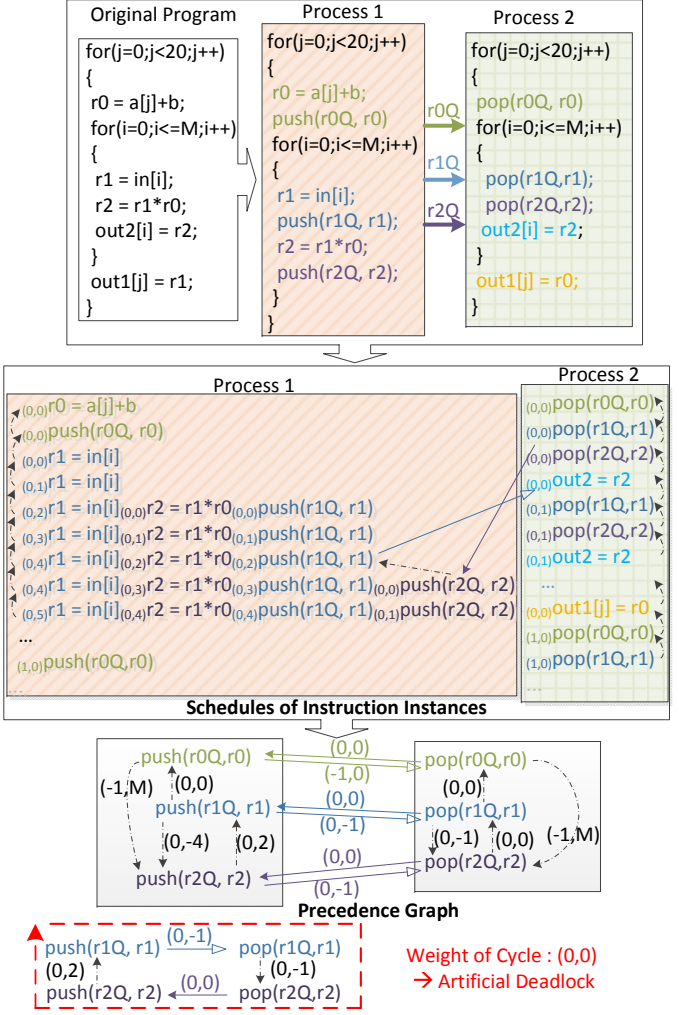
Original Program
for(j=0;j<20;j++)
{
r0 = a[j]+b;
for(i=0;i<=M;i++)
{
r1 = in[i];
r2 = r1*r0;
out2[i] = r2;
}
out1[j] = r1;
}
Process 1
for(j=0;j<20;j++)
{
r0 = a[j]+b;
push(r0Q, r0)
for(i=0;i<=M;i++)
{
r1 = in[i];
push(r1Q, r1);
r2 = r1*r0;
push(r2Q, r2);
}
}
r0Q
r1Q
r2Q
Process 2
for(j=0;j<20;j++)
{
pop(r0Q, r0)
for(i=0;i<=M;i++)
{
pop(r1Q,r1);
pop(r2Q,r2);
out2[i] = r2;
}
out1[j] = r0;
}
Process 1
(0,0)r0 = a[j]+b
(0,0)push(r0Q, r0)
(0,0)r1 = in[i]
(0,1)r1 = in[i]
(0,2)r1 = in[i](0,0)r2 = r1*r0(0,0)push(r1Q, r1)
(0,3)r1 = in[i](0,1)r2 = r1*r0(0,1)push(r1Q, r1)
(0,4)r1 = in[i](0,2)r2 = r1*r0(0,2)push(r1Q, r1)
(0,4)r1 = in[i](0,3)r2 = r1*r0(0,3)push(r1Q, r1)(0,0)push(r2Q, r2)
(0,5)r1 = in[i](0,4)r2 = r1*r0(0,4)push(r1Q, r1)(0,1)push(r2Q, r2)
...
(1,0)push(r0Q,r0)
...
Process 2
(0,0)pop(r0Q,r0)
(0,0)pop(r1Q,r1)
(0,0)pop(r2Q,r2)
(0,0)out2 = r2
(0,1)pop(r1Q,r1)
(0,1)pop(r2Q,r2)
(0,1)out2 = r2
...
(0,0)out1[j] = r0
(1,0)pop(r0Q,r0)
(1,0)pop(r1Q,r1)
...
Schedules of Instruction Instances
push(r0Q,r0)
(-1,M)
(0,0)
push(r1Q, r1)
(0,-4)
(0,2)
push(r2Q, r2)
(0,0)
(-1,0)
(0,0)
(0,-1)
(0,0)
(0,-1)
pop(r0Q,r0)
(0,0)
pop(r1Q,r1)
(0,-1)
(0,0)
(-1,M)
pop(r2Q,r2)
Precedence Graph
push(r1Q, r1)
(0,-1)
pop(r1Q,r1)
(0,2)
(0,-1)
push(r2Q, r2)
(0,0)
pop(r2Q,r2)
Weight of Cycle : (0,0)
→ Artificial Deadlock