发信息.图 3 中右图则是通信 Petri 网的一个编码,注意到,只有 C 一个信道和 *ready* 一个信息,该全局变量α是一维的.整个编码与程序的控制流是非常相似的.这里,库所的名字与左边程序的行号直接挂钩.另外,如果一个令牌在库所 p_i 里,则表示这个令牌所代表的进程运行到了第 i 行,则这个程序能否 return 便取决于库所 p_2 里是否能有令牌.

4 k-型限制下的通信 Petri 网到数据 Petri 网的编码

本节将给出 k-型限制下的通信 Petri 网的可覆盖性可判定性的证明,证明的核心思路是:通过说明在 k-型限制下的通信 Petri 网中每个令牌所对应的一个二维向量可以认为是在全序集上的一个映射,从而证明受限的通信 Petri 网的可覆盖性问题可以规约到数据 Petri 网的可覆盖性问题上,而数据 Petri 网的可覆盖性问题可以通过规约至良序转移系统(WSTS)的可覆盖性问题上,从而证明其是可判定的[6].

直观上来理解,数据 Petri 网的拓展是将令牌直接绑定到了一个数据集上的一个数据,而在每次转移的过程中,令牌绑定的数据可能会发生变化,但这个变化是由规则绑定的,所以尽管有无穷多种令牌,但是由于有限条规则的限制,使得数据的变化是有限种.对于通信 Petri 网上性质的证明正是利用了这一点,使得在 k-型限制下的通信 Petri 网能够规约到数据 Petri 网的问题上.

给定一个 k-型限制下的通信 Petri 网 $k\text{-}CPN=(P,T,H,In,Out,f,\alpha)$和它的一个初始标记$(c_{ini},\alpha_{ini})=((p_1,M_1[N^2])(p_2,M_2[N^2])\dots(p_k,M_k[N^2]),\alpha_{ini})$以及一个目标标记 $(c,\alpha)=((p'_1,M'_1[N^2])(p'_2,M'_2[N^2])...(p'_k,M'_k[N^2]),\alpha)$, 其转换为一个数据 Petri 网的可覆盖性问题说明如下.

第 1 步,将 $k\text{-}CPN$ 的全局向量α编码进库所里.额外地,令 d 个库所 $p_{m1},p_{m2},\dots,p_{md}$,其中,$d$ 为向量α的维度,每当存在一条需要对α做加法的转移 t,即(In_t,t,n,Out_t),则将在转移 t 出发后,在 p_{mn} 里产生一个向量为(0,0)的令牌;每当存在一条需要对α做减法的转移 t,即(In_t,t,n,Out_t),则将在转移 t 出发后,在 p_{mn} 里消耗一个向量为(0,0)的令牌.在这些额外的 d 个库所中,由于是被用来相当于当成信道存储程序发送的信息,因此特别令在这些额外的 d 个库所中的令牌所代表的向量都是一样的.所以通过额外的 d 个库所的编码,便将全局向量α编码进了库所.

第 2 步,将每个令牌所代表的第 2 维向量编码进库所里.这一点由于有 k-型限制,可如下实现.

对于原先通信 Petri 网的每一个库所 p_i,定义对应的 k 个库所 $p_{i1},p_{i2},\dots,p_{ik}$,其中,$p_{ij}$ 存储在原本通信 Petri 网的库所 p_i 中存储的第 2 维向量 j 的令牌.在这样的情形下,新的模型里的库所数量是原来通信 Petri 网的 k 倍.因为库所被分离了,所以需要对现有的转移进行一些修改.下面根据原来的 5 类转移一一进行说明:第 1、第 3、第 4 和第 5 种转移是比较容易说明的,事实上,由于其不会修改里面库所中令牌第 2 维向量的值,因此对于一个需要消耗在 $p_1,p_2,\dots,p_s$ 库所里的令牌并且在 $p'_1,p'_2,...,p'_t$ 库所里生成对应令牌的转移,只需要添加类似的转移,使得转移是需要消耗在 $p_{1i_1},p_{2i_2},...,p_{si_s}$ 库所里的令牌并且在 $p'_{1i_1},p'_{2i_2},...,p'_{ti_t}$ 库所里生成对应令牌,其中,$i_m,j_n\in[k]$.我们注意到:在这种情形下,对于原先的一条转移,由于第 2 维向量并没有发生变化,因此为了在将第 2 维向量编码进状态的数据 Petri 网中模拟该转移,我们需要考虑到所有的情况.简单来说,比如:如果要消耗 p_1 库所里的 m_1 个令牌,注意到,这 m_1 个令牌的第 2 维向量可能是 0~k 中的任一数字,也就是说一共有 $(k+1)^{m_1}$ 种情况,对每个库所消耗的令牌进行考虑,假设在 p_i 库所里需要消耗 m_i 个令牌,则一共会有 $(k+1)^{\sum_{i=1}^{s}m_i}$ 种不同的可能,也就是说需要在数据 Petri 网中添加这么多种的转移.

现在来说明第 2 种转移的修改.注意到:这种类型的转移每次被触发时,一个令牌的第 2 维向量只需要加 1,因此这种转移的添加也比较方便.对于属于第 2 种转移的 t,即$((p_i,m),t,(p_j,m))$这样一个转移,只需要同样构造$((p_{ik},m),t_k,(p_{j(k+1)},m))$的 k 条转移即可.图 4 给出了一个在 2-型下的通信 Petri 网的转移的修改方式.直观上来理解,就是由于第 2 维向量是有限制的,因此可以在状态里控制第 2 维向量.

在对上述的转移进行修改以后,编码已经将 k-型限制下的通信 Petri 网的每一个令牌的第 2 维向量编码进库所里,所以从现在开始,将限定这里的每个令牌对应的只是一个自然数值.注意到,N 是一个全序集,因此可以采用数据 Petri 网使用的标记方法来表示现在一个通信 Petri 网下的标记.

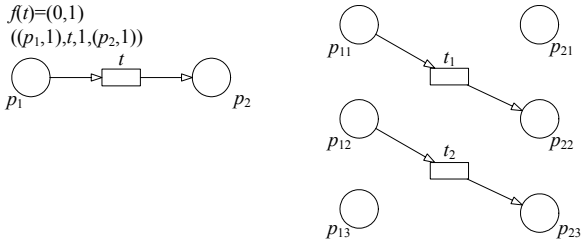

Fig.4 An example of 2-shaped C-Petri net transition

图 4 一个 2-型限制下的通信 Petri 网的转移修改

最后将证明,经过了如此变换的 k-型限制下的通信 Petri 网是一个数据 Petri 网.事实上,只需要说明每一个转移都可以表示成固定常数的一个 N^k 上的列即可.由于每次通信 Petri 网上的操作至多只需要对令牌所表示的向量完成+1,–1 的操作,也就是说,每个通信 Petri 网下涉及到 n 个令牌的转移至多只需写成 n 个 N^k 上的列即可,因此,原本 k-型限制下的通信 Petri 网已经被编码成了一个数据 Petri 网,并且保证了两个模型所能产生的路径是相同的.现在来形式化整个过程.

定义 4.1. 给定一个 k-型限制下的通信 Petri 网 $k\text{-}CPN=(P,T,H,In,Out,f,\alpha)$,其中,

1) P 是一个有限的元素集合 $P=\{p_1,p_2,\ldots,p_m\}$;

2) T 是一个有限的元素集合 $T=\{t_1,t_2,\ldots,t_n\}$;

3) $\alpha\in N^d$,

则 $D(k\text{-}CPN)$所对应的一个 $k\times m+d$ 维的数据 Petri 网是一个元组 $PDN=(P',T',F,H)$,其中,

1) $P'=\{P_{11},P_{12},\ldots,P_{1k},P_{21},\ldots,P_{m1},P_{m2},P_{mk},P_{msg1},P_{msg2},P_{msgd}\}$;

2) 对于 $k\text{-}CPN$ 中的每个转移 t:

i) 如果 $f(t)=(1,0),(In_t=(p_i,1),t,n,Out_t=(p_j,1))\in H$,则 $\{t_1,t_2,\ldots,t_k\}\subseteq T'$,并且 $F_{tl},H_{tl}\in N^{k\times m+d}$,其中,$F_{tl}((i-1)\times k+l)=1$,其余为 0;$H_{tl}((j-1)\times k+l)=1,H_{tl}(m\times k+n)=1$,其余为 0.

ii) 如果 $f(t)=(0,1),(In_t=(p_i,1),t,n,Out_t=(p_j,1))\in H$,则 $\{t_1,t_2,\ldots,t_k\}\subseteq T'$,并且 $F_{tl},H_{tl}\in(N^{k\times m+d})^2$,其中,$F_{tl}[1]((i-1)\times k+l)=1,H_{tl}[1](m\times k+n)=1$,其余为 0;$H_{tl}[2]((j-1)\times k+l)=1$,其余为 0.

iii) 如果 $f(t)=(0,0)$,并且 $(In_t=(p_{i1},1)(p_{i2},1)\ldots(p_{ip},1),t,n,Out_t=(p_{j1},a_1)(p_{j2},a_2)\ldots(p_{jq},a_q))\in H$,则 $\{t_{11},\ldots,t_{12}^{p-1},t_{21},\ldots,t_{k2}^{p-1}\}\in T'$, 并且对于 $x\in[k],y\in\left[\sum_{i=0}^{n-1}(p,i),\sum_{i=0}^{n}(p,i)\right),F_{txy},H_{txy}\in(N^{k\times m+d})^n$, 其中,$p_{i1},p_{i2},\ldots,p_{ip}$ 重新记标号为 $P_{11},P_{12},\ldots,P_{1b1},P_{2b2},\ldots,P_{nbn}$.其中,$\sum_{i=1}^{n}b_i=p$, 则 $F_{txy}[c]((i_d-1)\times k+x)=1,p_{id}$ 被重新记为 p_{ce},其余为 0;$H_{txy}[c]((j_d-1)\times k+x)=1,H_{txy}[c]((j_d-1)\times k+1)=a_d-1$,其余为 0.

iv) 如果 $f(t)=(2,1),(In_t=(p_i,1),t,n,Out_t=(p_j,1))\in H$,则 $\{t_1,t_2,\ldots,t_k\}\subseteq T'$,并且 $F_{tl},H_{tl}\in(N^{k\times m+d})^2$,其中,$F_{tl}[2]((i-1)\times k+l)=1$,其余为 0;$H_{tl}[1]((j-1)\times k+l)=1$,其余为 0.

v) 如果 $f(t)=(1,2),(In_t=(p_i,1),t,n,Out_t=(p_j,1))\in H$,则 $\{t_1,t_2,\ldots,t_k\}\subseteq T'$,并且 $F_{tl},H_{tl}\in(N^{k\times m+d})^2$,其中,$F_{tl}[1]((i-1)\times k+l)=1$,其余为 0;$H_{tl}[2]((j-1)\times k+l)=1$,其余为 0.

vi) $F_{d1},H_{d1},F_{d2},H_{d2}\in(N^{k\times m+d})^2$,其中,$F_{d1}[2](m\times k+d)=1$,其余为 0;$H_{d1}[1](m\times k+d)=1$,其余为 0;$F_{d2}[1](m\times k+d)=1$,其余为 0;$H_{d2}[2](m\times k+d)=1$,其余为 0.

定义 4.2. 定义函数 $m:(P,M_1[N^2])^*\times N^d\to(N^{k\times m+d})^*$和函数 $h:(P\times M_1[N^2])^*\to N$,其中,

1) $h(c)=\max\ d,\prod_{M_i}(d,\_)>0$;

2) $m(c,\alpha)\in(N^{m\times k+d})^{k(c)}$,其中,$m(c,\alpha)[i]((k\times(a-1)+j)=d$,如果在 c 中 $\prod_{M_a}((i,j))=d$.特别地,

$$\sum m(c,\alpha)[i](k\times m+j)=\alpha[j].$$

我们对这两个函数定义进行一定的说明.h 函数实际上是一个作用在一个通信 Petri 网的标记上的函数,返回在该通信 Petri 网当前标记下令牌中第 1 维向量的最大值.这个函数的作用在于服务接下来的 m 函数.实际上,m 函数是一个通信 Petri 网标记转换成将令牌第 2 维编码进状态的数据 Petri 网标记的转换函数.也就是说,如果(c,α)是 k-型限制下的通信 Petri 网的一个标记,则 $m(c,\alpha)$是 $D(k\text{-}CPN)$上对应的一个标记.并且由 m 函数的定义,我们不难定义出 m 函数的逆函数 $m^{-1}:(N^{k\times m+d})^*\to(P,M_1[N^2])^*\times N^d$.关于这两个函数,我们有如下的性质.

命题 4.1. 对于函数 h,m 以及其逆函数 m^{-1},我们有如下性质.

1. 对任意 $c,c'\in(P,M_1[N^2])^*$,如果有 $c\geqslant c'$,我们有 $h(c)\geqslant h(c')$;
2. (c,α)是 k-型限制下的通信 Petri 网的一个标记,则 $m(c,\alpha)$是 $D(k\text{-}CPN)$上对应的一个标记;
3. 对任意 $d,d'\in(N^{k\times m+d})^*$,并且 $m^{-1}(d)=(c,\alpha),m^{-1}(d')=(c',\alpha')$,若$(c,\alpha)=(c',\alpha')$,则对$\forall i\in N,\forall j\in N$,有:
$$d[i](j)=d'[i](j);$$
4. 对任意$(c,\alpha),(c',\alpha')\in(P,M_1[N^2])^*\times N^d,c\geqslant c'$,当且仅当 $m(c,\alpha)[i](j)\geqslant m(c',\alpha')[i](j)$,其中,$i\in[h(c)],j\in[k\times m]$.

证明:从函数 h,m 和函数 m^{-1} 的定义不难得到.

1. 由函数 h 的定义立即得出;
2. 由函数 m 的定义及 $D(k\text{-}CPN)$的定义立即得出;
3. 因为有 $m^{-1}(d)=m^{-1}(d')$,因此有 $c=c'$,所以对于任意的 $i+k\times j\in[k\times m],\prod_{M_i}((a,j))=\prod_{M_i'}((a,j))$ 对任意 a 均成立.因而有 $d[a][i\times k+j]=d'[a][i\times k+j]$;
4. 由 $c\geqslant c'$可知:对于任意的$(i,j)\in N^2$,有$\prod_{M_a}((i,j))\geqslant\prod_{M_a'}((i,j))$,其中,$a\in[m],j\in[k]$,因此对于任意的 $i_0\in[h(c)],j_0\in[k\times m]$,有 $m(c,\alpha)[i_0](j_0)\geqslant m(c',\alpha')[i_0](j_0)$;反之,同理.因而结论成立. □

接下来将说明 k-型限制下的通信 Petri 网 $k\text{-}CPN$ 和一个对应的数据 Petri 网 $D(k\text{-}CPN)$之间的关系.

引理 4.1. 对于一个 k-型限制下的通信 Petri 网 $k\text{-}CPN$ 和一个对应的数据 Petri 网 $D(k\text{-}CPN)$以及 $k\text{-}CPN$ 的两个标记$(c,\alpha),(c',\alpha')$,$(c,\alpha)\to(c',\alpha')$当且仅当在 $D(k\text{-}CPN)$中 $m(c,\alpha)\to m(c',\alpha')$.

证明:只需证明对于在 $k\text{-}CPN$ 上$(c,\alpha)\to(c',\alpha')$,当且仅当在 $D(k\text{-}CPN)$上 $d\to d',m(c,\alpha)=d,m(c',\alpha')=d'$.

($\Rightarrow$)的证明:

假设$(c,\alpha)\to_t(c',\alpha')$,并且

$$(c,\alpha)=((p_1,M_1[N^2])(p_2,M_2[N^2])...(p_k,M_k[N^2]),\alpha),(c',\alpha')=((p_1,M_1'[N^2])(p_2,M_2'[N^2])...(p_k,M_k'[N^2]),\alpha').$$

接下来将说明存在 $D(k\text{-}CPN)$的一条转移 t',使得 $m(c,\alpha)\to_{t'}m(c,\alpha)'$,并且 $m(c,\alpha)'=m(c',\alpha')$.其中,$m(c,\alpha)'$表示为 $m(c,\alpha)$作转移 t'后新的数据 Petri 网的标记.这里,以第 1 类、第 3 类转移作为说明,其余 3 类转移的说明是类似的,这里不再重复.

1) 如果 $f(t)=(1,0),(In_t=(p_i,1),t,n,Out_t=(p_j,1))\in H$,则存在一个大小为 1 的多重集合 $M[N^2]$,使得:
$$M_{j'}[N^2]=M_j[N^2]+M[N^2],M_{i'}[N^2]=M_i[N^2]+M[N^2],\alpha'[n]=\alpha[n]+1.$$

令 $M[N^2]=\{(b,e)\}$.由 $D(k\text{-}CPN)$的构造可知,存在 $F_{t'},H_{t'}\in N^{k\times m+d},F_{t'}((i-1)\times k+e)=1,H_{t'}((j-1)\times k+e)=1,H_{t'}(m\times k+n)=1$.在 $m(c,\alpha)$中考察第 b 个向量 $m(c,\alpha)[b]$,注意到$(b,e)\in M_i[N^2]$,因此,由 m 的规则,$m(c,\alpha)[b](k\times(i-1)+e)>0$.因此,$m(c,\alpha)$可作该条转移 t',使得 $m(c,\alpha)'$满足:

$$\begin{aligned}&m(c,\alpha)'[b](k\times(i-1)+e)=m(c,\alpha)[b](k\times(i-1)+e)-1,\\&m(c,\alpha)'[b](k\times(j-1)+e)=m(c,\alpha)[b](k\times(j-1)+e)+1,\\&m(c,\alpha)'[b](k\times m+n)=m(c,\alpha)[b](k\times m+n)+1.\end{aligned}$$

其余位置与 $m(c,\alpha)$相等.

另一边,由于 $\alpha'[n]=\alpha[n]+1,\prod_{M_i'}(b,e)=\prod_{M_i}(b,e)-1,\prod_{M_j'}(b,e)=\prod_{M_j}(b,e)+1$, 因此,由定义 $m(c',\alpha')$,满足:

$$\begin{aligned}&m(c',\alpha')[b](k\times(i-1)+e)=m(c,\alpha)[b](k\times(i-1)+e)-1,\\&m(c',\alpha')[b](k\times(j-1)+e)=m(c,\alpha)[b](k\times(j-1)+e)+1,\\&m(c',\alpha')[b](k\times m+n)=m(c,\alpha)[b](k\times m)+n)+1.\end{aligned}$$

其余位置与 $m(c,\alpha)$相等.因此有 $m(c,\alpha)'=m(c',\alpha')$.

2) 如果 $f(t)=(0,0)$,并且$(In_t=(p_{i1},1)\ldots(p_{ip},1),t,n,Out_t=(p_{j1},a_1)\ldots(p_{jq},a_q))\in H$,则存在 p 个大小为 1 的多重集合 $M_u[N^2]$. 令 $M_u[N^2]=\{(b_u,e_u)\}$, 使得 $M_{iu'}[N^2]=M_{iu}[N^2]-M_u[N^2]M_{jk'}[N^2]=M_{jk}[N^2]+M_u[N^2]+M[\{(b_u,0)\}]$, 并且 $|M[\{(b_u,0)\}]|=a_k-1$.不妨令 b_i 有 k 个不同的值.由 $D(k\text{-}CPN)$的构造可知,存在 $F_t,H_t\in(N^{k\times m+d})^k$,使得 $F_t[g]((i-1)\times k+e)=1$, $H_t[g]((z-1)\times k+e)=1,H_t[g]((i-1)\times k+1)=a_i-1$.其中,$i,g$ 满足 b_i 在这 p 个值里排在第 g 小的位置,$z=j_s$.由于 $h(c)\geqslant\max b_i$,因此在 $m(c,\alpha)$中存在 k 个位置上的向量,使得 $m(c,\alpha)[x_i]\geqslant F_t[i]$,于是,$m(c,\alpha)$可作该条转移 t',使 $m(c,\alpha)'$满足:

$$m(c,\alpha)'[x_i]((i-1)\times k+e_i)=m(c,\alpha)[x_i]((i-1)\times k+e_i)-1,$$
$$m(c,\alpha)[x_i]((k_i-1)\times k+e_i)=m(c,\alpha)[x_i]((k_i-1)\times k+e_i)+1,$$
$$m(c,\alpha)'[x_i]((k_i-1)\times k+1)=m(c,\alpha)'[x_i]((k_i-1)\times k+e_i)+a_i-1.$$

其余与 $m(c,\alpha)$相等.

另一方面,由 $M_{jk'}[N^2]$的变化及 m 函数的定义可知:

$$m(c',\alpha')[x_i]((i-1)\times k+e_i)=m(c,\alpha)[x_i]((i-1)\times k+e_i)-1,$$
$$m(c',\alpha')[x_i]((k_i-1)\times k+e_i)=m(c,\alpha)[x_i]((k_i-1)\times k+e_i)+1,$$
$$m(c',\alpha')[x_i]((k_i-1)\times k+1=m(c',\alpha')[x_i]((k_i-1)\times k+e_i)+a_i-1.$$

其余位置与 $m(c,\alpha)$相等.

因此有 $m(c,\alpha)'=m(c',\alpha')$.因此由归纳假设,结论成立.

($\Leftarrow$)的证明:

反过来的证明是相对比较容易的,只要注意到:对于 $D(k\text{-}CPN)$的每个转移 t,都是由原先通信 Petri 网上的一个转移 t 生成而来.因此,如果在 $D(k\text{-}CPN)$上有 $d\rightarrow d'$,在 $k\text{-}CPN$ 上即有 $m^{-1}(d)\rightarrow m^{-1}(d')$. □

因此,k-型下的通信 Petri 网可覆盖性问题规约到了数据 Petri 网的可覆盖性问题上去.由引理 4.1、定理 1.1,便得到了下面的定理.

定理 4.1. k-型限制下的通信 Petri 网的可覆盖性问题是可判定的.

5 相关工作

异步通信程序分析理论与方法是最近几年计算机理论和程序理论研究者关注的热点,研究者们针对上述 5 个无限状态因素进行限制和抽象,目前主要的研究分支和研究结果如下.

- 限制递归.

即:每个进程不允许函数调用,同时假设各个进程通过无界容量的无序缓存进行通信.这样的系统是异步通信系统,其可覆盖性可通过规约至 Petri 网的可覆盖性,因此是可判定的[7].基于该模型,研究者们设计了一个 Erlang 语言的分析器 Soter[8].文献[9]中,研究者对递归进行了 k-型的限制,在这样的情况下,同时对输入通信的次数进行有界限制,系统的可覆盖性依然可判定,且表达能力严格超过 Petri 网,达到 Tower 完备[9].

- 限制并发.

假设只有一个处理器执行程序,且线程执行时不允许被挂起,也即假设异步并发程序可以线性化.在这种限制下,只有一个程序栈就足够了.即使这样的模型,其表达能力依然很强,是否是图灵完备目前仍然是一个开放问题.文献[2]在上述限制的同时,对通信也进行了限制——只有在栈空的时候才能发生通信,这样的程序模型的状态可达性(EXPSPACE 难)[2]和可覆盖性[10]是可判定的.

- 限制通信.

最极端的限制方法是不允许不同线程间进行通信,这样的程序也即异步程序.在这样的限制下,模型的可覆盖性可以通过规约至 Petri 网的可覆盖性得到可判定的结论[11],其活性(liveness)可以通过规约至 Petri 网的可达性,因此也是可判定的[12].文献[13]详尽地列出了在不允许通信的情况下,各种程序模型子集的判定性结论,对应于不同的程序类型.对于这样的程序模型系统,研究者开发了一个程序分析器原型[14].最新的研究扩展了不允许线程之间的通信,得到了一个限制性比较强的可判定性结论[6].

还有一类型研究是为了对异步的 Web 程序进行分析,假设系统有一个主线程以及无界数量个同类型的辅线程,通信只发生在主辅线程之间,辅线程之间没有通信.因此,辅线程可以参数化.这样的模型叫做参数化异步通信系统.文献[15]首次提出了这样的系统,并给出这一系统的可覆盖性的复杂性上界是 EXPSPACE 的.最后,可覆盖性被确定为 PSPACE 完备[16].在最新的研究成果中,该系统的活性上界被证明是 NEXPTIME 的[17].同时,一些更为扩展的模型的可判定性结论也被提了出来[18].

此外,计算机理论界的一些研究也对本研究具有很强的借鉴作用,比如最新的关于下推向量加法系统(pushdown vector addition system)[19],其实际表达是单进程异步通信程序模型的一个超集,它的可判定结论可以直接影响后者的判定性结论.目前的研究成果是下推向量加法系统在一维情况下可覆盖性可判定[20],这一结论尚不能指导异步通信程序的理论模型(因为程序模型不可能只有一维向量).良结构传递系统[21]及其扩展[10,22]也可以对程序模型的判定性结论起到指导以及证明作用,其优点是可得到通用的证明过程,但不足之处是很难得到复杂性结论以及可实现的算法,因此不太被程序语言理论界所认可.

由这些研究可知,目前大多数的可判定性结论都是规约至 Petri 网的,因此使得这一古老的模型重新焕发新机.最近,很多 Petri 网的高效验证器[23,24]被研究开发出来,作为程序分析器的引擎.文献[25]在不完备的情况下,基于通过 SMT-solver 实现了基于模式检测的 Petri 网可覆盖性验证.文献[11]将异步程序的进程间数据流分析形式化成了一个 AIFDS 问题,并提出了相应的算法.其结论表示:尽管解决该问题是 EXPSPACE 难的,但在实际中效率还是相当快的.

在 20 世纪末,大量 Petri 网的扩展模型也开始应用于并发程序.文献[26]提出了颜色 Petri 网(coloured Petri net),对同步的通信进行了研究.文献[27,28]则提出了对象 Petri 网(object Petri net),对面向对象的程序进行了建模研究.近些年,一些更为具体的应用方面的研究也已经产生.文献[8]针对 Erlang 语言提出了一个叫做 Soter 的自动研究工具,文献[14]也提出了一个针对异步程序开发的程序分析器原型.

此外,除了基于 Petri 网的研究,在应用中也有不少其他方式的探索.文献[29]对 MHP 问题,也就是问两个活动能否在程序里并发地进行这一基础的问题进行了探索,并且在基于类 x10 语言上提出了一种静态 MHP 分析的算法,有着较快的效率.文献[30]则对异步程序进行了依靠/保证(rely/guarantee)的抽象,即,每个异步进程都需要满足这两个断言.这样,整个程序的正确性就依赖于这些断言需要被满足,文献[30]基于此给出一些 C 语言中的例子.文献[31]为了能够编程有效的异步程序,定义了一种高可靠的程序语言 P#,通过静态数据竞争检测等手段来满足有效的要求.

但是,基于这样的模型存在着一些问题,比如:目前,所有的程序模型通信都不考虑传值(value-passing);在文献[3]中提出的异步通信下推系统(asynchronous communicating pushdown systems)中,便将消息限制在了有限种;文献[6]在对基于事件同步的异步程序分析的过程中,也将消息种类作了限定,因此无法表达和分析像 future 和 promise 这样的语言特性.再比如:目前,所有模型也无法表达某线程是否可以知道所接受的信息是某个特定线程所发,等等.因此,研究者们希望研究出表达能力更强的理论模型来描述和分析异步通信程序.与此同时,研究者们也希望能基于此模型开发出高效的验证器来自动实现这些程序的程序分析.

6 总结以及未来的工作

本文针对异步通信程序给出了一个抽象的模型通信 Petri 网来编码,证明了在 k-型限制下的通信 Petri 网的可覆盖性问题是可判定的,并且简单地给出了一个复杂性的下界.通信 Petri 网是将交互与程序调用合理地加以分离,更方便直观理解和实现.

未来的工作包括:

- 在理论上,我们希望得到更为一般化的可判定模型来对异步通信程序进行建模和验证;同时验证其他性质,比如活性.此外,我们希望能对其与其他类似的模型做出能力上的比较,希望能严格证明其是当前最大的可判定模型;
- 在实现上,我们希望能够通过该模型提出有效使用的算法,以此实现强大的通信 Petri 网的验证工具和

从 Erlang 语言到该工具的自动转化程序,从而得到自动化 Erlang 语言的验证工具.

References:

[1] Ramalingam G. Context-Sensitive synchronization-sensitive analysis is undecidable. ACM Trans. on Programming Languages and Systems (TOPLAS), 2000,22(2):416−430. [doi: 10.1145/349214.349241]

[2] Sen K, Viswanathan M. Model checking multithreaded programs with asynchronous atomic methods. In: Proc. of the Int'l Conf. on Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2006. 300−314. [doi: 10.1007/11817963_29]

[3] Kochems J, Ong CHL. Safety verification of asynchronous pushdown systems with shaped stacks. In: Proc. of the Int'l Conf. on Concurrency Theory. Berlin, Heidelberg: Springer-Verlag, 2013. 288−302. [doi: 10.1007/978-3-642-40184-8_21]

[4] Petri CA. Kommunikation mit automaten [Ph.D. Thesis]. University of Bonn, 1962.

[5] Lazić R, Newcomb T, Ouaknine J, Roscoe AW, Worrell J. Nets with tokens which carry data. Fundamenta Informaticae, 2008, 88(3):251−274.

[6] Emmi M, Ganty P, Majumdar R, Rosa-Velardo F. Analysis of asynchronous programs with event-based synchronization. In: Proc. of the European Symp. on Programming Languages and Systems. Berlin, Heidelberg: Springer-Verlag, 2015. 535−559. [doi: 10.1007/978-3-662-46669-8_22]

[7] D'Osualdo E, Kochems J, Ong CHL. Automatic verification of Erlang-style concurrency. In: Proc. of the Int'l Static Analysis Symp. Berlin, Heidelberg: Springer-Verlag, 2013. 454−476. [doi: 10.1007/978-3-642-38856-9_24]

[8] D'Osualdo E, Kochems J, Ong L. Soter: An automatic safety verifier for Erlang. In: Proc. of the 2nd Edition on Programming Systems, Languages and Applications Based on Actors, Agents, and Decentralized Control Abstractions. ACM Press, 2012. 137−140. [doi: 10.1145/2414639.2414658]

[9] Kochems J. Verification of asynchronous concurrency and the shaped stack constraint [Ph.D. Thesis]. University of Oxford, 2015.

[10] Cai X, Ogawa M. Well-Structured pushdown systems. In: Proc. of the Int'l Conf. on Concurrency Theory. Berlin, Heidelberg: Springer-Verlag, 2013. 121−136. [doi: 10.1007/978-3-642-40184-8_10]

[11] Jhala R, Majumdar R. Interprocedural analysis of asynchronous programs. ACM SIGPLAN Notices, 2007,42(1):339−350. [doi: 10.1145/1190215.1190266]

[12] Ganty P, Majumdar R, Rybalchenko A. Verifying liveness for asynchronous programs. ACM SIGPLAN Notices, 2009,44(1): 102−113. [doi: 10.1145/1594834.1480895]

[13] Ganty P, Majumdar R. Algorithmic verification of asynchronous programs. ACM Trans. on Programming Languages and Systems (TOPLAS), 2012,34(1):6. [doi: 10.1145/2160910.2160915]

[14] Majumdar R, Wang Z. BBS: A phase-bounded model checker for asynchronous programs. In: Proc. of the Int'l Conf. on Computer Aided Verification. Springer Int'l Publishing, 2015. 496−503. [doi: 10.1007/978-3-319-21690-4_33]

[15] Hague M. Parameterised pushdown systems with non-atomic writes. In: Proc. of the 31st Int'l Conf. on Foundations of Software Technology and Theoretical Computer Science. 2011. 457. http://arxiv.org/abs/1109.6264

[16] Esparza J, Ganty P, Majumdar R. Parameterized verification of asynchronous shared-memory systems. In: Proc. of the 25th Int'l Conf. on Computer Aided Verification (CAV 2013), Vol.8044. Saint Petersburg: Springer-Verlag, 2013. 124. [doi: 10.1007/978-3-642-39799-8_8]

[17] Durand-Gasselin A, Esparza J, Ganty P, Majumdar R. Model checking parameterized asynchronous shared-memory systems. In: Proc. of the Int'l Conf. on Computer Aided Verification. Springer Int'l Publishing, 2015. 67−84. [doi: 10.1007/978-3-319-21690-4_5]

[18] La Torre S, Muscholl A, Walukiewicz I. Safety of parametrized asynchronous shared-memory systems is almost always decidable. In: Proc. of the 26th Int'l Conf. on Concurrency Theory. 2015. 72. http://drops.dagstuhl.de/opus/volltexte/2015/5381

[19] Leroux J, Praveen M, Sutre G. Hyper-Ackermannian bounds for pushdown vector addition systems. In: Proc. of the Joint Meeting of the 23rd EACSL Annual Conf. on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symp. on Logic in Computer Science (LICS). ACM Press, 2014. 63. [doi: 10.1145/2603088.2603146]

[20] Leroux J, Sutre G, Totzke P. On the coverability problem for pushdown vector addition systems in one dimension. In: Proc. of the Int'l Colloquium on Automata, Languages, and Programming. Berlin, Heidelberg: Springer-Verlag, 2015. 324−336. [doi: 10.1007/978-3-662-47666-6_26]

[21] Finkel A, Schnoebelen P. Well-Structured transition systems everywhere. Theoretical Computer Science, 2001,256(1):63−92. [doi: 10.1016/S0304-3975(00)00102-X]

[22] Chadha R, Viswanathan M. Decidability results for well-structured transition systems with auxiliary storage. In: Proc. of the Int'l Conf. on Concurrency Theory. Berlin, Heidelberg: Springer-Verlag, 2007. 136−150. [doi: 10.1007/978-3-540-74407-8_10]

[23] Rodríguez C. Verification Based on Unfoldings of Petri Nets with Read Arcs. École Normale Supérieure de Cachan, 2013.

[24] Siebert M, Flochová J. Pnets—The verification tool based on Petri nets. In: Proc. of the World Congress on Engineering. 2013. 1. http://www.iaeng.org/publication/WCE2013/WCE2013_pp369-373.pdf

[25] Esparza J, Ledesma-Garza R, Majumdar R, Meyer P, Niksic F. An SMT-based approach to coverability analysis. In: Proc. of the Int'l Conf. on Computer Aided Verification. Springer Int'l Publishing, 2014. 603−619. [doi: 10.1007/978-3-319-08867-9_40]

[26] Christensen S, Hansen ND. Coloured Petri nets extended with channels for synchronous communication. In: Proc. of the 15th Int'l Conf. on the Application and Theory of Petri Nets. LNCS 815, Zaragoza: Springer-Verlag, 1994. 159−178. [doi: 10.1007/3-540-58152-9_10]

[27] Lakos C. The consistent use of names and polymorphism in the definition of object Petri nets. In: Proc. of the 17th Int'l Conf. on the Application and Theory of Petri Nets. LNCS, Springer-Verlag, 1996. 380−399. [doi: 10.1007/3-540-61363-3_21]

[28] Valk R. Petri nets as token objects? An introduction to elementary object nets. In: Proc. of the 19th Int'l Conf. on the Application and Theory of Petri Nets. LNCS 1420, Springer-Verlag, 1998. [doi: 10.1007/3-540-69108-1_1]

[29] Lee JK, Palsberg J, Majumdar R, Hong H. Efficient may happen in parallel analysis for async-finish parallelism. In: Proc. of the Int'l Static Analysis Symp. Berlin, Heidelberg: Springer-Verlag, 2012. 5−23. [doi: 10.1007/978-3-642-33125-1_4]

[30] Gavran I, Niksic F, Kanade A, Rupak M, Viktor V. Rely/Guarantee reasoning for asynchronous programs. In: Proc. of the LIPIcs-Leibniz Int'l Proceedings in Informatics. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015. 42. http://drops.dagstuhl.de/opus/volltexte/2015/5390

[31] Deligiannis P, Donaldson AF, Ketema J, Lal A, Thomson P. Asynchronous programming, analysis and testing with state machines. ACM SIGPLAN Notices, 2015,50(6):154−164. [doi: 10.1145/2813885.2737996]

杨启哲(1994－),男,上海人,博士生,主要研究领域为形式化方法,可计算学习理论.

李国强(1979－),男,博士,副教授,CCF 专业会员,主要研究领域为形式化方法,程序语言理论,可计算学习理论.