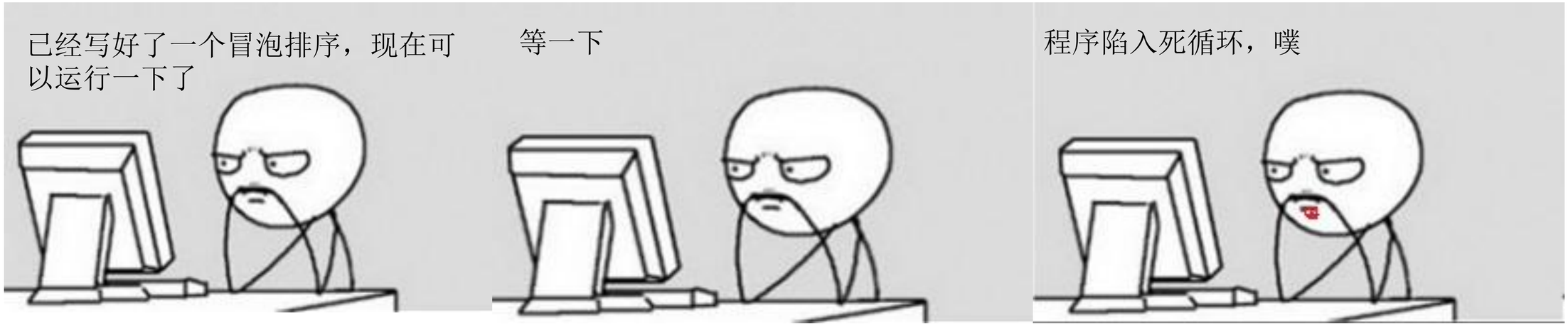
学术论文

针对程序终止性验证的高级自动机算法

李勇、Andrea Turrini、张立军等

论文会议录取：Advanced Automata-Based Algorithms for Program Termination Checking. In ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI’18).

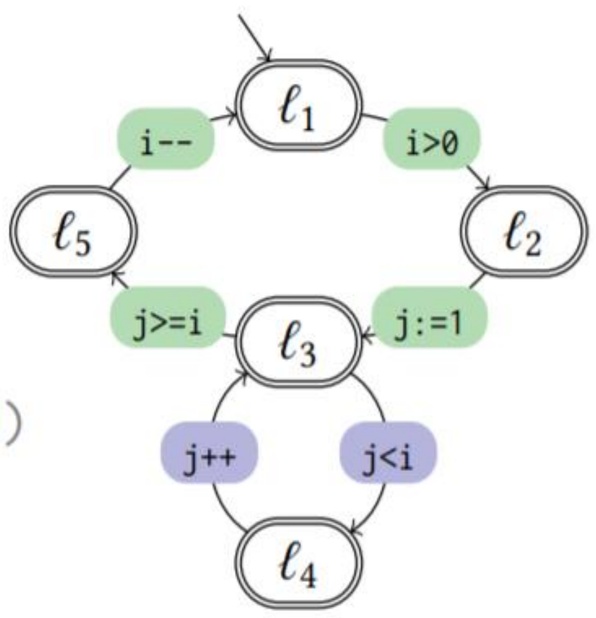
论文联系人：李勇 (liyong@ios.ac.cn, 185 1583 1627)



长知识了，Büchi 自动机(BA)可以用来表示程序并检验程序终止性哦！

```
program sort(int i):
ℓ1: while (i>0):
ℓ2:   int j:=1
ℓ3:   while (j<i):
        // if (a[j]>a[i]):
        //   swap(a[j],a[i])
ℓ4:     j++
ℓ5:   i--
```

(a) Program P^{sort}



(b) The BA $\mathcal{A}^{P^{sort}}$

1. 每次从BA里面取出一条循环路径并用现有工具来验证其是否终止，如路径i>0;j:=1;(j<i;j++)$^{\omega}$终止是因为每次循环j增加1，并且可以使用一个Rank函数i-j使得每次循环该函数值都会下降来说明。因此可以将每次已经证明了终止的路径从BA中去掉。
2. 当BA的路径全部去掉为空的时候就说明程序已经证明终止。如果某条路径被证明不终止便可得到一个出现死循环的反例，这样也可以方便找程序错误。但是有时候路径终止与否是不一定能判定的，为什么呢？（图灵机停机问题？）
3. 上述已证明的终止路径可以进行扩展得到一个BA使得这个BA里所有路径都能够终止并且能使用相同的Rank函数证明其终止性。

问题1：去掉已证明路径涉及对一般的有n个状态BA的取反问题，这个问题是非常困难的，复杂度是$2^{(O(n\ logn))}$

解决思路：. 设计一个多层级BA取反策略，即尽量将路径扩展后的BA优先转化为(1)只有一个吸收接收状态BA(复杂度为O(1)); (2) 确定性BA(复杂度为O(n)); (3)半确定性BA(复杂度为$O(2^n)$); (4) 一般的BA(复杂度为$2^{(O(n\ logn))}$)

)

问题2：实验中发现多层级BA取反中半确定性BA出现次数多且半确定性BA取反为效率瓶颈，其取反困难主要因为不确定猜测多

解决思路：.设计一个半确定性BA的优化算法，由于取反困难的主要原因是不确定性猜测多，于是经过分析我们发现很多猜测其实可以进行延迟或者减少。另外我们在取反搜索空间中还运用了基于语言包含的状态减枝策略，使取反算法更加优化。