APEX: An analyzer for open probabilistic programs

Stefan Kiefer[1] Andrzej S. Murawski[2] Joël Ouaknine[1]
Björn Wachter[1] James Worrell[1]

[1]University of Oxford, UK

[2]University of Leicester, UK

CAV 2012, Berkeley
11 July 2012

APEX Overview

open = program may have unspecified variables or functions

APEX key technology: game semantics
$\Rightarrow$ translates probabilistic programs to probabilistic automata
$\Rightarrow$ automaton represents the
observable behavior of an algorithm or protocol
$\Rightarrow$ observable: input, output, maybe timing, ...
unobservable: internal computation, maybe timing, ...

APEX can analyze:

- dining cryptographers
- Hibbard's algorithm for random tree insertion
- Herman's self-stabilization protocol
- ...

Equivalence

Verification of open programs
reduces to checking program equivalence.

Theorem (Murawski, Ouaknine, CONCUR'05)

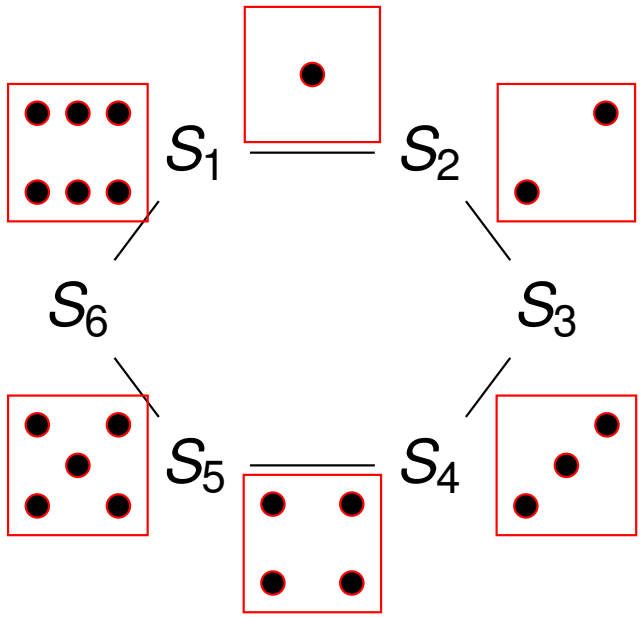
Two open probabilistic program are equivalent
if and only if
the corresponding prob. automata are language equivalent.

Language equivalence of prob. automata reduces to a linear algebra problem with efficient solutions, see [KMOWW, CAV'12]

APEX also performs the language equivalence check.
Provides a counterexample in case of inequivalence.

Example: The Grades Protocol

Students want to find out the sum of their grades.
No student wants to reveal anything about her/his own grade.



Each student announces $(g + \ell - r) \bmod N$.
The sum is telescoping $\Rightarrow$ equals the sum of grades (mod N).

But maybe individual grades leak?

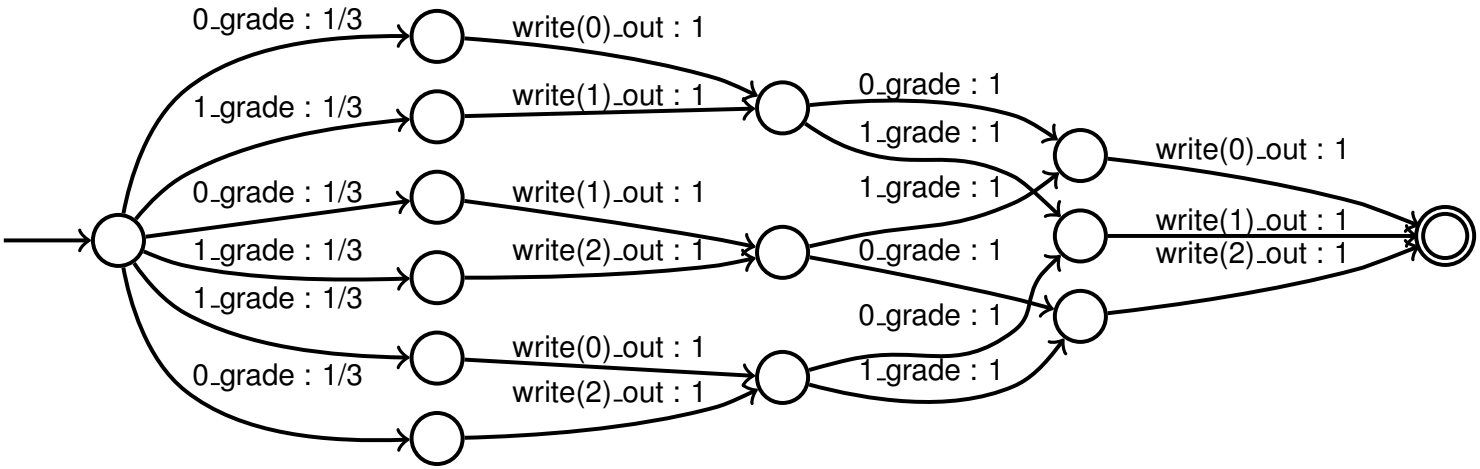
Example: The Grades Protocol

```
\\ Implementation

const N := S * (G-1) + 1;

grade:int%G, out:var%N |-
    var%(S+1) i; i := 0;
    var%N first; first := rand[N];
    var%N r; r := first;
    while(i<S) do {
      var%N l;
      i := succ(i);
      if(i=S) then
        l := first
      else
        l := rand[N];
        out := (grade + l) - r;
        r := l;
    }
```

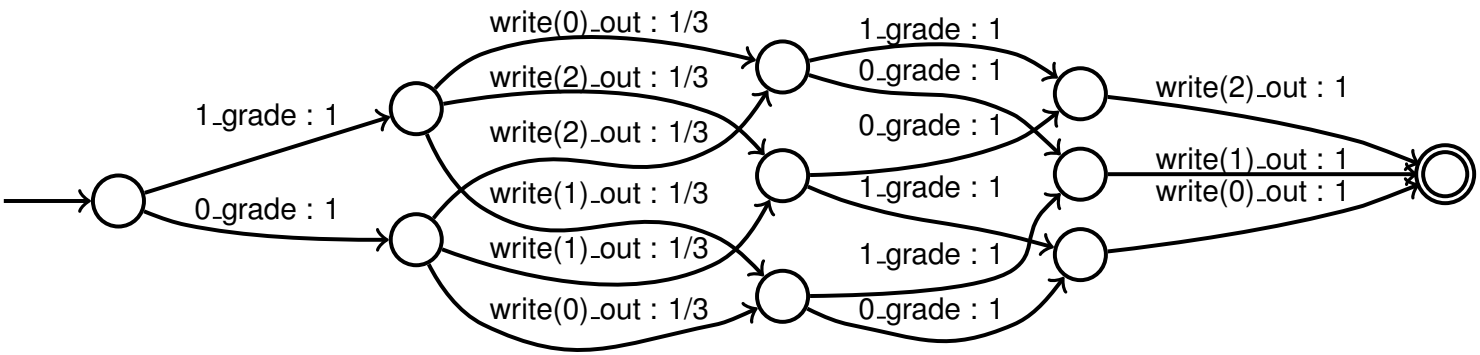
```
\\ Specification

const N := S * (G-1) + 1;

grade:int%G, out:var%N |-
    var%S i;
    var%N total;
    i := 1;

    while(i) do {
      total := grade + total;
      var%N r;
      r := rand[N];
      out := r;
      total := total - r;
      i := succ(i)
    };
    out := grade + total
```

Example: The Grades Protocol

Implementation:

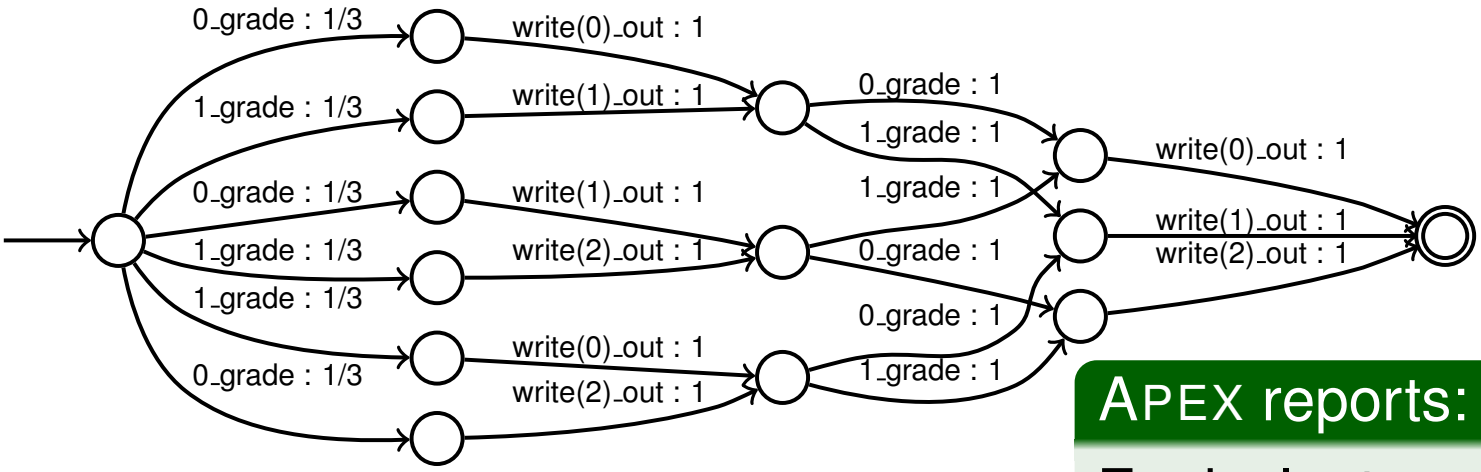


Specification:



Example: The Grades Protocol

Implementation:

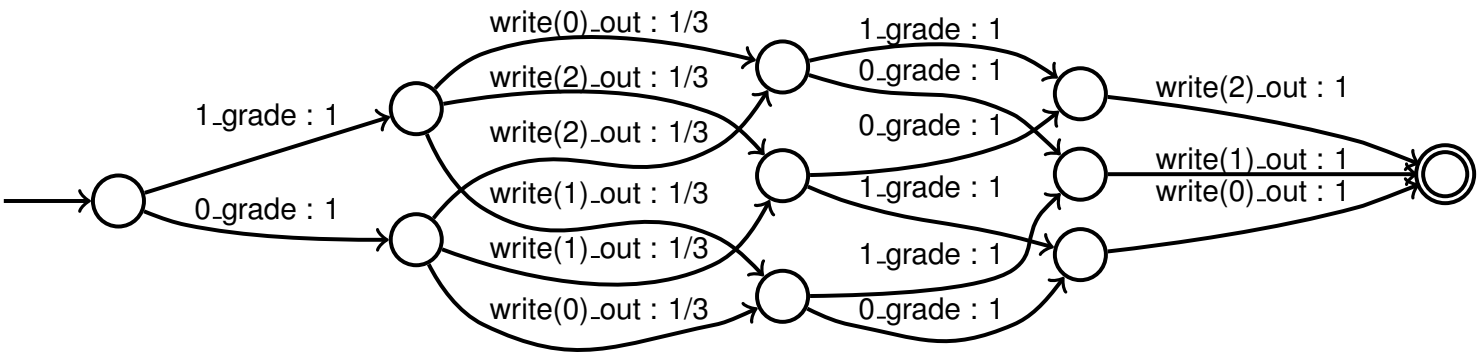


APEX reports:

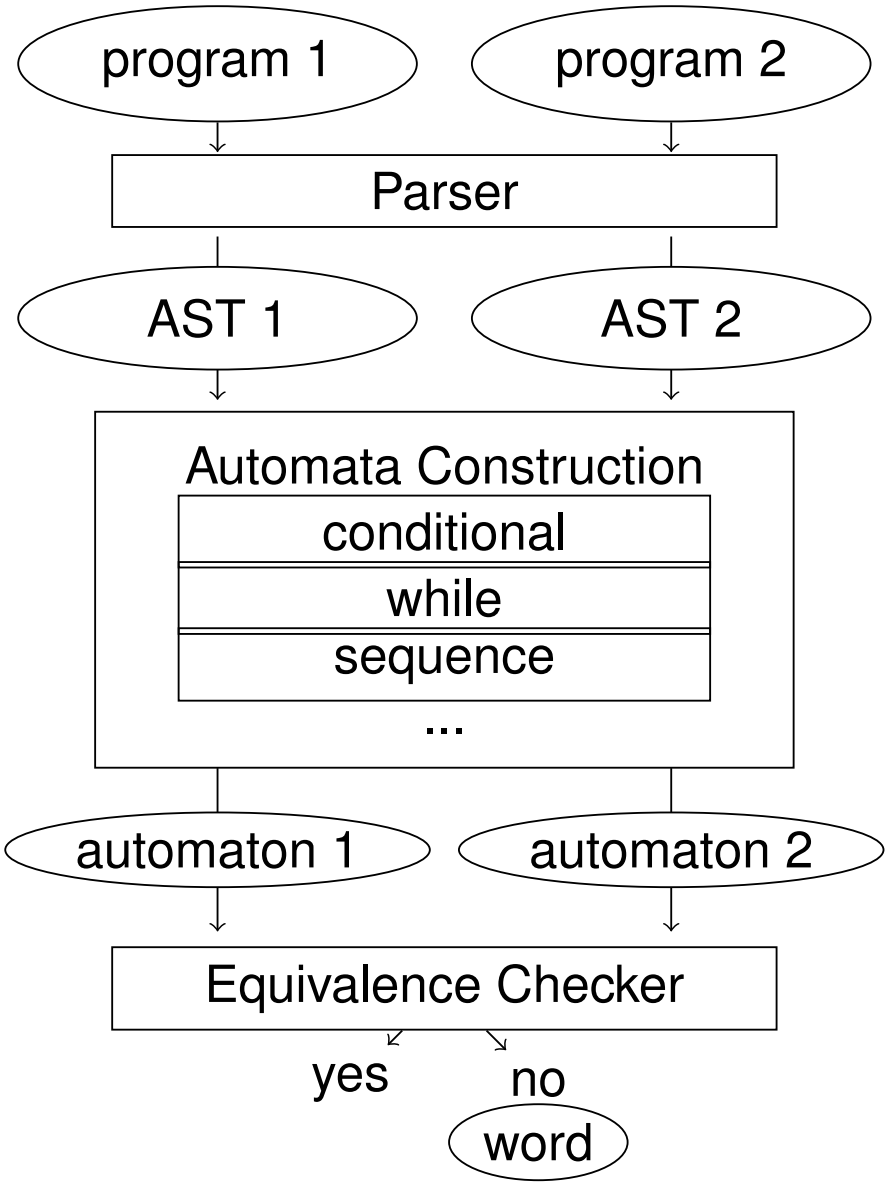
Equivalent.

Specification:

$\Rightarrow$ anonymity



APEX Architecture



APEX Online Tool Demo

Try our online tool demo at

cs.ox.ac.uk/apex