Specifying and Verifying Secrecy

Pavol Černý

University of Pennsylvania

March 10, 2006

Prologue: Sherlock Holmes.

It is an old maxim of mine that when you have excluded the impossible, whatever remains, however improbable, must be the truth. [1]

Circumstantial evidence is a very tricky thing. It may seem to point very straight to one thing, but if you shift your own point of view a little, you may find it pointing in an equally uncompromising manner to something entirely different. [2]

[1] Sir A.C. Doyle: The Adventure of the Beryl Coronet
[2] Sir A.C. Doyle: The Boscombe Valley Mystery

Example: TENEX bug.

- TENEX: operating system for PDP-10 in 1970s.

```
int i;
for (i=0; i<8; i++) {
  if (password[i] != trial[i]) break;
}
if (i < 8) return( ERROR );
```

Example: TENEX bug.

- TENEX: operating system for PDP-10 in 1970s.

```
int i;
for (i=0; i<8; i++) {
  if (password[i] != trial[i]) break;
}
if (i < 8) return( ERROR );
```

- Suppose `password` is `password` and trial is `aaaaaaaa`. We deduce that the first letter is not a.

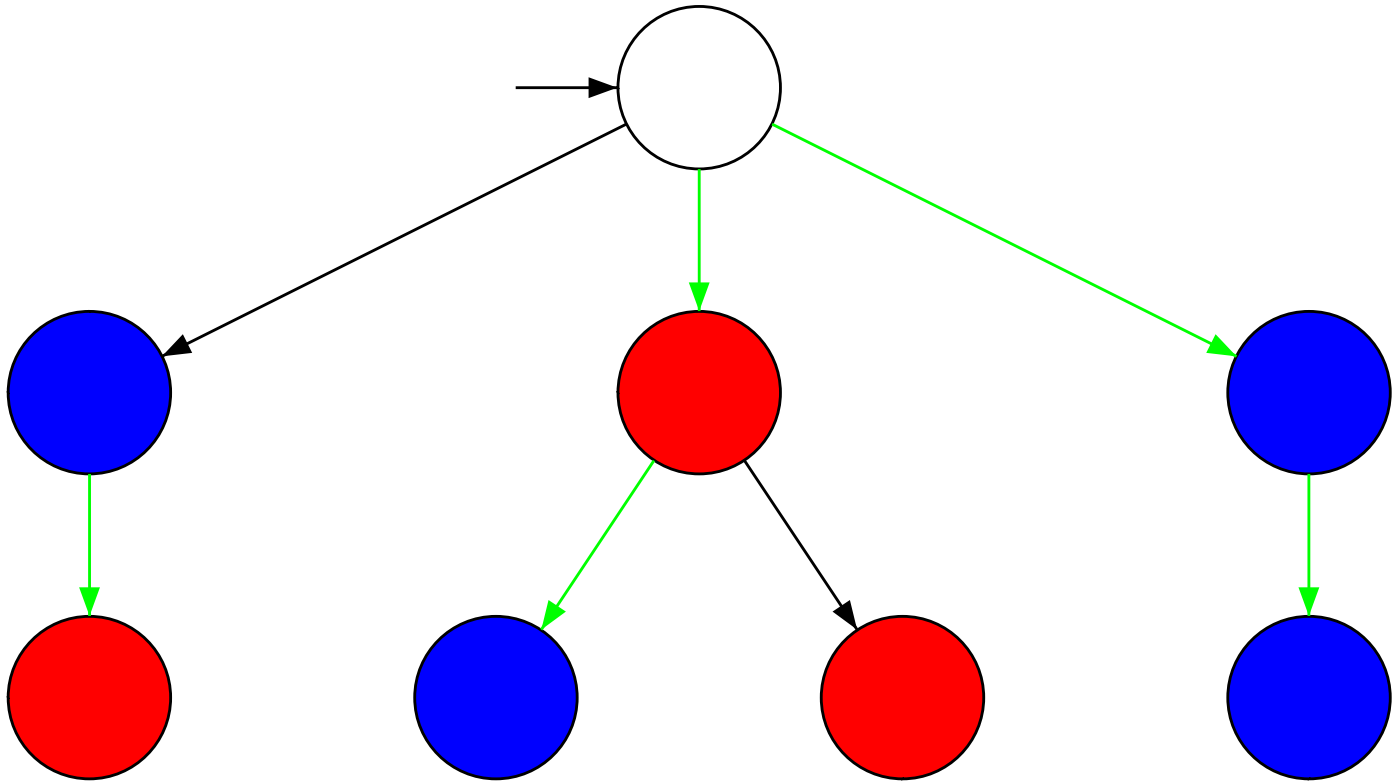
Example: TENEX bug.

- TENEX: operating system for PDP-10 in 1970s.

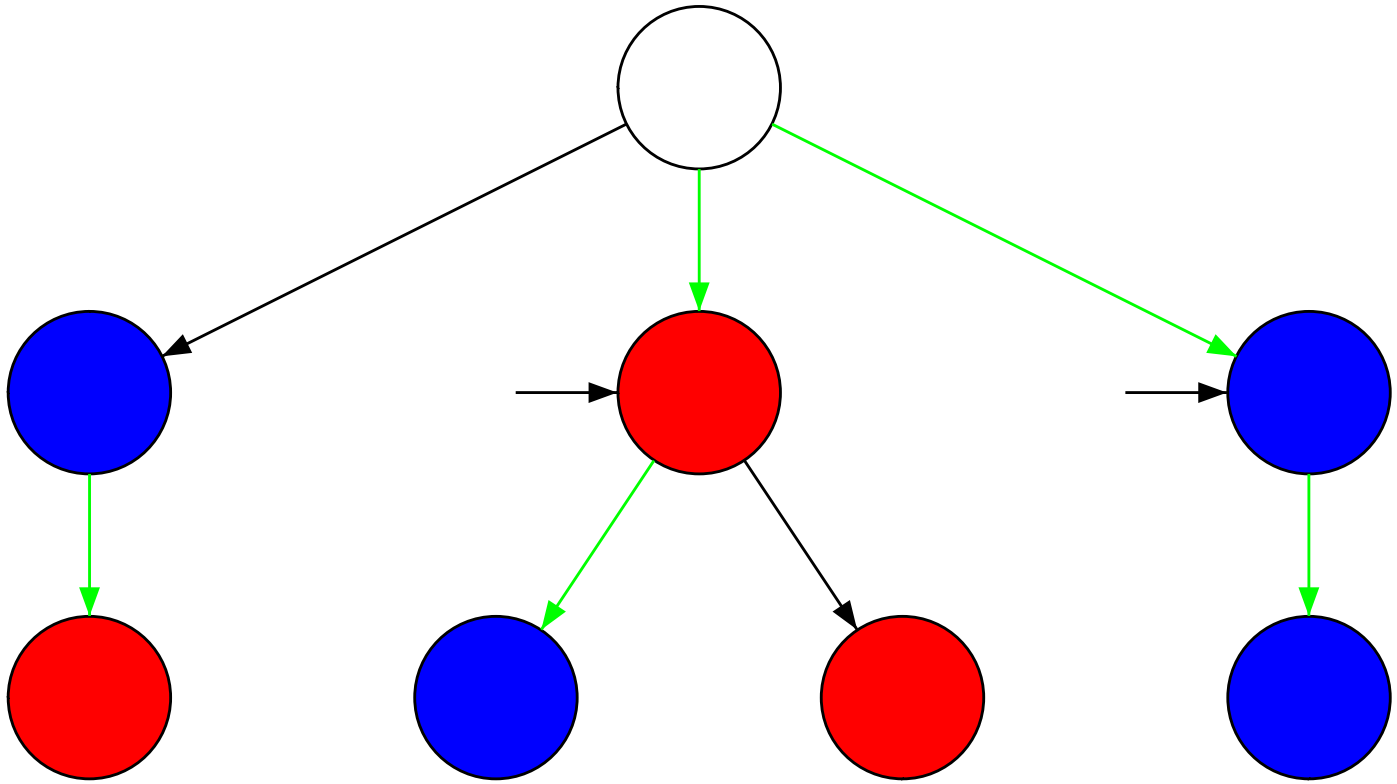
```
int i;
for (i=0; i<8; i++) {
  if (password[i] != trial[i]) break;
}
if (i < 8) return( ERROR );
```

- Suppose `password` is `password` and trial is `aaaaaaaa`. We deduce that the first letter is not a.
- Exploited using paging.

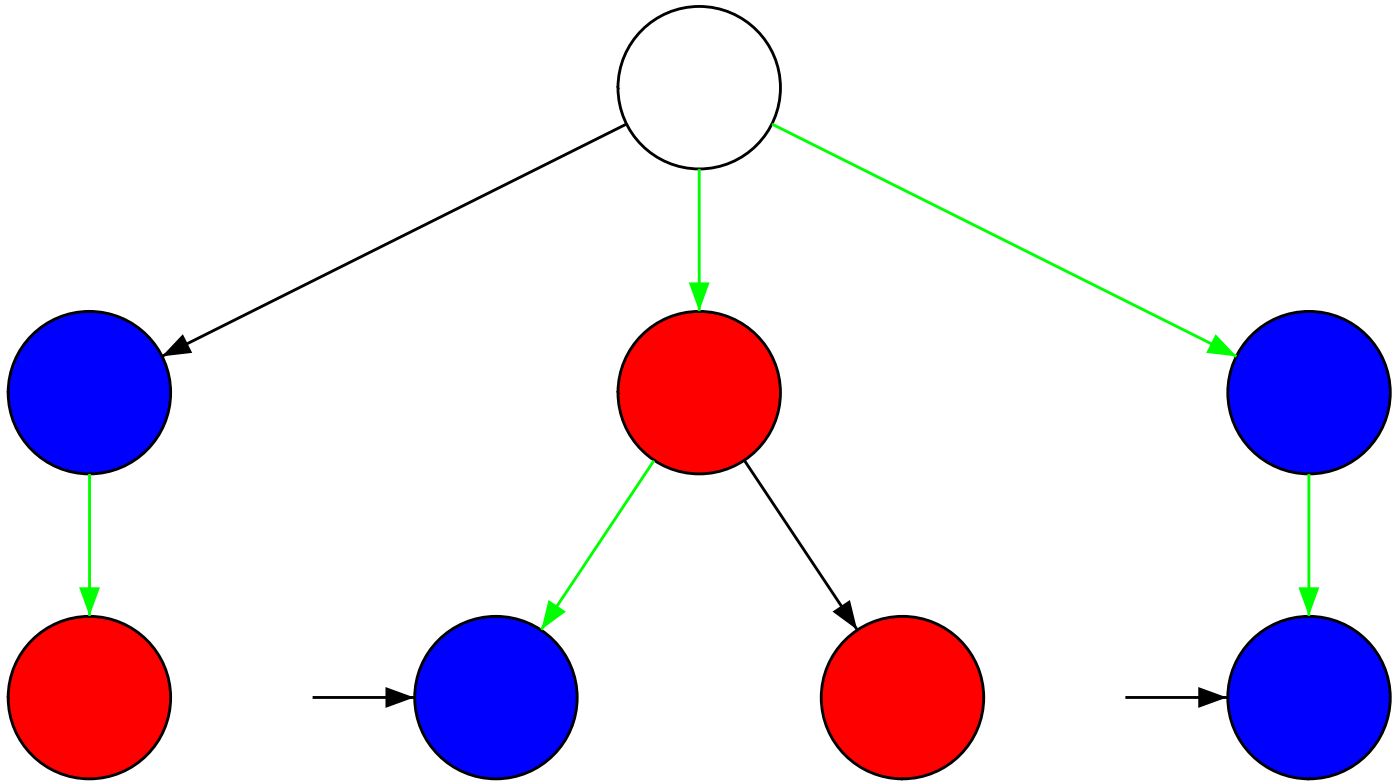
Observations and Secrets.



Observations and Secrets.



Observations and Secrets.



Specifying and Verifying Secrecy

Problem

- Secrecy is not a predicate on a single trace (i.e. not a safety or liveness property).
- Secrecy not expressible in μ-calculus.

Solutions

1. Secrecy-preserving refinements. [submitted; joint work with R. Alur and S. Zdancewic]
2. Logic for secrecy. [current work]

Specifying and Verifying Secrecy

Problem

- Secrecy is not a predicate on a single trace (i.e. not a safety or liveness property).
- Secrecy not expressible in μ-calculus.

Solutions

1. Secrecy-preserving refinements. [submitted; joint work with R. Alur and S. Zdancewic]
2. Logic for secrecy. [current work]

Specifying and Verifying Secrecy

Problem

- Secrecy is not a predicate on a single trace (i.e. not a safety or liveness property).
- Secrecy not expressible in μ-calculus.

Solutions

1. Secrecy-preserving refinements. [submitted; joint work with R. Alur and S. Zdancewic]
2. Logic for secrecy. [current work]

Specifying and Verifying Secrecy

Problem

- Secrecy is not a predicate on a single trace (i.e. not a safety or liveness property).
- Secrecy not expressible in μ-calculus.

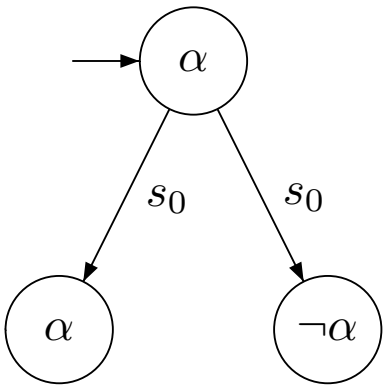
Solutions

1. Secrecy-preserving refinements. [submitted; joint work with R. Alur and S. Zdancewic]
2. Logic for secrecy. [current work]

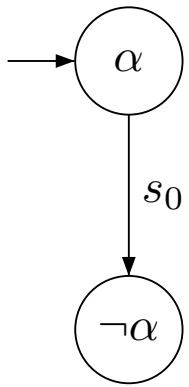
Penn

Standard Refinement.

- Standard refinement: T_{imp} refines T_{spec} iff all behaviors of T_{imp} are allowed by T_{spec}.
- Problem: This notion of refinement does not in general preserve the secrecy of properties.



T_{spec}



T_{imp}

Secrecy-preserving Refinement.

- Secrecy-preserving refinement - implementation leaks a secret only when specification also leaks it.
- A simulation-based proof technique.
  - $Tr(T_{imp}) \subseteq Tr(T_{spec})$
  - $T_{spec} \lesssim T_{imp}$

