Certifying Positive Almost Sure Termination of Probabilistic Pushdown Automata

Tobias Winkler, Joost-Pieter Katoen

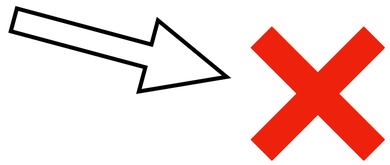
Highlights 2024

Certificates in Verification

Certificates in Verification

System

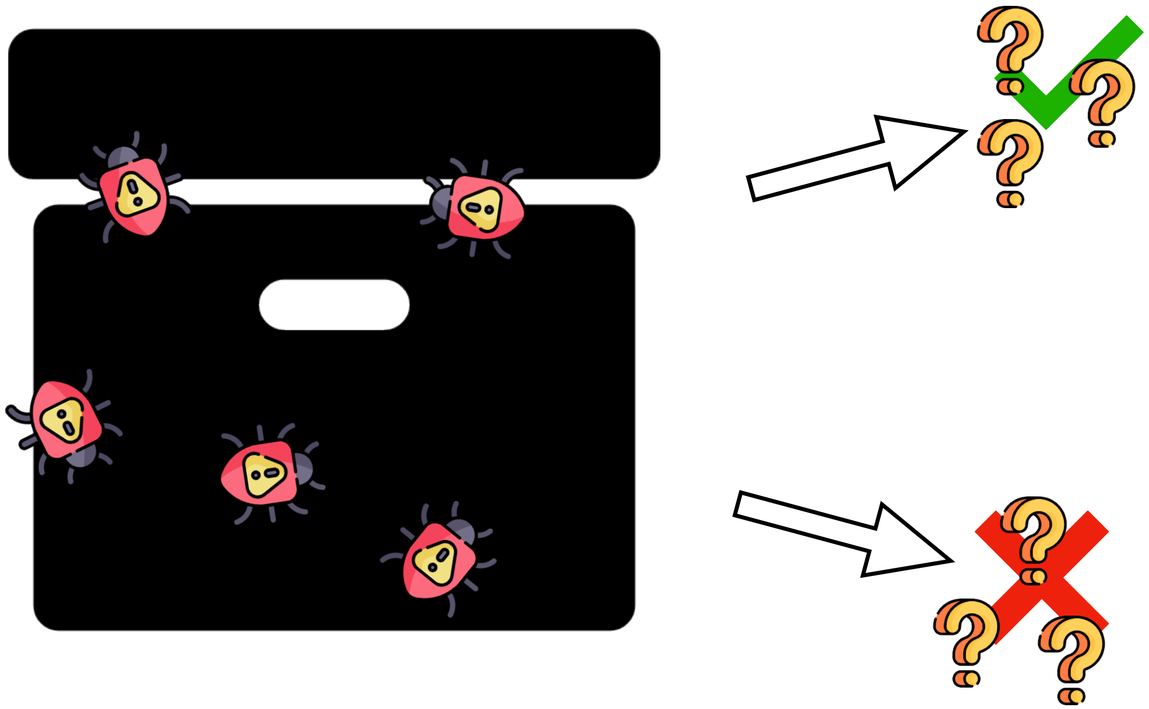
Property



Certificates in Verification

System

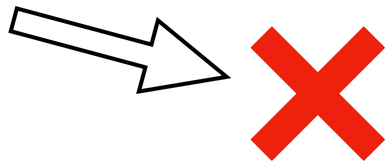
Property



Certificates in Verification

System

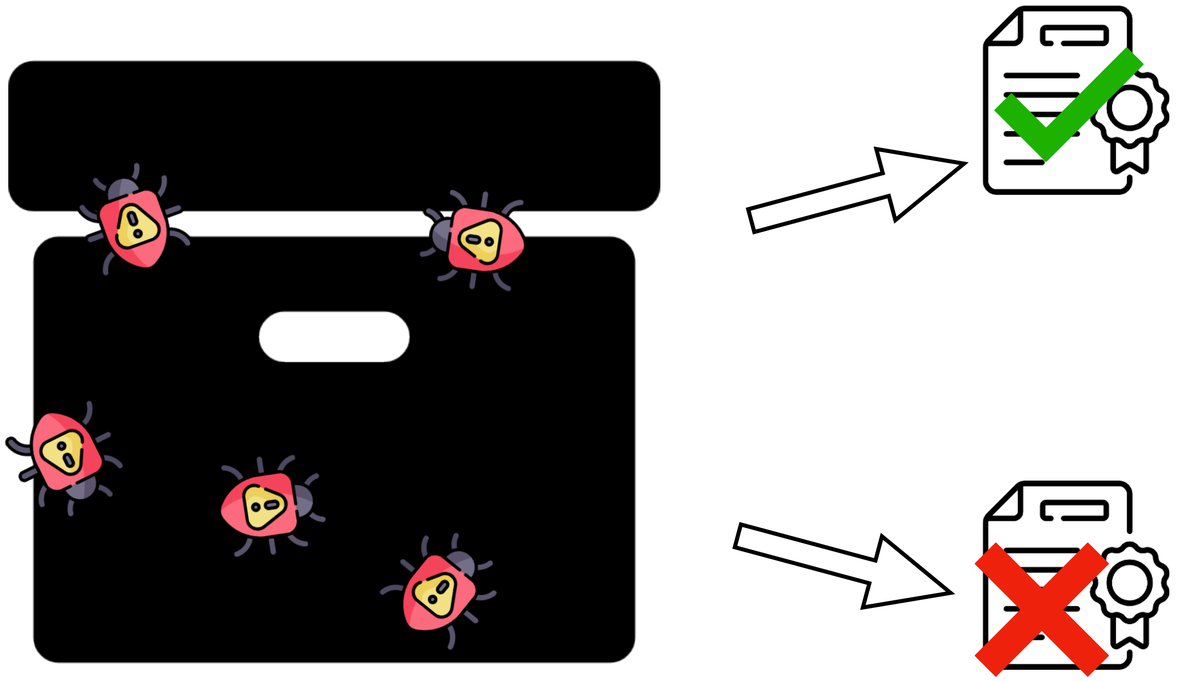
Property



Certificates in Verification

System

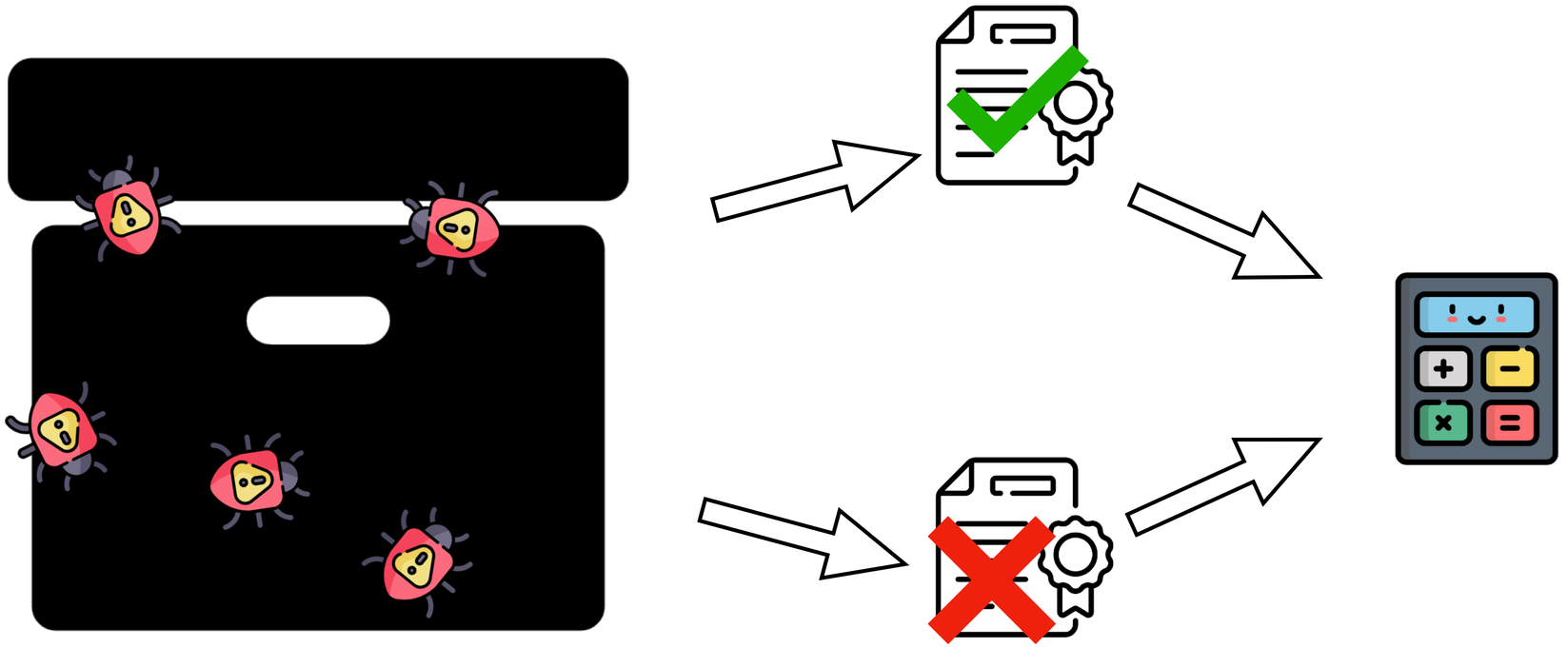
Property



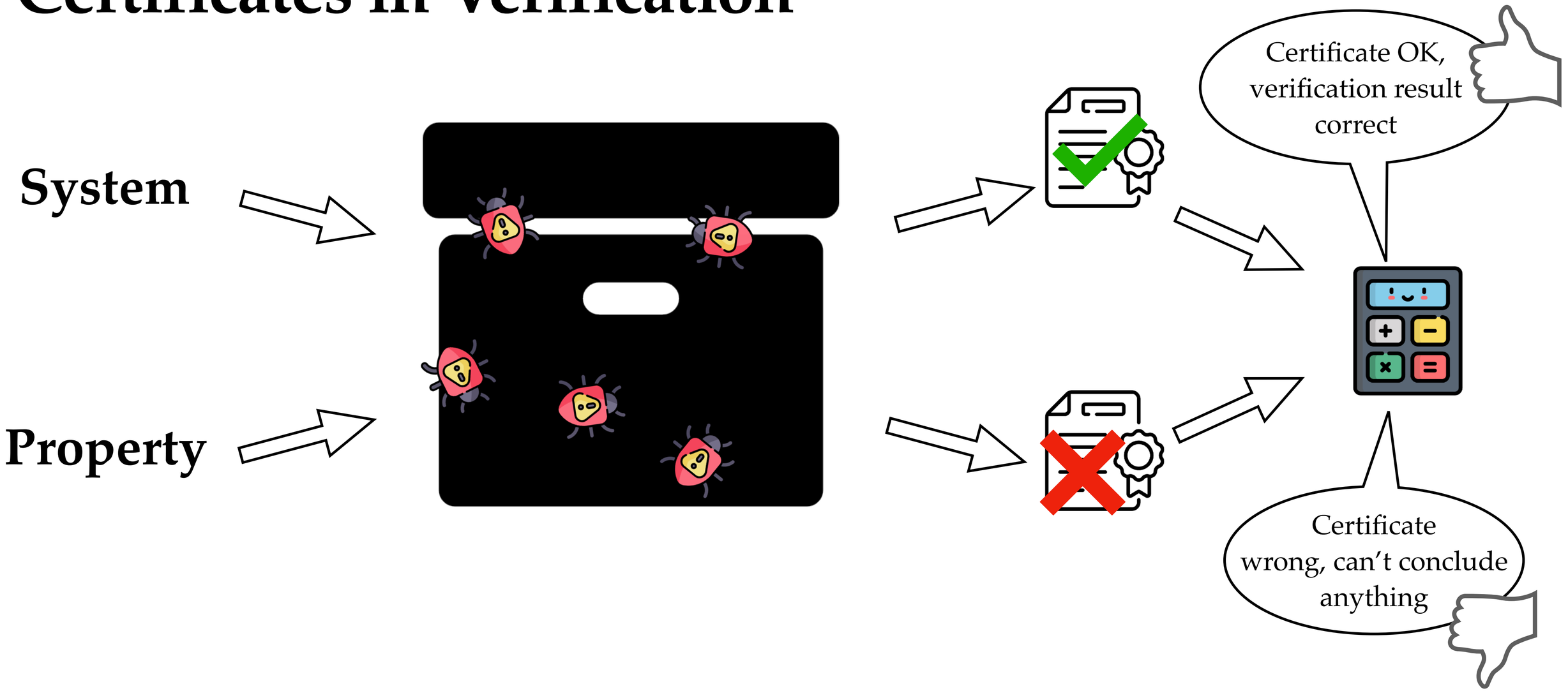
Certificates in Verification

System

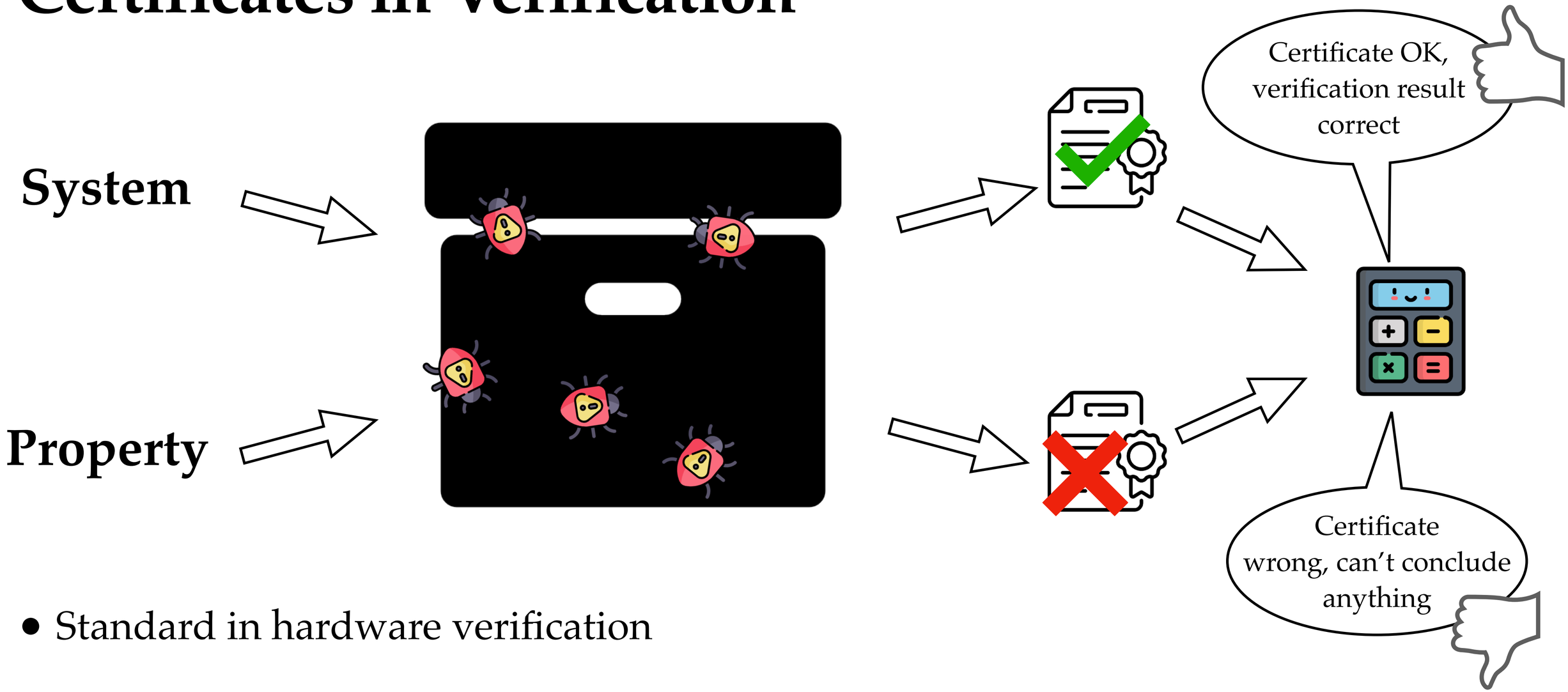
Property



Certificates in Verification

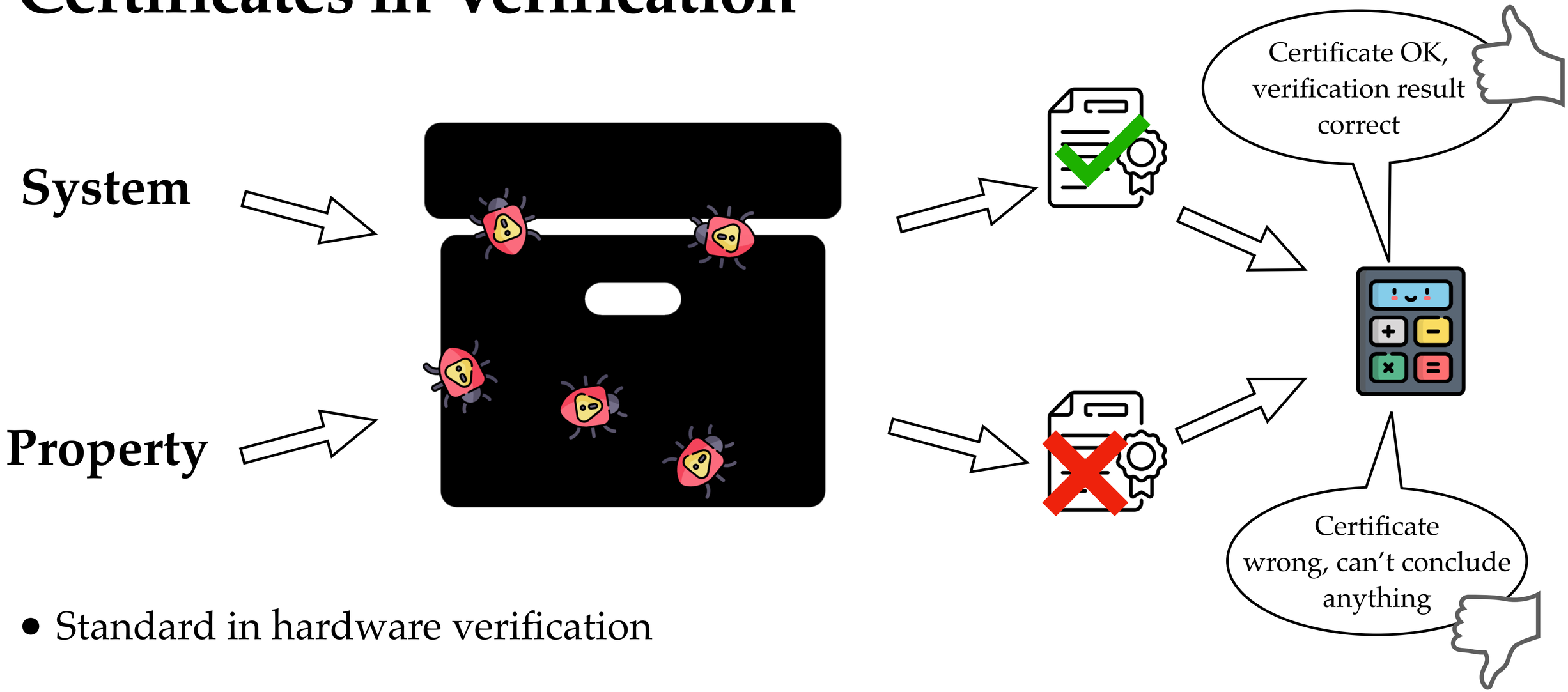


Certificates in Verification



- Standard in hardware verification

Certificates in Verification



- Standard in hardware verification
- Our goal: Certified verification of **probabilistic** systems

Probabilistic Pushdown Automata

[Esparza, Kucera, Mayr, LICS 2004 + 2005]

Probabilistic Pushdown Automata

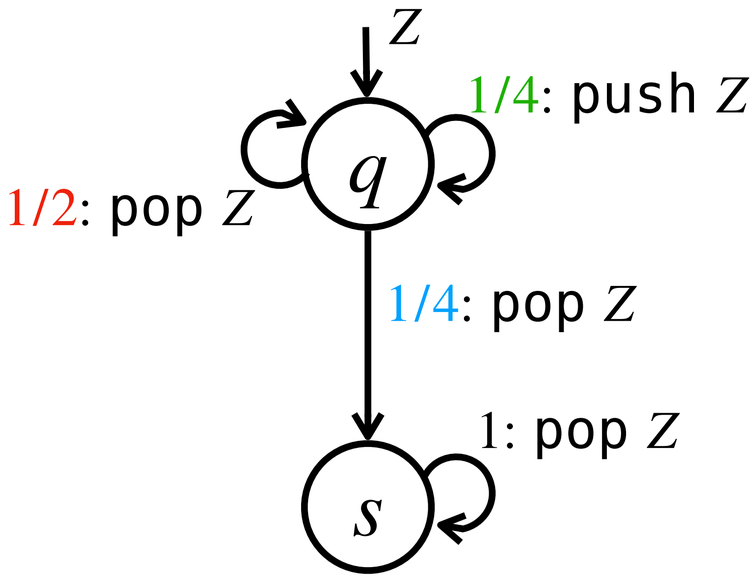
[Esparza, Kucera, Mayr, LICS 2004 + 2005]

Like standard PDA, but probabilities instead of input letters:

Probabilistic Pushdown Automata

[Esparza, Kucera, Mayr, LICS 2004 + 2005]

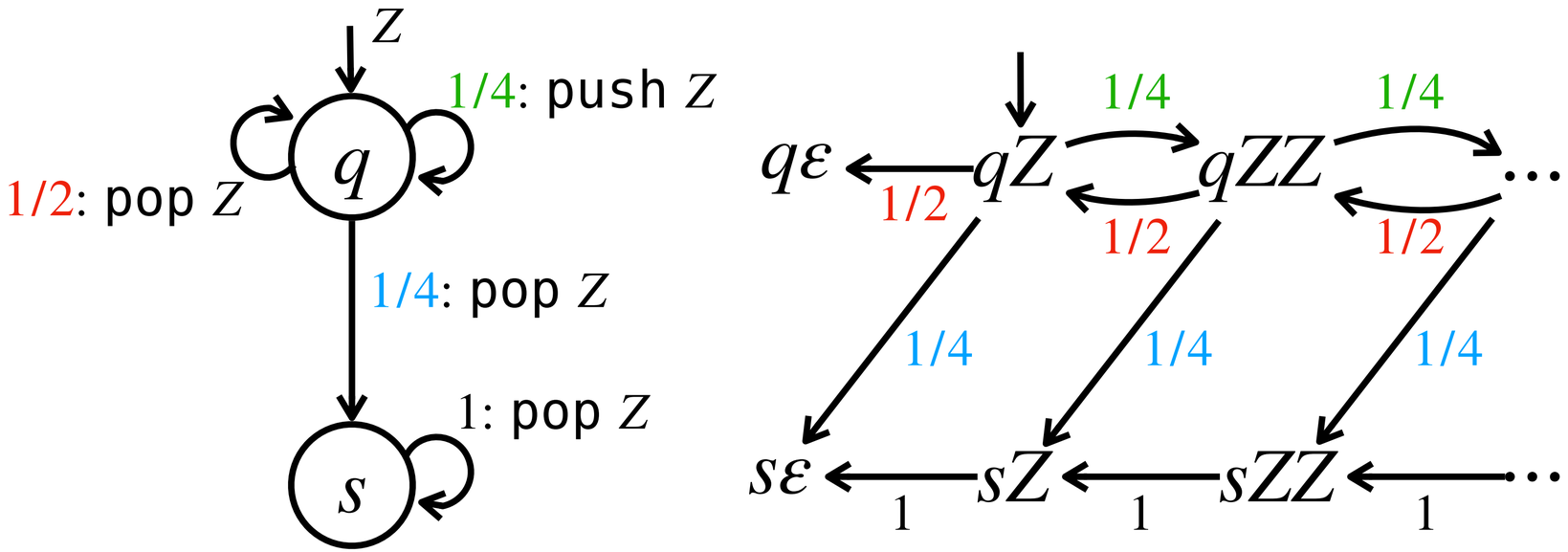
Like standard PDA, but probabilities instead of input letters:



Probabilistic Pushdown Automata

[Esparza, Kucera, Mayr, LICS 2004 + 2005]

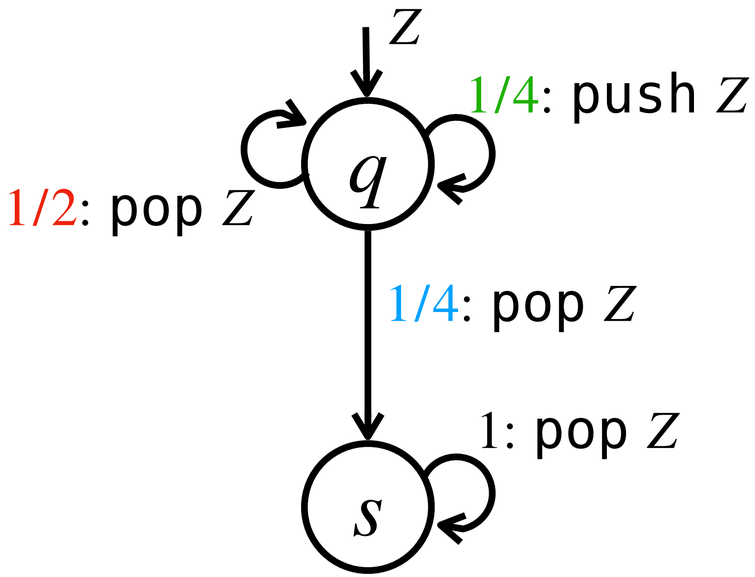
Like standard PDA, but probabilities instead of input letters:

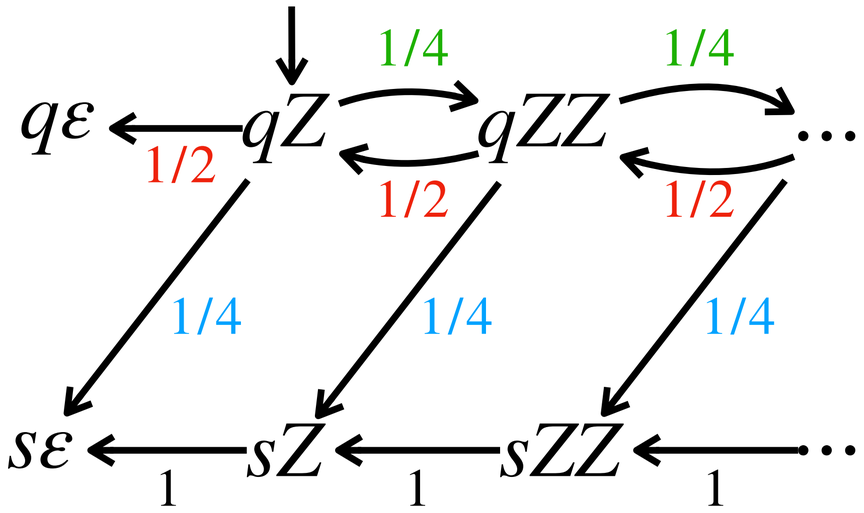


Probabilistic Pushdown Automata

[Esparza, Kucera, Mayr, LICS 2004 + 2005]

Like standard PDA, but probabilities instead of input letters:



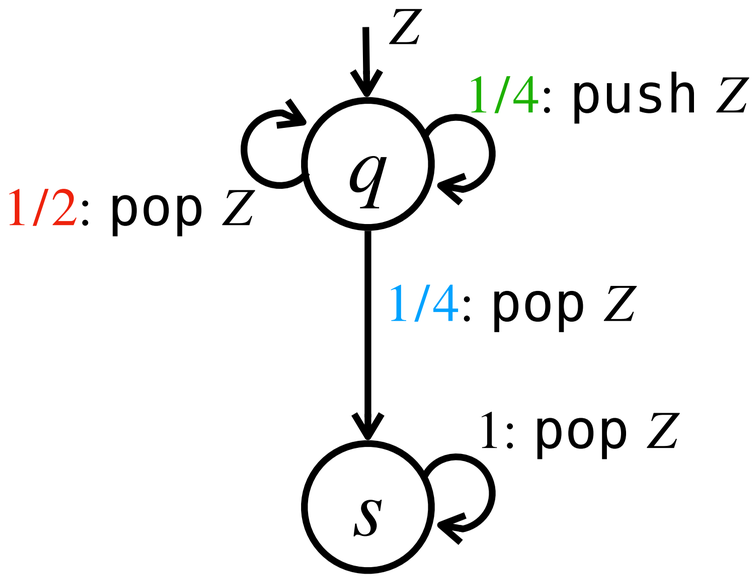


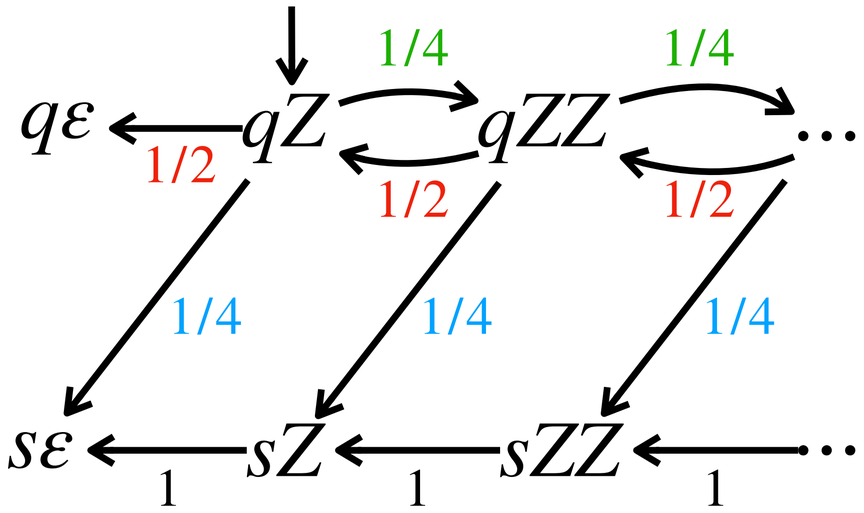

```
bool f(bool b) {
  if(b) {
    choice {
      0.25: return f(f(b));
      0.50: return true;
      0.25: return false;
    }
  } else return false;
}
```

Probabilistic Pushdown Automata

[Esparza, Kucera, Mayr, LICS 2004 + 2005]

Like standard PDA, but probabilities instead of input letters:





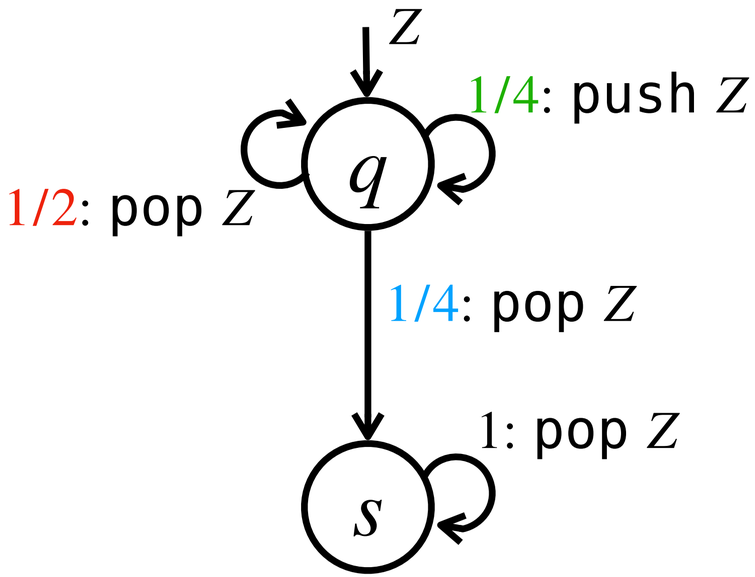
f(true) returns true with probability $2-\sqrt{2}$

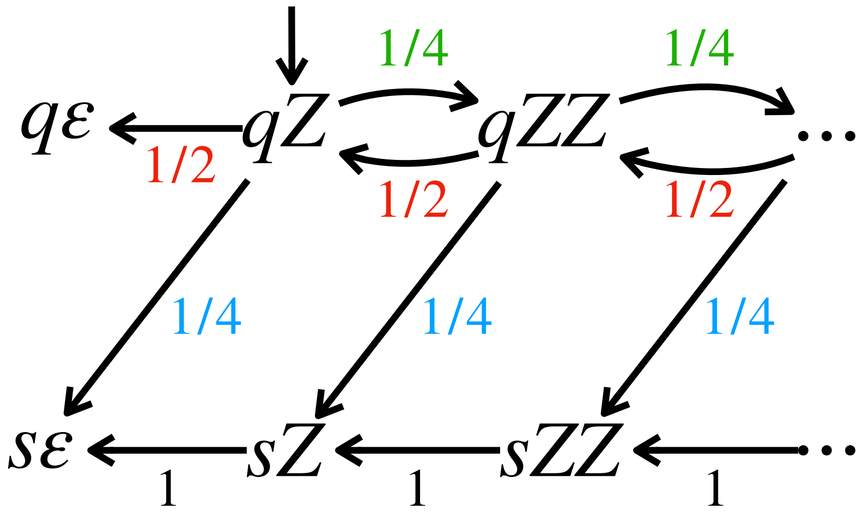
```
bool f(bool b) {
  if(b) {
    choice {
      0.25: return f(f(b));
      0.50: return true;
      0.25: return false;
    }
  } else return false;
}
```

Probabilistic Pushdown Automata

[Esparza, Kucera, Mayr, LICS 2004 + 2005]

Like standard PDA, but probabilities instead of input letters:



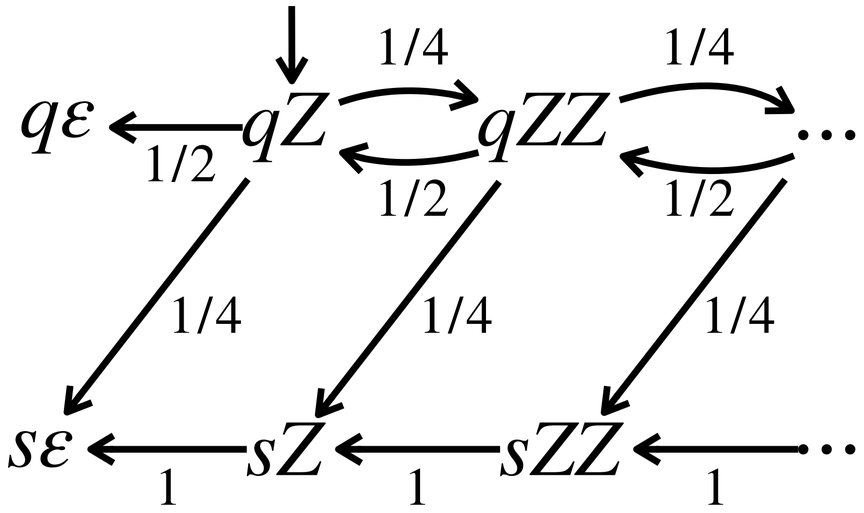


f(true) returns true with probability $2-\sqrt{2}$

```
bool f(bool b) {
  if(b) {
    choice {
      0.25: return f(f(b));
      0.50: return true;
      0.25: return false;
    }
  } else return false;
}
```

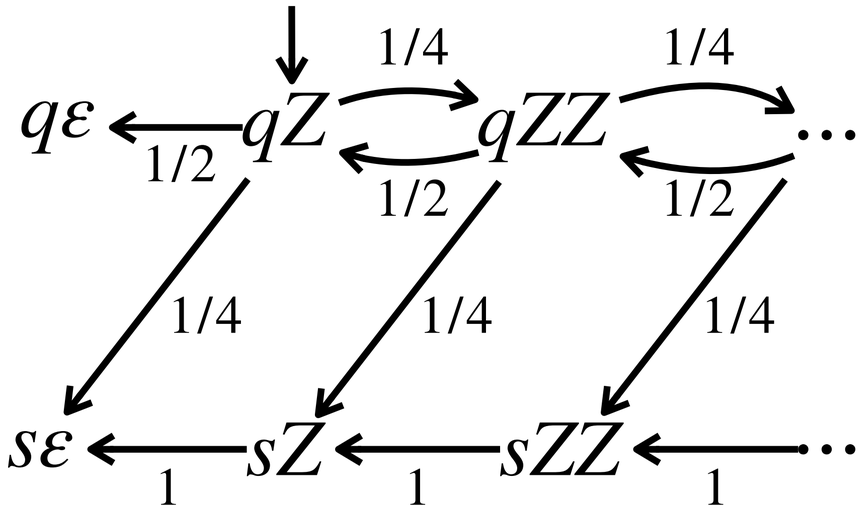
Certificates for **positive almost sure termination** (PAST)
= empty stack is reached after *finitely* many steps *on average*

Certifying Empty Stack Reachability



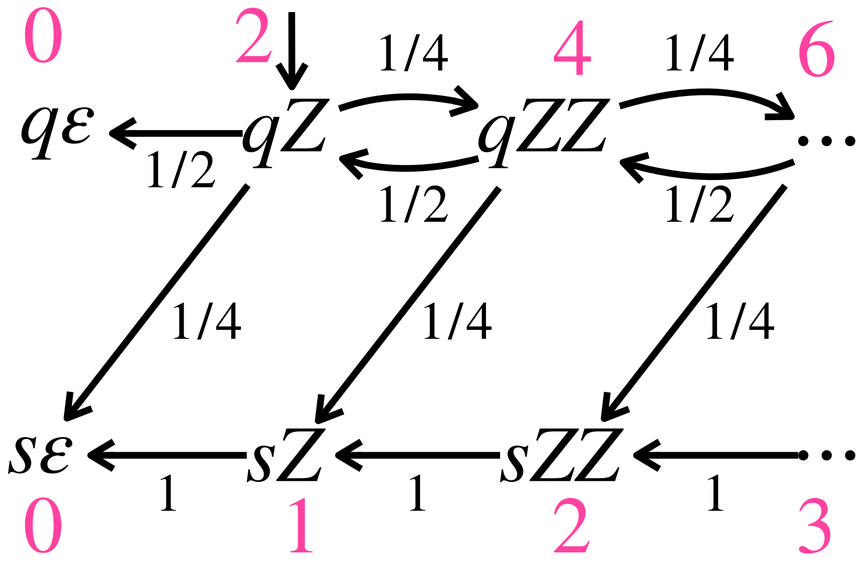
Certifying Empty Stack Reachability

Assign a rank to each configuration:



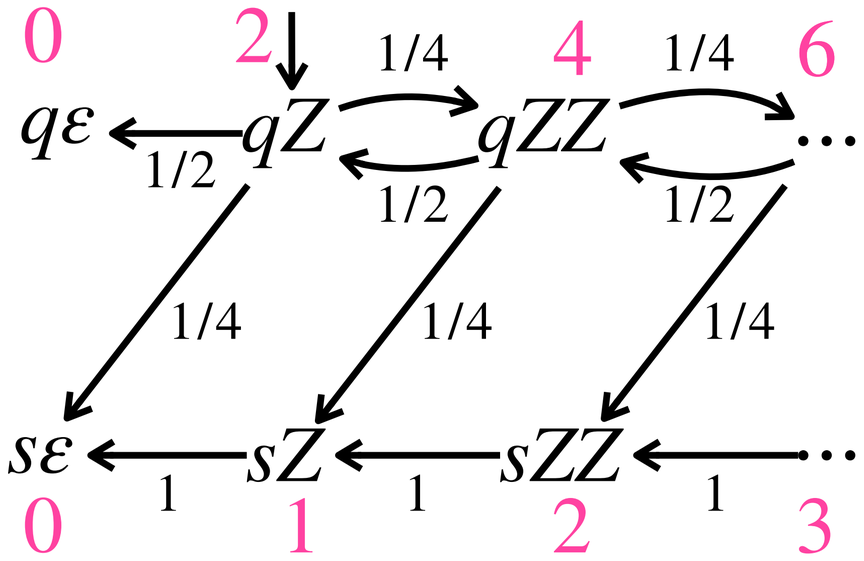
Certifying Empty Stack Reachability

Assign a rank to each configuration:



Certifying Empty Stack Reachability

Assign a rank to each configuration:

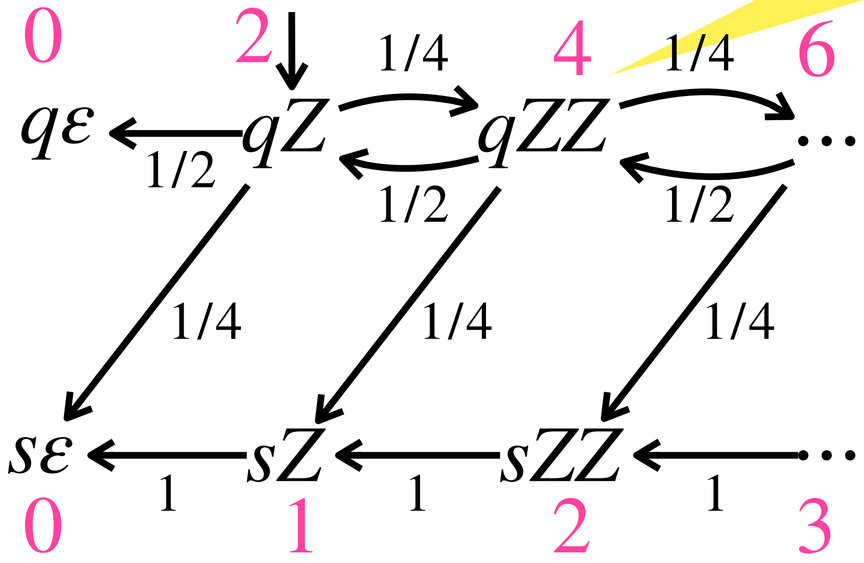


Ranks decrease by ≥ 1 *on average* in one step $\implies$ PAST

Certifying Empty Stack Reachability

Assign a rank to each configuration:

$$\frac{1}{2}\cdot 2+\frac{1}{4}\cdot 6+\frac{1}{4}\cdot 1 = 2.75 < 4$$



Ranks decrease by ≥ 1 *on average* in one step $\implies$ PAST

Finite Representation of ∞-many Ranks?

Finite Representation of ∞-many Ranks?

We only need

- ranks for each configuration qZ — quadratically many
- the probabilities $\mathbb{P}(qZ \xrightarrow{*} s\varepsilon)$ — cubically many

to infer ranks for **all** (∞-many) configurations.

Finite Representation of ∞-many Ranks?

We only need

- ranks for each configuration qZ — quadratically many
- the probabilities $\mathbb{P}(qZ \xrightarrow{*} s\varepsilon)$ — cubically many

to infer ranks for **all** (∞-many) configurations.

This is due to the specific **structure** of the configuration graph generated by a PDA.

Finite Representation of ∞-many Ranks?

We only need

- ranks for each configuration qZ — quadratically many
- the probabilities $\mathbb{P}(qZ \xrightarrow{*} s\varepsilon)$ — cubically many

to infer ranks for **all** (∞-many) configurations.

solution of polynomial equation system

This is due to the specific **structure** of the configuration graph generated by a PDA.

Theorem: Certificates for PAST

[W & Katoen, LICS 2023]

The probabilistic PDA Δ is **PAST**

$$\Longleftrightarrow$$

Theorem: Certificates for PAST

[W & Katoen, LICS 2023]

The probabilistic PDA Δ is **PAST**

$\Longleftrightarrow$

there exist non-negative ranks $\vec{r}$ and probabilities $\vec{p}$ satisfying a system of polynomial inequalities (that can be "read off" from Δ in polynomial time).

Theorem: Certificates for PAST

[W & Katoen, LICS 2023]

The probabilistic PDA Δ is **PAST**

$\Leftrightarrow$

there exist non-negative rational ranks $\vec{r}$ and probabilities $\vec{p}$ satisfying a system of polynomial inequalities (that can be "read off" from Δ in polynomial time).

Theorem: Certificates for PAST

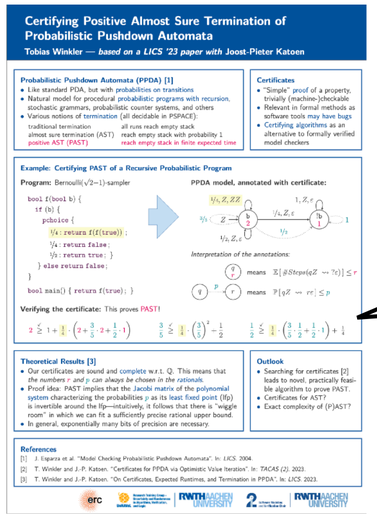
[W & Katoen, LICS 2023]

The probabilistic PDA Δ is **PAST**

$\Leftrightarrow$

there exist non-negative ranks $\vec{r}$ and probabilities $\vec{p}$ satisfying a system of polynomial inequalities (that can be "read off" from Δ in polynomial time).

rational



Check poster for concrete example

Link to paper

images: Flaticon.com