ACM Conference on Computer and Communications Security

Search-based Local Blackbox Deobfuscation: Understand, Improve and Mitigate

Grégoire Menguy – CEA LIST

Sébastien Bardin – CEA LIST

Richard Bonichon – TWEAG I/O

Cauim de Souza Lima – CEA LIST

25/2022

Speaker

Grégoire MENGUY

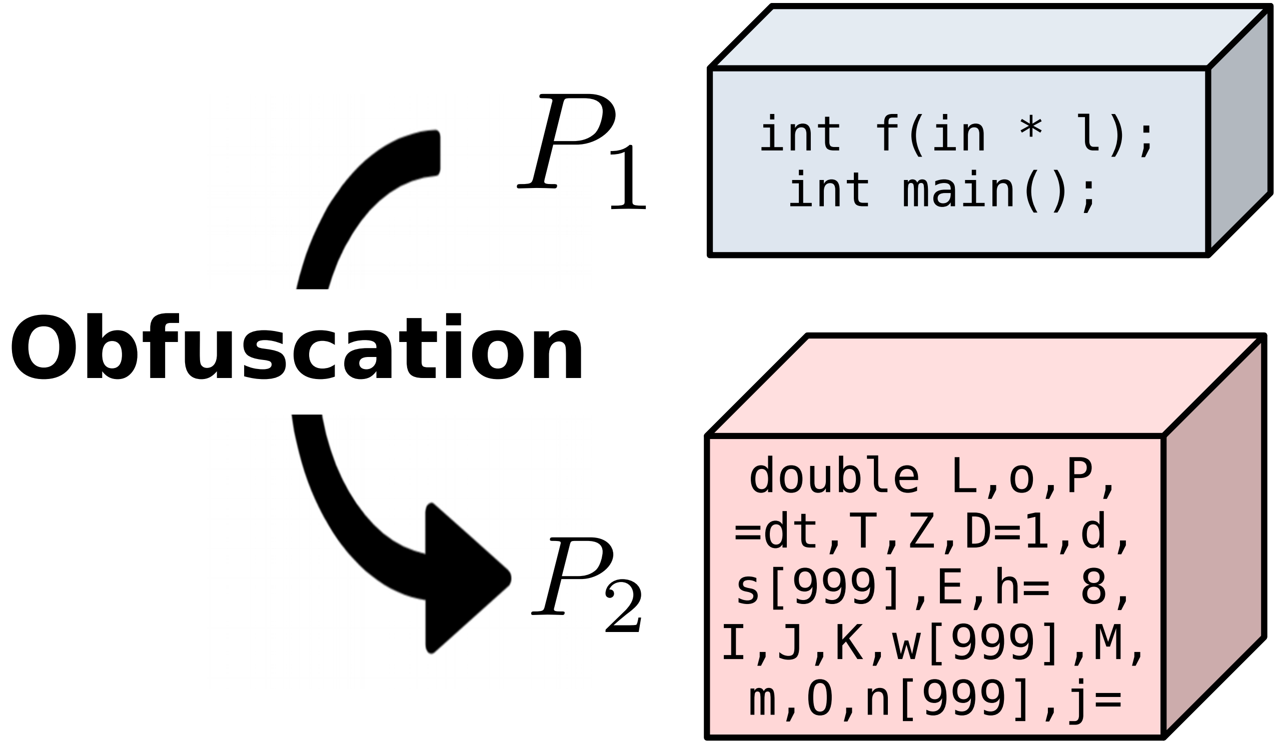
PhD Student at CEA LIST

BINSEC Team (https://binsec.github.io/)

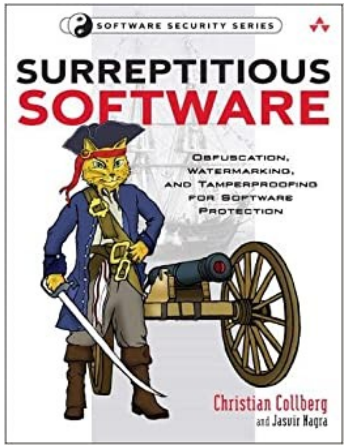
https://www.linkedin.com/in/gregoire-menguy/

@grmenguy

Obfuscation



Obfuscation



P_1

```
int f(in * l);
int main();
```

Obfuscation

P_2

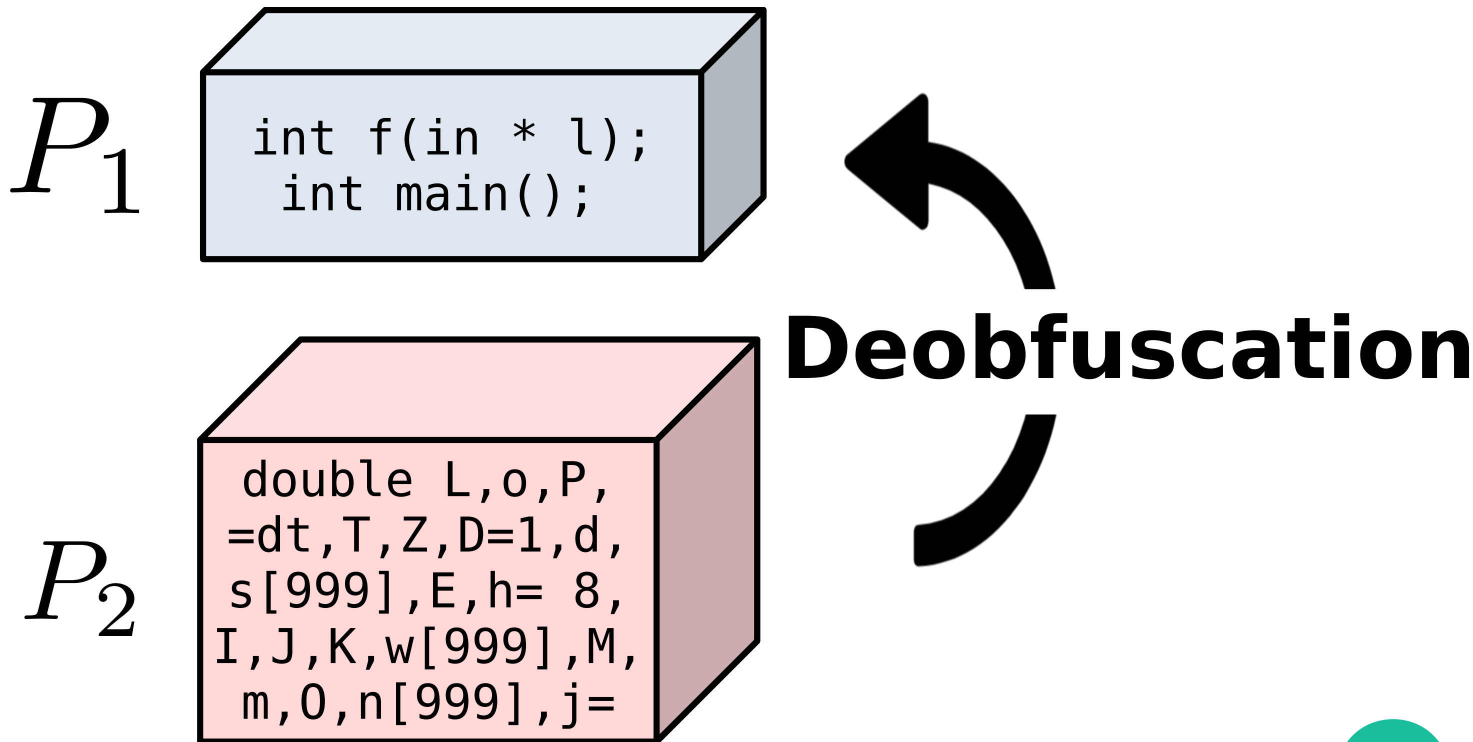
```
double L,o,P,
=dt,T,Z,D=1,d,
s[999],E,h= 8,
I,J,K,w[999],M,
m,O,n[999],j=
```

Epona

cloakware by irdeto

emproof
Safeguard what counts

Deobfuscation



Deobfuscation

Protecting Software through Obfuscation: Can It Keep Pace with Progress in Code Analysis?

SEBASTIAN SCHRITTWIESER, St. Pölten University of Applied Sciences, Austria
STEFAN KATZENBEISSER, Technische Universität Darmstadt, Germany
JOHANNES KINDER, Royal Holloway, University of London, United Kingdom
GEORG MERZDOVNIK and EDGAR WEIPPL, SBA Research, Vienna, Austria

A Generic Approach to Automatic Deobfuscation of Executable Code

Babak Yadegari Brian Johannesmeyer Benjamin Whitely Saumya Debray
Department of Computer Science
The University of Arizona
Tucson, AZ 85721
{babaky, bjohannesmeyer, whitely, debray}@cs.arizona.edu

Symbolic deobfuscation: from virtualized code back to the original*

Jonathan Salwan[1], Sébastien Bardin[2], and Marie-Laure Potet[3]

Backward-Bounded DSE: Targeting Infeasibility Questions on Obfuscated Codes*

Sébastien Bardin, CEA, LIST, 91191 Gif-Sur-Yvette, France, sebastien.bardint@cea.fr
Robin David, CEA, LIST, 91191 Gif-Sur-Yvette, France, robin.david@cea.fr
Jean-Yves Marion, Université de Lorraine, CNRS and Inria, LORIA, France, jean-yves.marion@loria.fr

BINSEC

TRITON
Dynamic Binary Analysis

MIASM

Deobfuscation

Protecting Software through Obfuscation: Can It Keep Pace with Progress in Code Analysis?

SEBASTIAN SCHRITTWIESER, St. Pölten University of Applied Sciences, Austria
STEFAN KATZENBEISSER, Technische Universität Darmstadt, Germany
JOHANNES KINDER, Royal Holloway, University of London, United Kingdom
GEORG MERZDOVNIK and EDGA

Backward-Bounded DSE: Targeting Infeasibility Questions on Obfuscated Codes*

Sébastien Bardin
CEA, LIST,

Robin David
CEA, LIST,

Jean-Yves Marion
Université de Lorraine,
CNRS and Inria, LORIA, France
jean-yves.marion@loria.fr

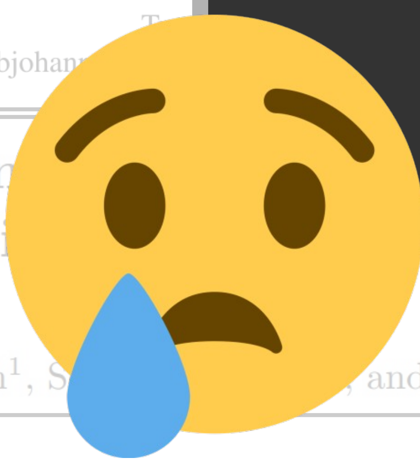
A Generic Approach to Autom

Babak Yadegari Brian Johannesn
Department
The Uni
{babaky, bjohan

Sy
from virtual ... o the original*

Jonathan Salwan[1], S ... and Marie-Laure Potet[3]

Whitebox deobfuscation is highly efficient



Whitebox Deobfuscation

But efficient countermeasures

Information Hiding in Software with Mixed Boolean-Arithmetic Transforms

Yongxin Zhou, Alec Main, Yuan X. Gu, and Harold Johnson

Cloakware Inc., USA
{yongxin.zhou,alec.main,yuan.gu,harold.johnson}@cloakware.com

How to Kill Symbolic Deobfuscation for Free
(or: Unleashing the Potential of Path-Oriented Protections)

Mathilde Ollivier
CEA, LIST,
Paris-Saclay, France
mathilde.ollivier2@cea.fr

Sébastien Bardin
CEA, LIST,
Paris-Saclay, France
sebastien.bardin@cea.fr

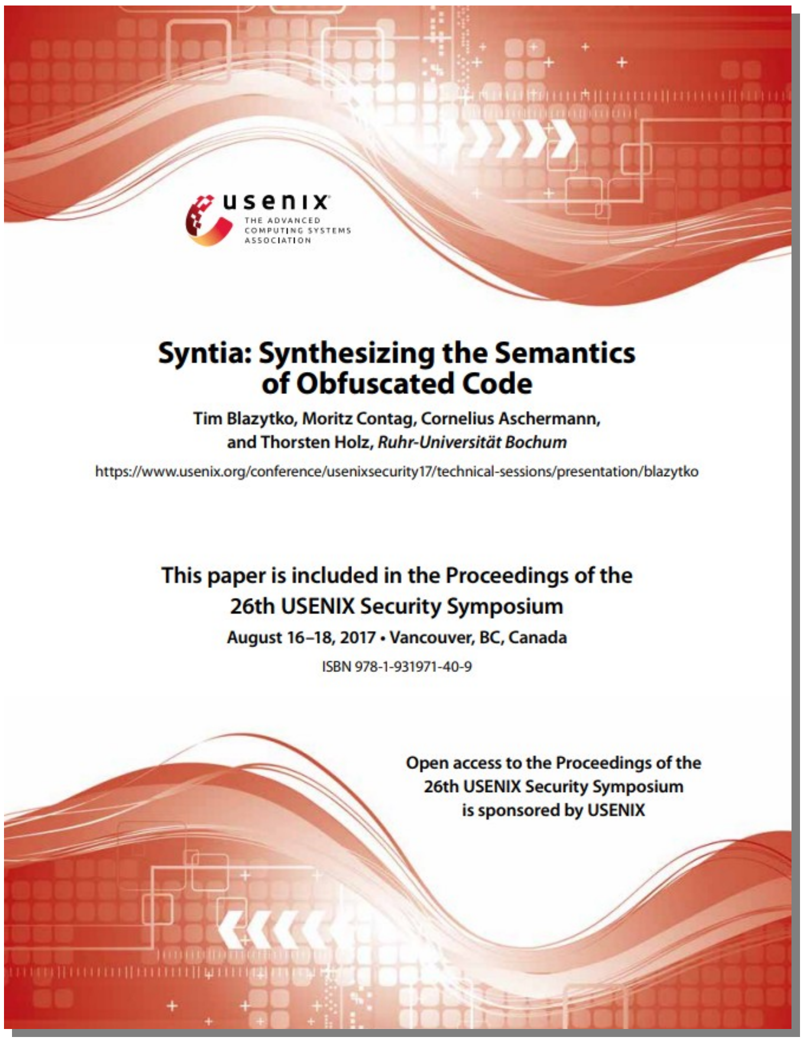
Richard Bonichon
CEA, LIST,
Paris-Saclay, France
richard.bonichon@cea.fr

Jean-Yves Marion
Université de Lorraine, CNRS, LORIA
Nancy, France
Jean-Yves.Marion@loria.fr

Probabilistic Obfuscation through Covert Channels

Jon Stephens Babak Yadegari Christian Collberg Saumya Debray Carlos Scheidegger
Department of Computer Science
The University of Arizona
Tucson, AZ 85721, USA
Email: {stephensj2, babaky, collberg, debray, cscheid}@cs.arizona.edu

New threat: Blackbox Deobfuscation


usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Syntia: Synthesizing the Semantics of Obfuscated Code

Tim Blazytko, Moritz Contag, Cornelius Aschermann, and Thorsten Holz, *Ruhr-Universität Bochum*

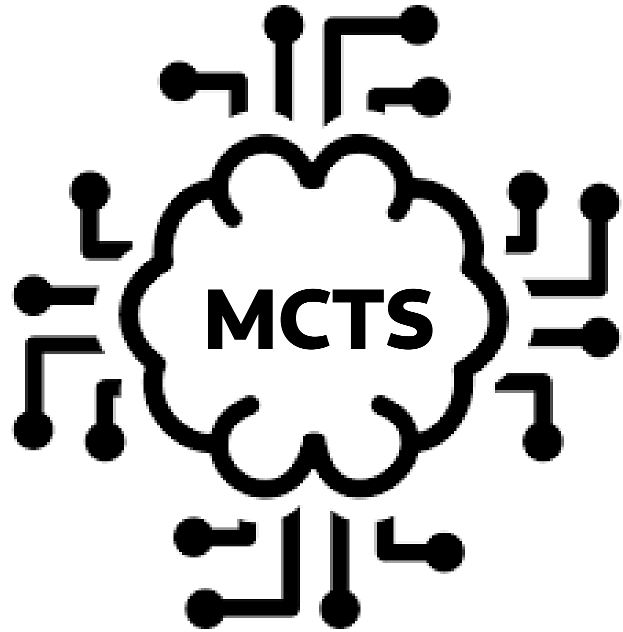
https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/blazytko

This paper is included in the Proceedings of the 26th USENIX Security Symposium

August 16–18, 2017 • Vancouver, BC, Canada

ISBN 978-1-931971-40-9

Open access to the Proceedings of the 26th USENIX Security Symposium is sponsored by USENIX



Bypasses whitebox methods limitations

Open questions

Understand

- Strength ?
- Weaknesses ?
- Why ?

Improve

- Why MCTS ?
- Can be improved?
- Impacted by SoA protections?

Mitigate

- How to protect ?

Contributions

Understand

- Propose missing formalization
- Refine Syntia experiments: new strengths and weaknesses
- Show and explain why MCTS is not appropriate

Improve

- S-metaheuristics > MCTS
- Implement our approach: Xyntia
- Evaluation of Xyntia

Mitigate

- Propose 2 protections
- Evaluate them against Xyntia and Syntia

The talk in a nutshell

I. Blackbox deobfuscation : what’s that ?

II. Deepen understanding

III. Improve state-of-the art

IV. Mitigate

Blackbox deobfuscation : what’s that ?

Blackbox deobfuscation

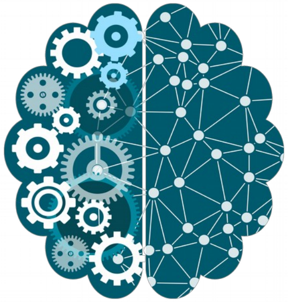
1) Sample

(t = 1, T = 2)
(t = 2, T = 5)
(t = 0, T = 6)
...

-1
-3
-6
...

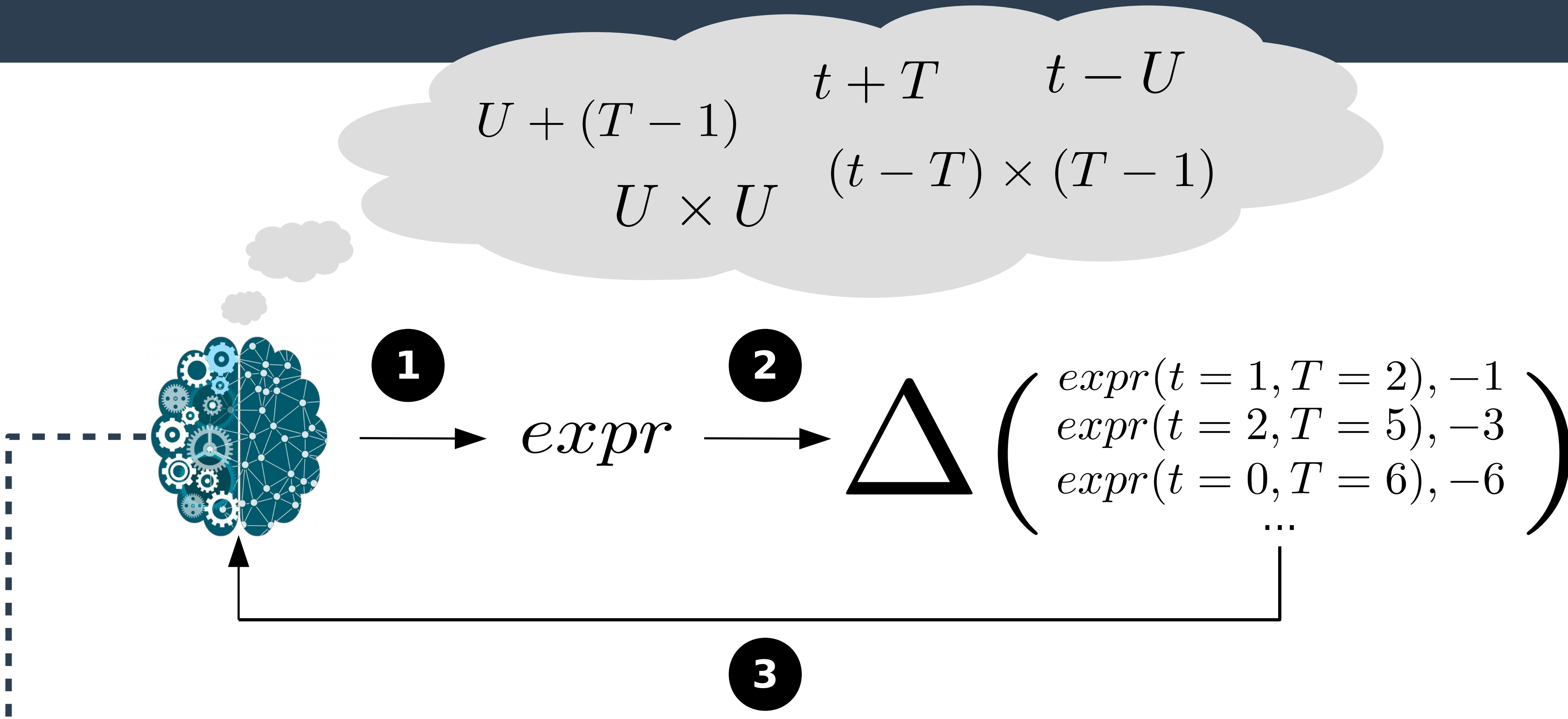
2) Learn

(t = 1, T = 2) → -1
(t = 2, T = 5) → -3
(t = 0, T = 6) → -6
...



$$t - T$$

Learning engine



Expression Grammar

U := U + U | U – U | U * U …
| t | T | 1

Why blackbox?

Given a language *L* and an expression *"e"* in *L*

Syntactic complexity	Semantic complexity
Size of the the expression *"e"*	Size of the smallest expression in *L* equivalent to *"e"*

Example

$t - T$ is syntactically simpler than $(t \vee -2T) \times 2 - (t \oplus -2T) + T$

but they share the same semantic complexity (being equivalent)

Why blackbox ?

Given a language *L* and an expression *“e”* in *L*

Syntactic complexity	Semantic complexity
Size of the the expression *“e”*	Size of the smallest expression in *L* equivalent to *“e”*

Example

$t - T$ is syntactically simpler than $(t \vee -2T) \times 2 - (t \oplus -2T) + T$

but they share the same semantic complexity (being equivalent)

Obfuscation increase syntactic complexity

→ No impact on blackbox methods

Understand

Zoom on SoA: Syntia

- **Dig into Syntia and deepen its evaluation:**
  - RQ1: stability of Syntia
  - **RQ2: efficiency of Syntia**
  - RQ3: Impact of operators set

Syntia: new results

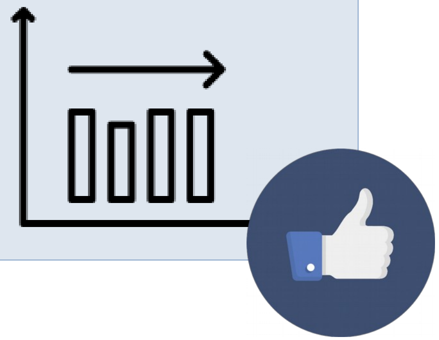
Stable

Quality

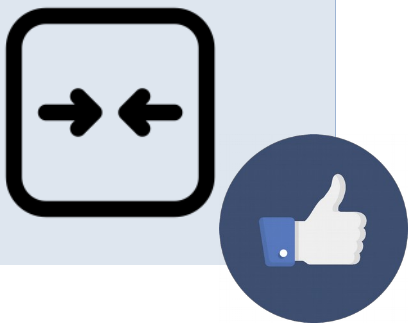
Correctness

Syntia: new results

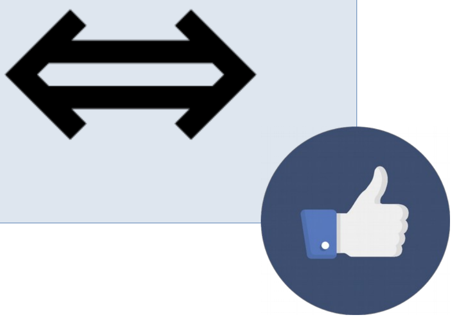
Stable



Quality



Correctness



Speed

Robustness

Experimental design

B1 (Syntia)

- 500 expressions
- Use up to 3 inputs
- **redundancy**
- Unbalanced w.r.t. type

B2 (ours)

- 1110 expressions
- Use 2 - 6 inputs
- **No redundancy**
- Balanced w.r.t. type

	Type			# Inputs				
	Bool.	Arith.	MBA	2	3	4	5	6
#Expr.	370	370	370	150	600	180	90	90

Table 1: Distribution of samples in benchmark B2

Evaluation of Syntia

B1 (Syntia)

- With a 60 s/expr. timeout : 75% of success rate
- With a 1 h/expr. timeout : 88.2% of success rate
- With a **12 h/expr. timeout : 97.6 % of success rate**

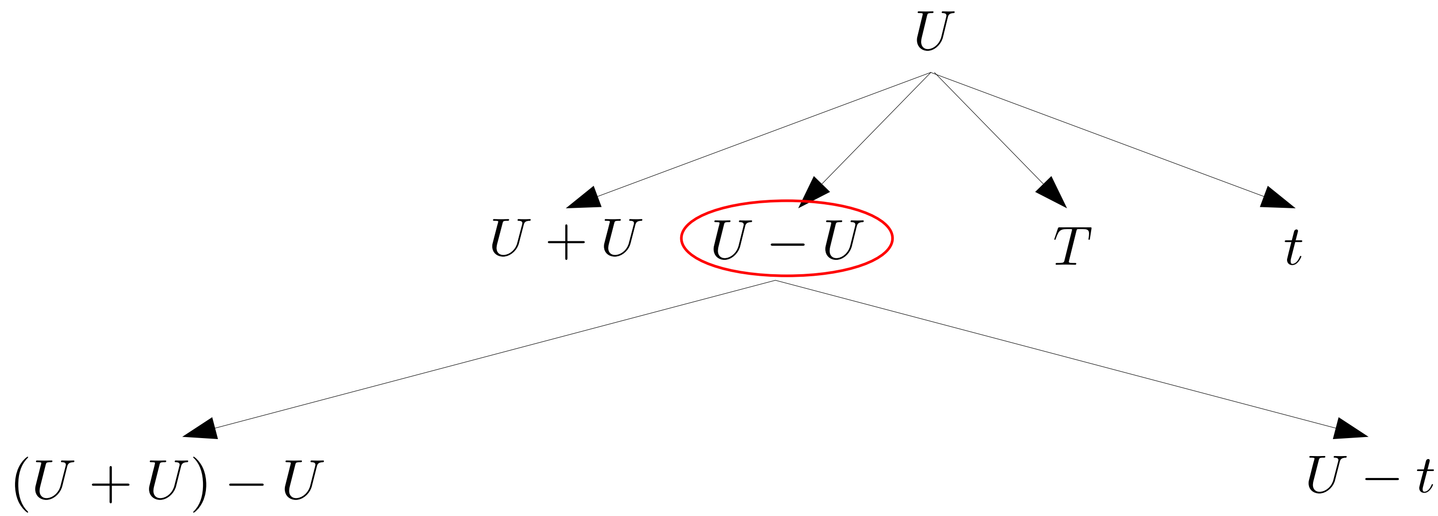
B2 (Ours)

Table 2: Syntia depending on the timeout per expression (B2)

	1s	10s	60s	600s
Succ. Rate	16.5%	25.6%	34.5%	42.3%
Equiv. Range	16.3%	25.1 - 25.3%	33.7 - 34.0%	41.4 - 41.6%
Mean Qual	0.35	0.49	0.59	0.67

Why ?

$t - T$



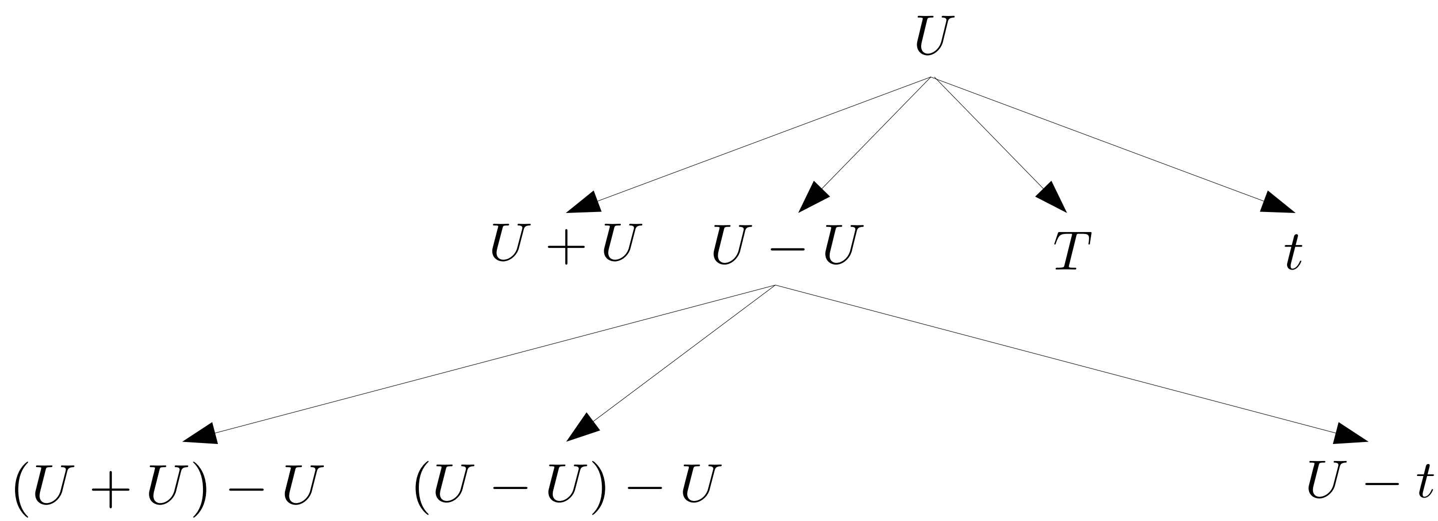
$\Delta\Big($ *Observed Samples:*

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **10**
(t = 10, T = 5) → **5**

; $\Big)$

Why ?

$t - T$



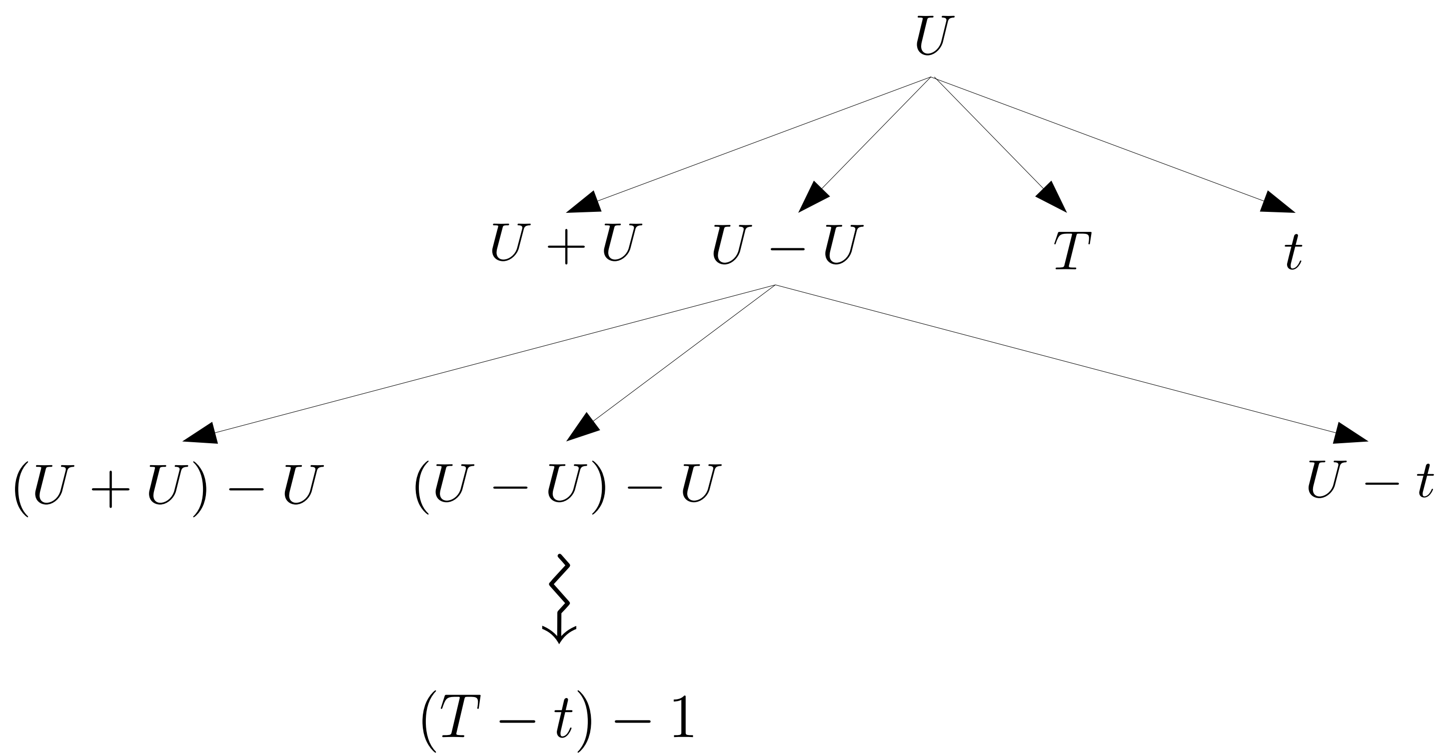
$\Delta\Big($ *Observed Samples:*

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **10**
(t = 10, T = 5) → **5**

; $\Big)$

Why ?

$t - T$



Δ (

Observed Samples:

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **10**
(t = 10, T = 5) → **5**

;

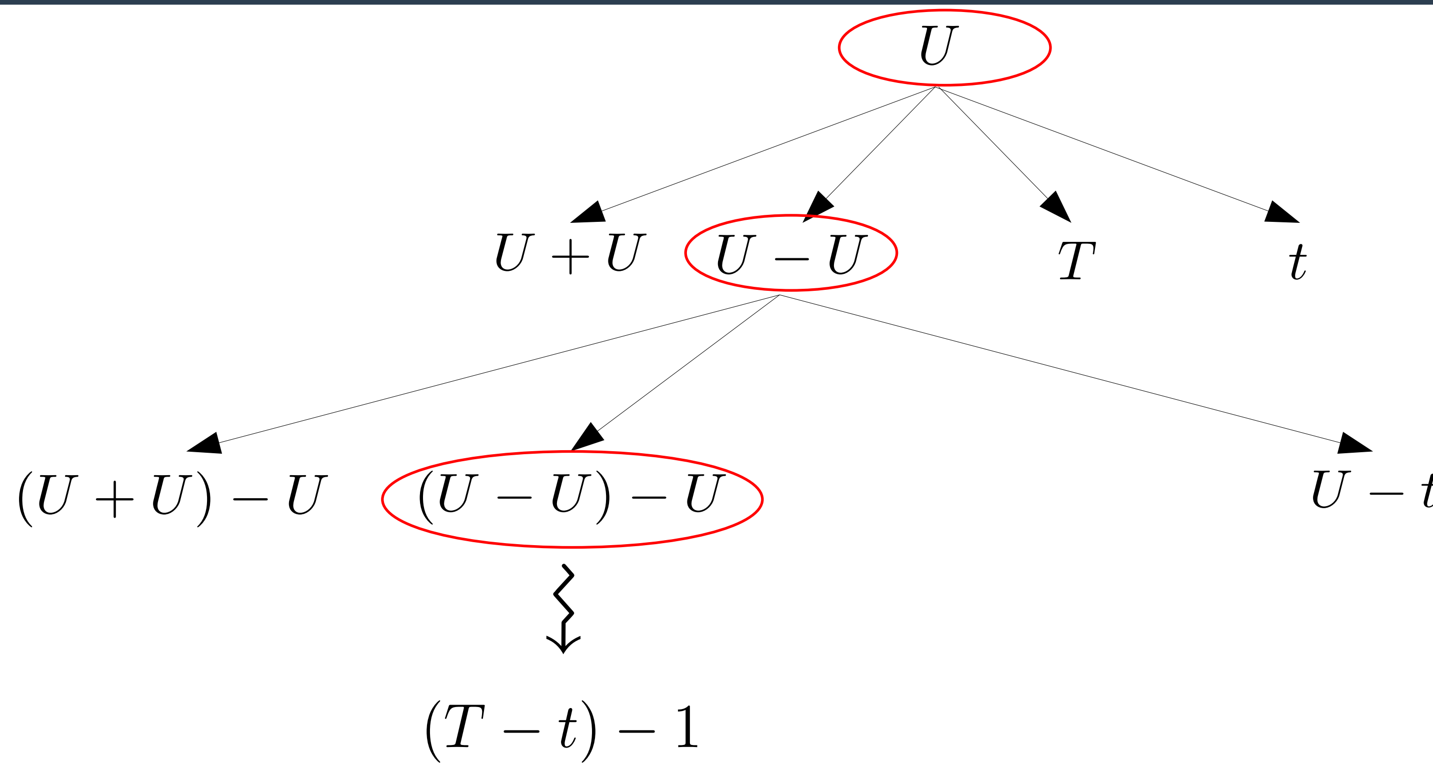
Synthesized Samples:

(t = 1, T = 2) → **0**
(t = 10, T = 0) → **-11**
(t = 10, T = 5) → **-6**

)

Why ?

$\boxed{t - T}$



$\Delta\Big($

Observed Samples:

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **10**
(t = 10, T = 5) → **5**

;

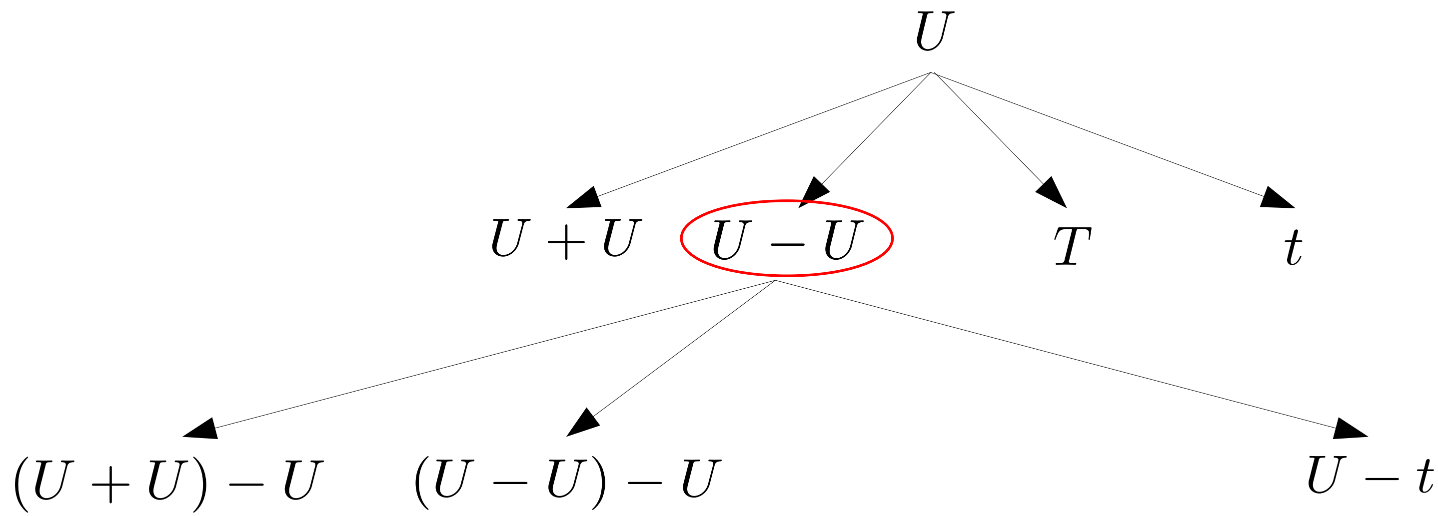
Synthesized Samples:

(t = 1, T = 2) → **0**
(t = 10, T = 0) → **-11**
(t = 10, T = 5) → **-6**

$\Big)$

Why ?

$t - T$



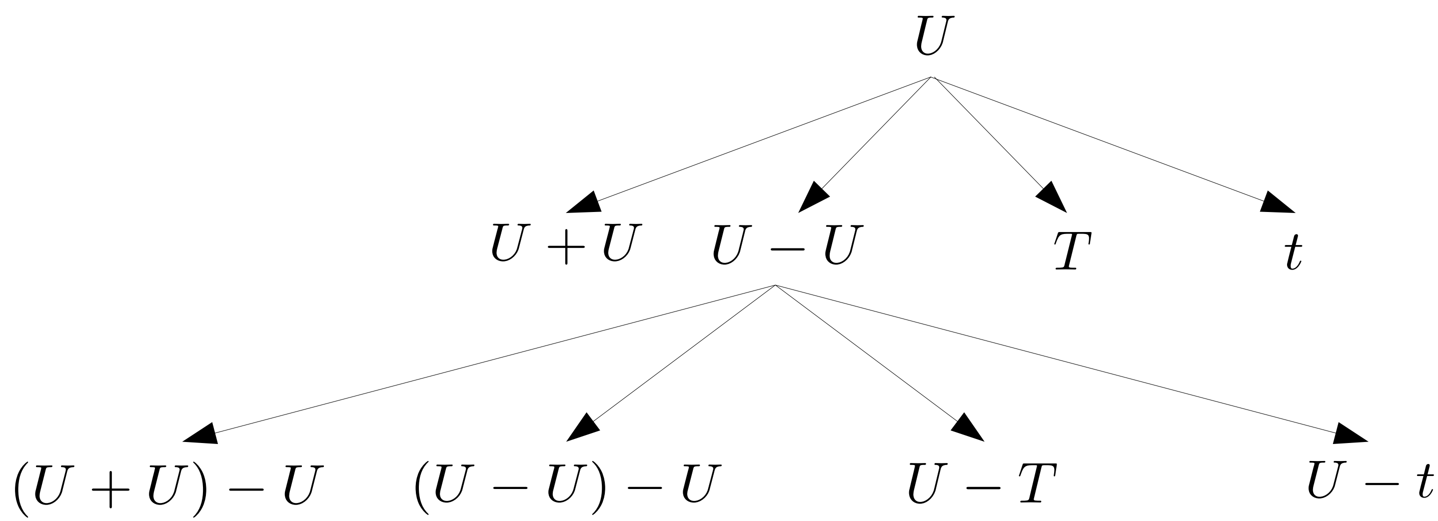
$\Delta\Bigg($ *Observed Samples:*

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **10**
(t = 10, T = 5) → **5**

; $\Bigg)$

Why ?

$t - T$



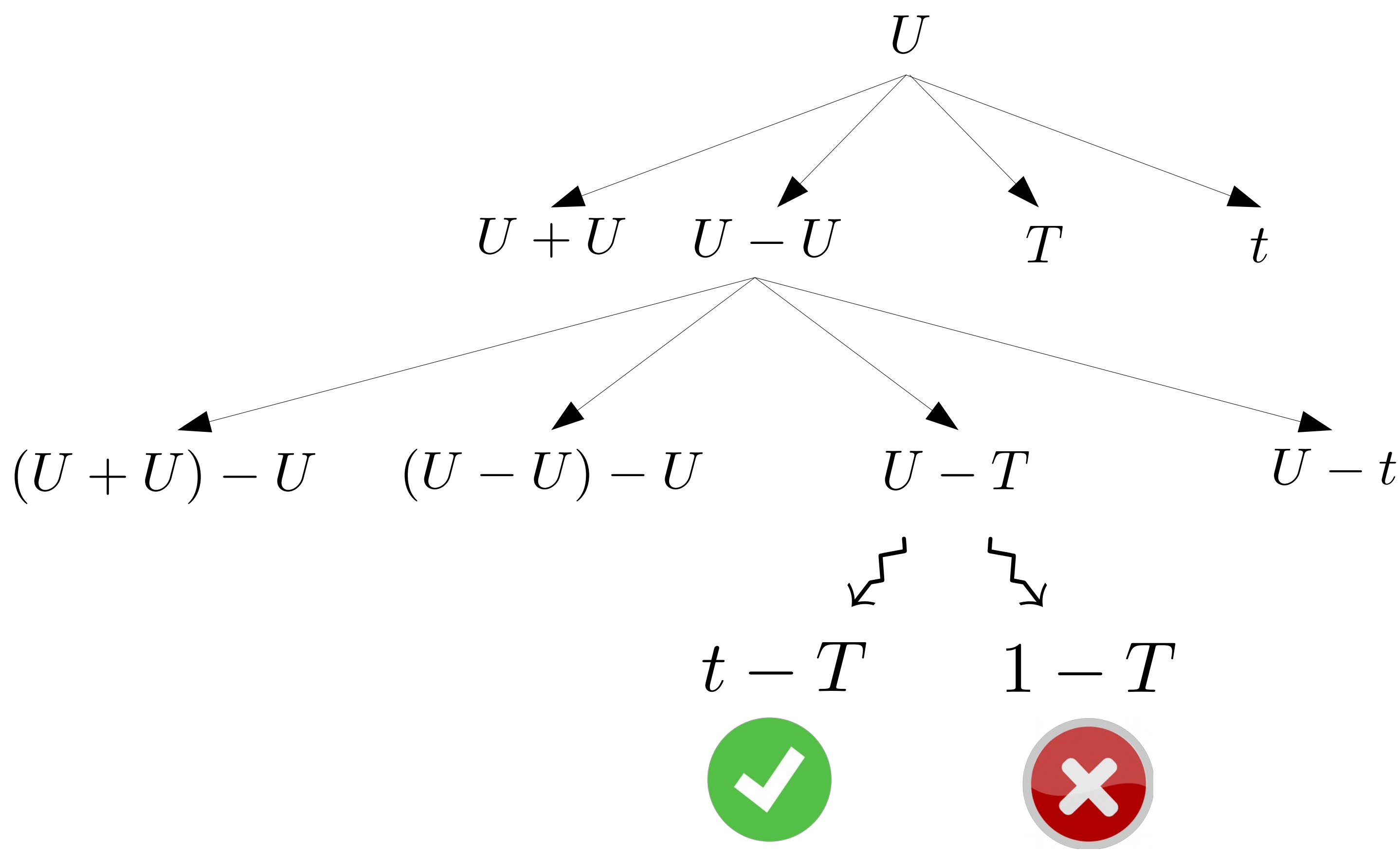
$\Delta\Big($ *Observed Samples:*

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **10**
(t = 10, T = 5) → **5**

$;\ \Big)$

Why ?

$t - T$



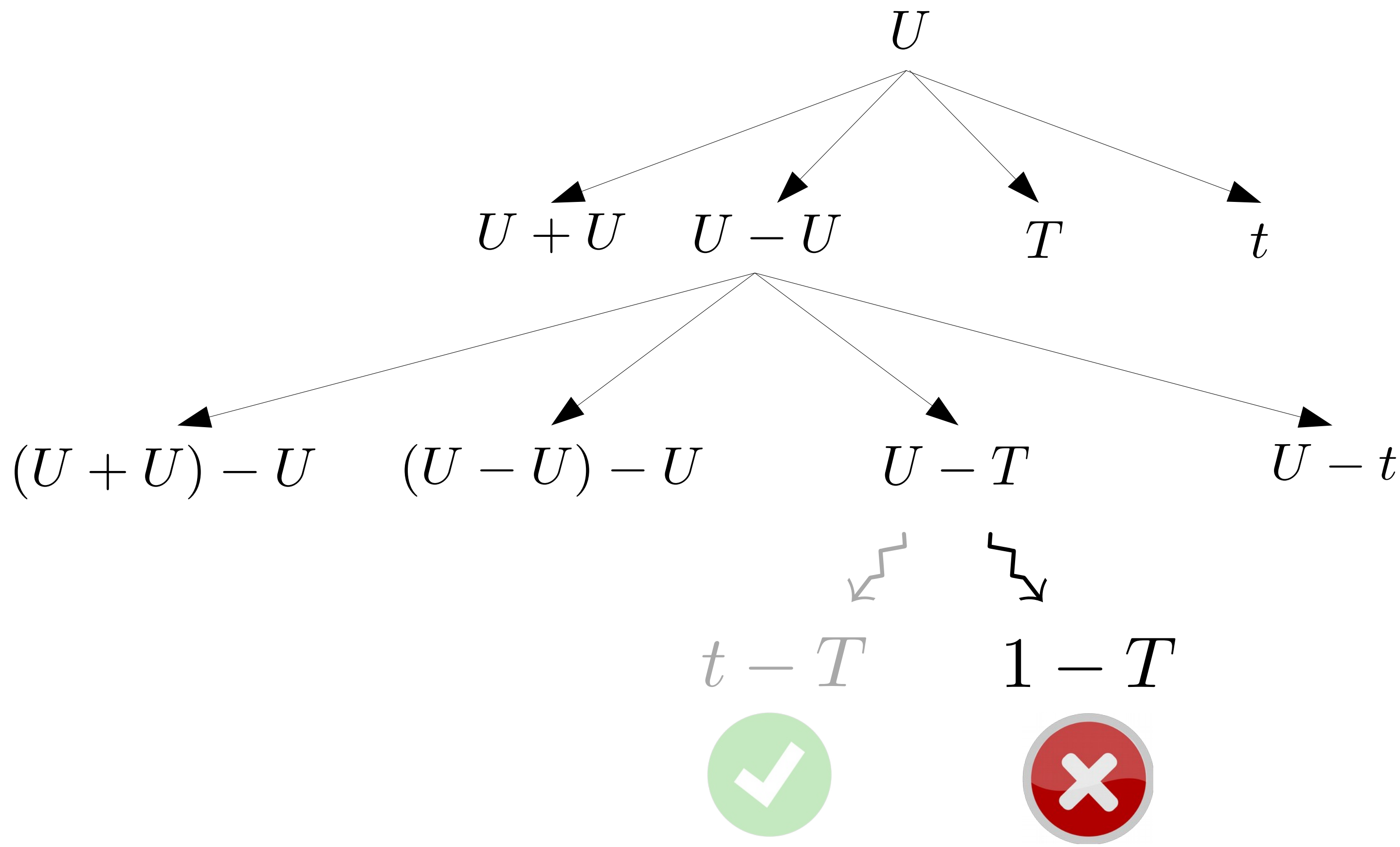
$\Delta\Big($ *Observed Samples:*

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **10**
(t = 10, T = 5) → **5**

; $\Big)$

Why ?

$t - T$



Δ (

Observed Samples:

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **10**
(t = 10, T = 5) → **5**

;

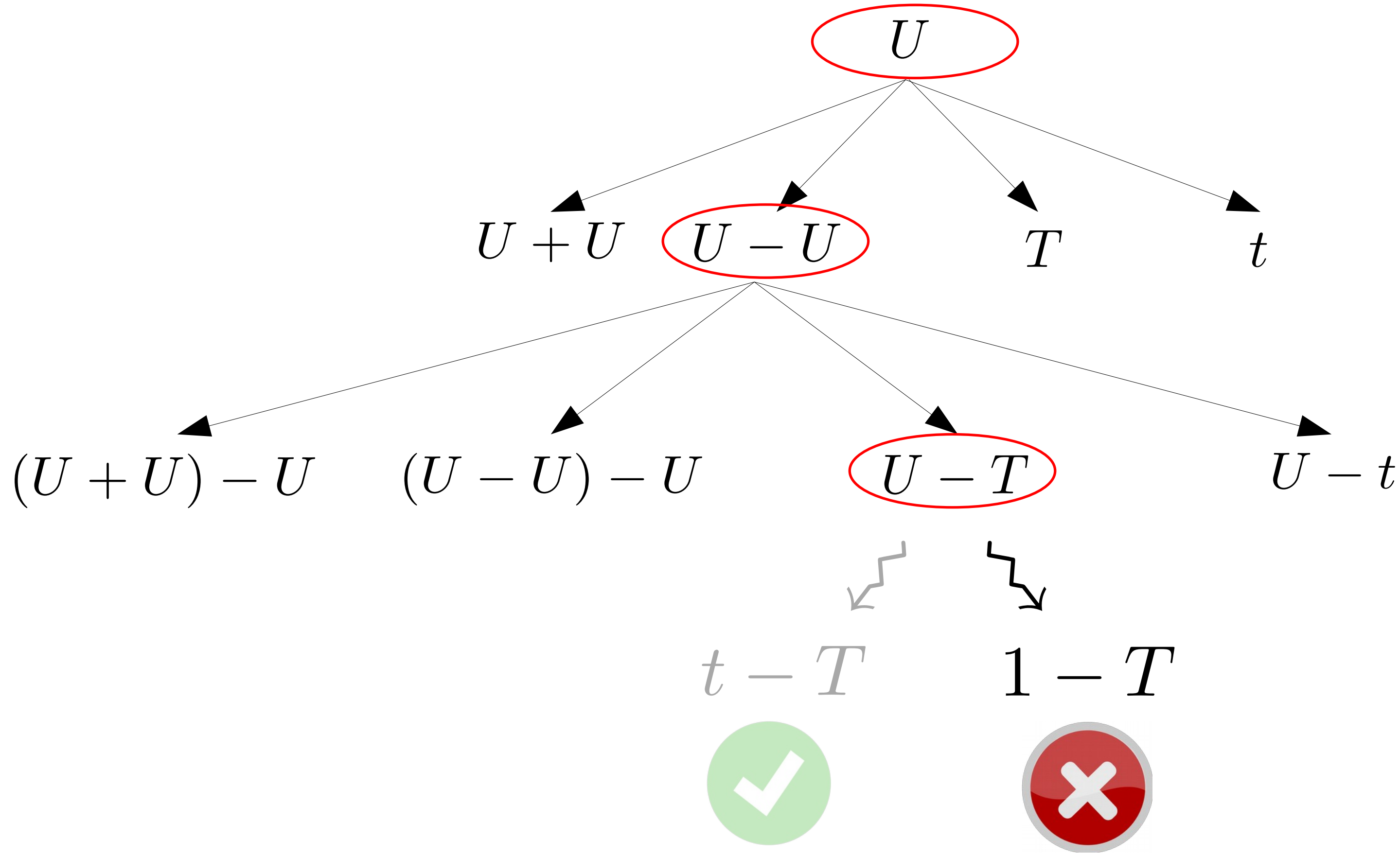
Synthesized Samples:

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **1**
(t = 10, T = 5) → **-4**

)

Why ?

$t - T$



$\Delta\Big($ *Observed Samples:*

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **10**
(t = 10, T = 5) → **5**

;

Synthesized Samples:

(t = 1, T = 2) → **-1**
(t = 10, T = 0) → **1**
(t = 10, T = 5) → **-4** $\Big)$

Claim

- Search space is **too unstable** for partial node evaluation
- Estimation of **non terminal expressions** is **misleading**

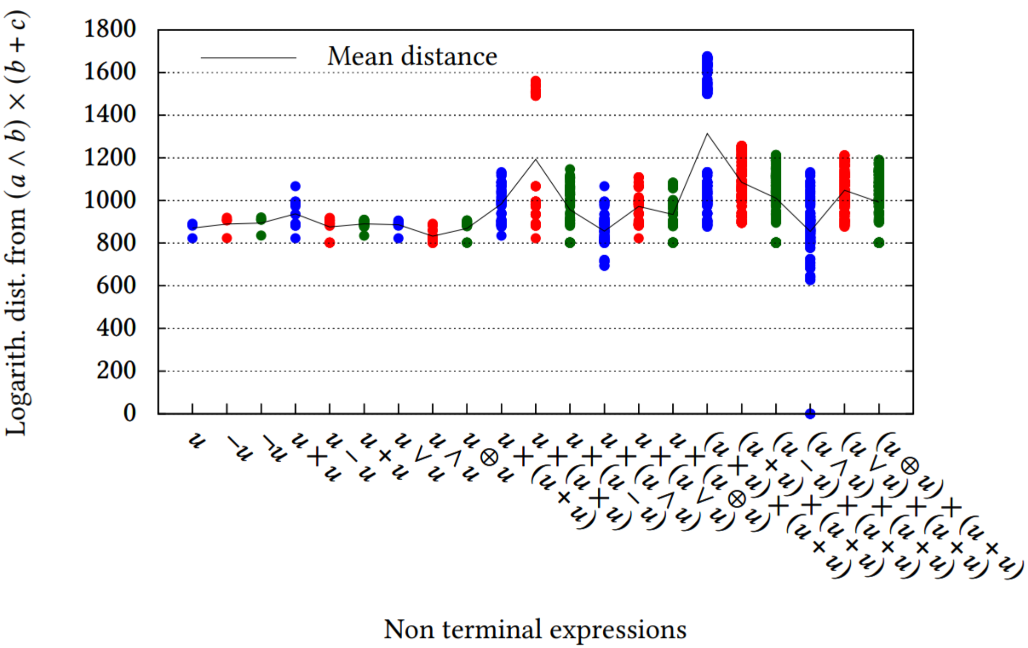
Evidence n°1 : 2 simulations can lead to very distinct distances

Evidence n°2 : Syntia does not benefit from partial evaluation

Evidence n°3 : Syntia behaves in practice almost as BFS

Evidence n°1 and 2

Evidence n°1



Simulation leads to completely different results

Evidence n°2

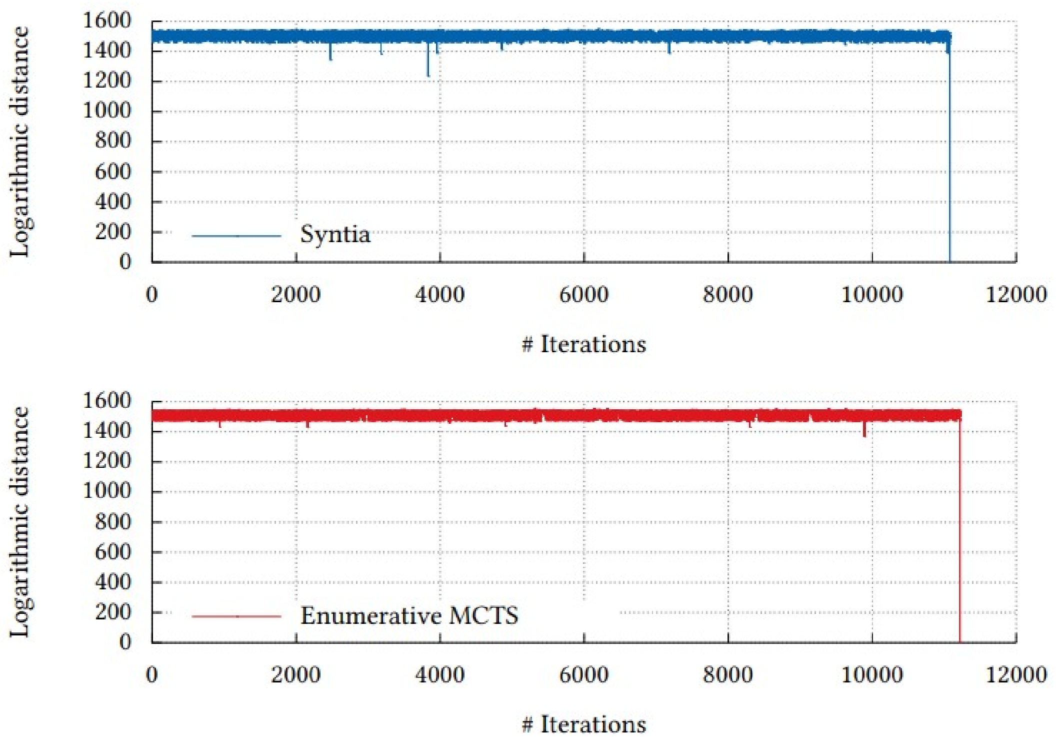
Number of expressions from exploitation steps ?

B1 (Syntia)
20 / 376

B2 (Ours)
34 / 341

Evidence n°3

- **Config. of Syntia makes MCTS almost BFS**



Syntia is not guided

+

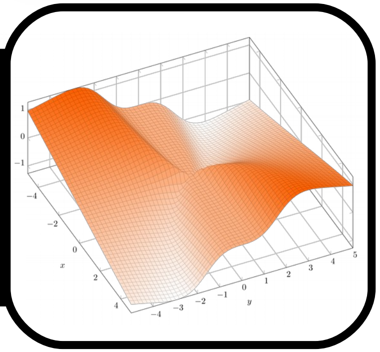
Over B2 Syntia and enum. MCTS reach similar results

Improve

Blackbox deobf., an optimization pb

Syntia sees blackbox deobfuscation as a
single player game

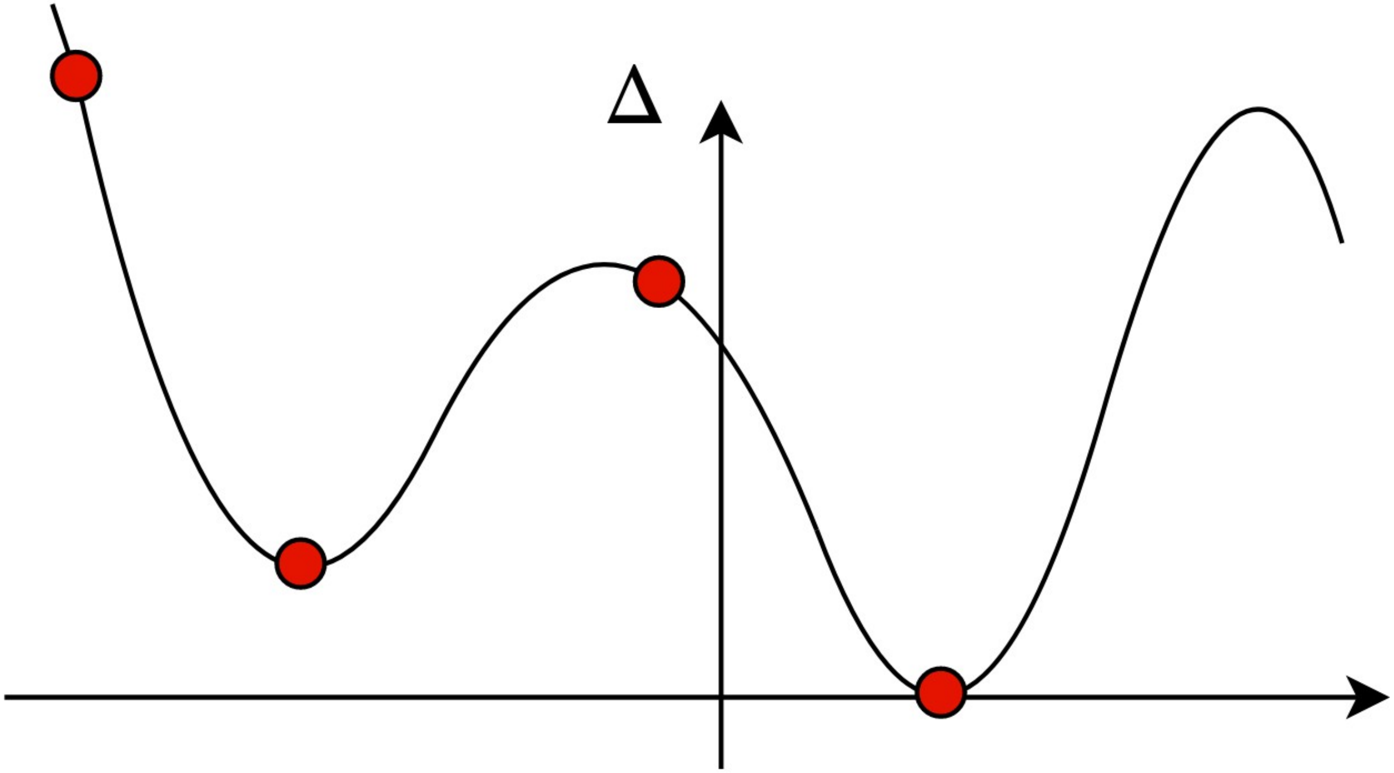
We propose to see it as an
optimization problem



Goal : find $\underbrace{s^*}_{\text{an expr.}}$ s.t. $\underbrace{f}_{\Delta}(s^*) \leq f(s), \forall s \in S$

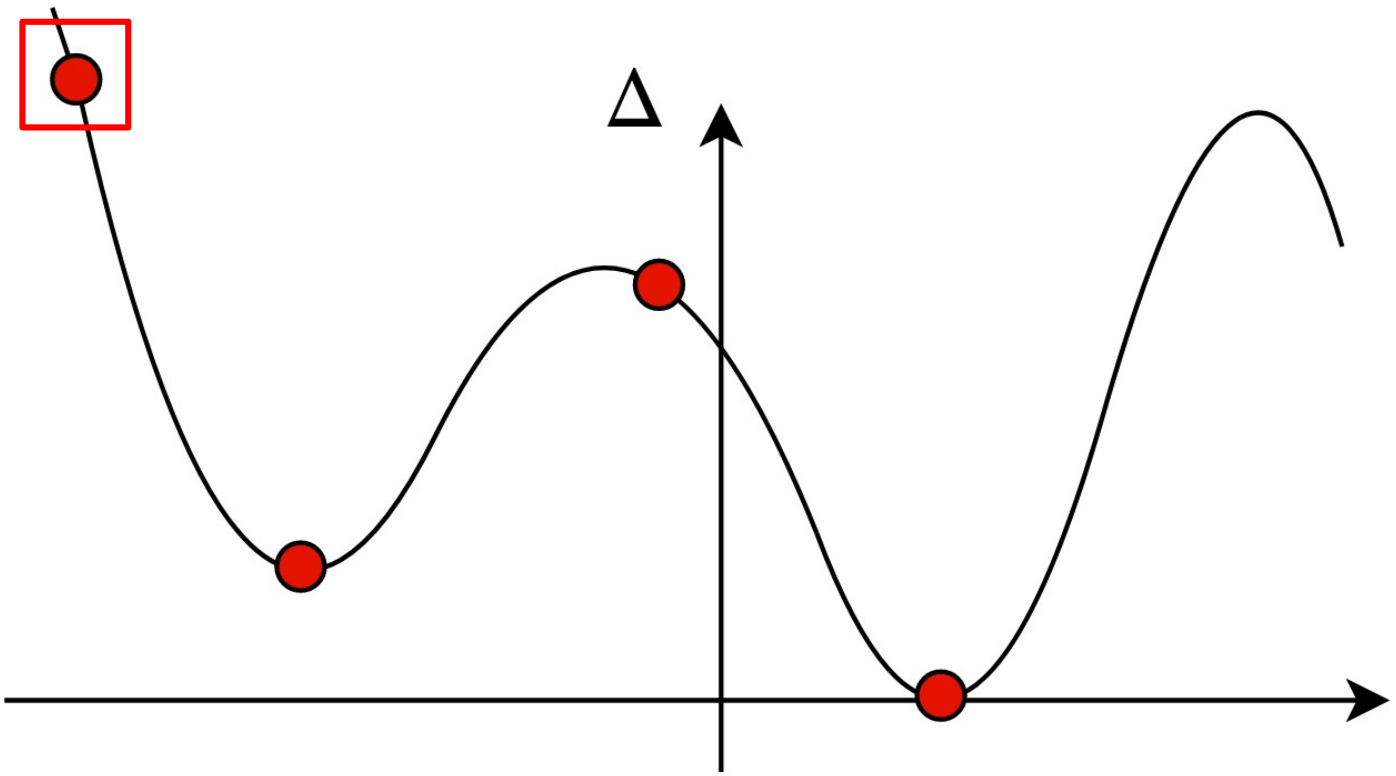
S-metaheuristics

- **Solve optimization problems**



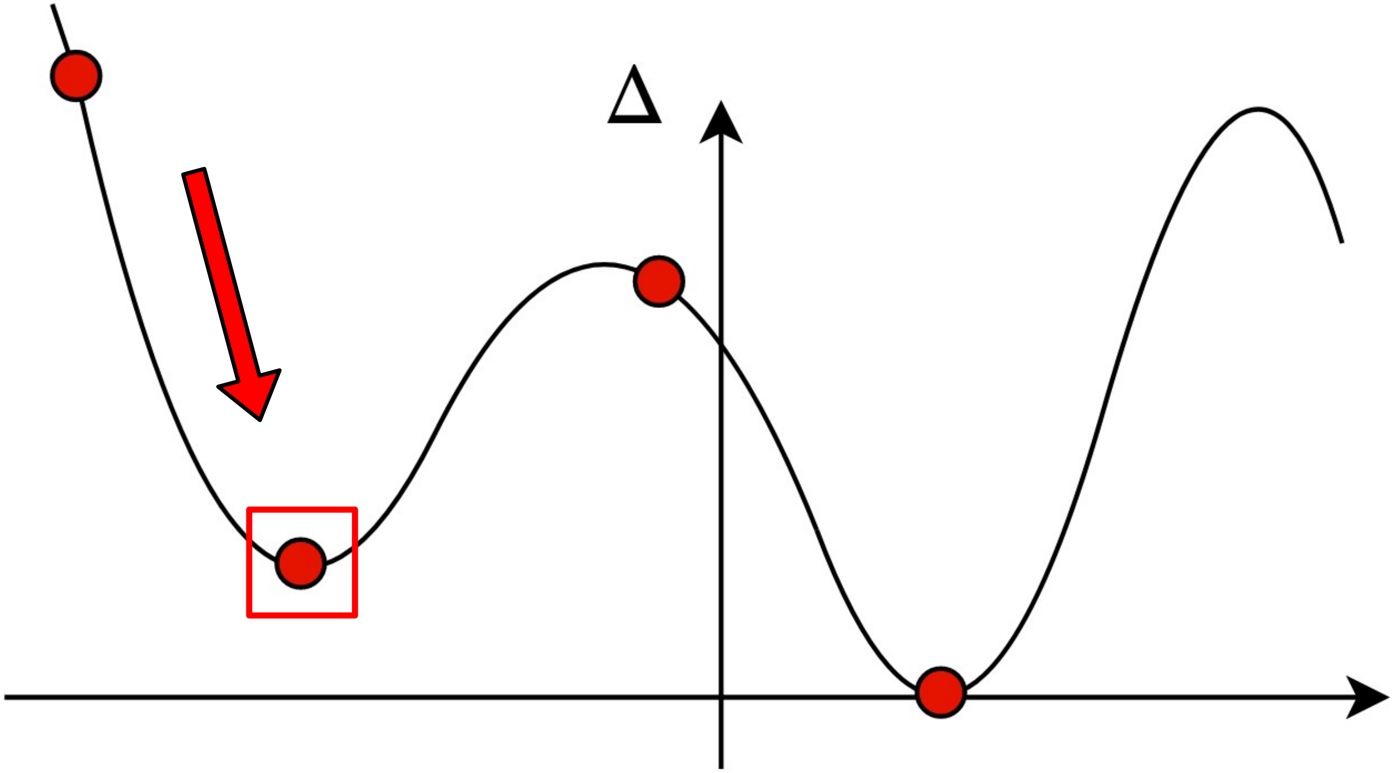
S-metaheuristics

- **Solve optimization problems**



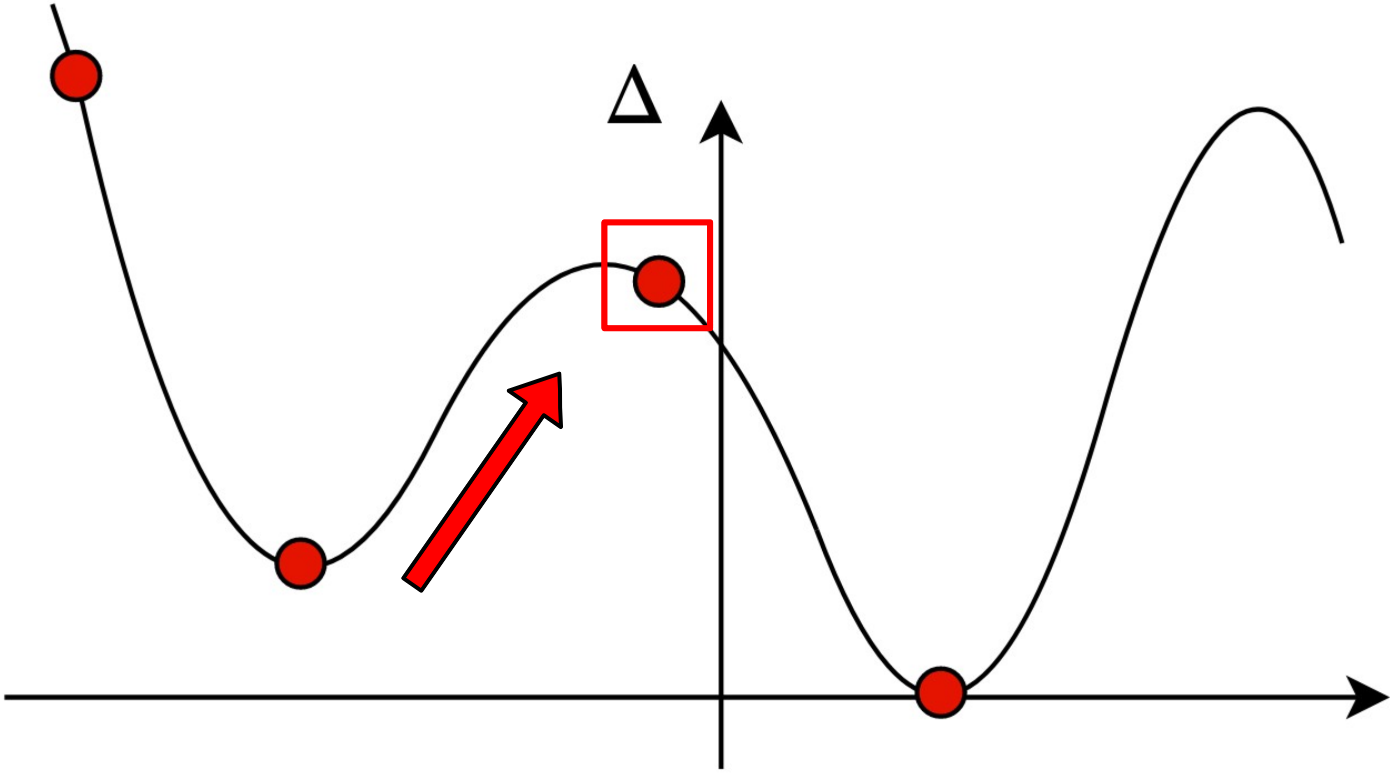
S-metaheuristics

- **Solve optimization problems**



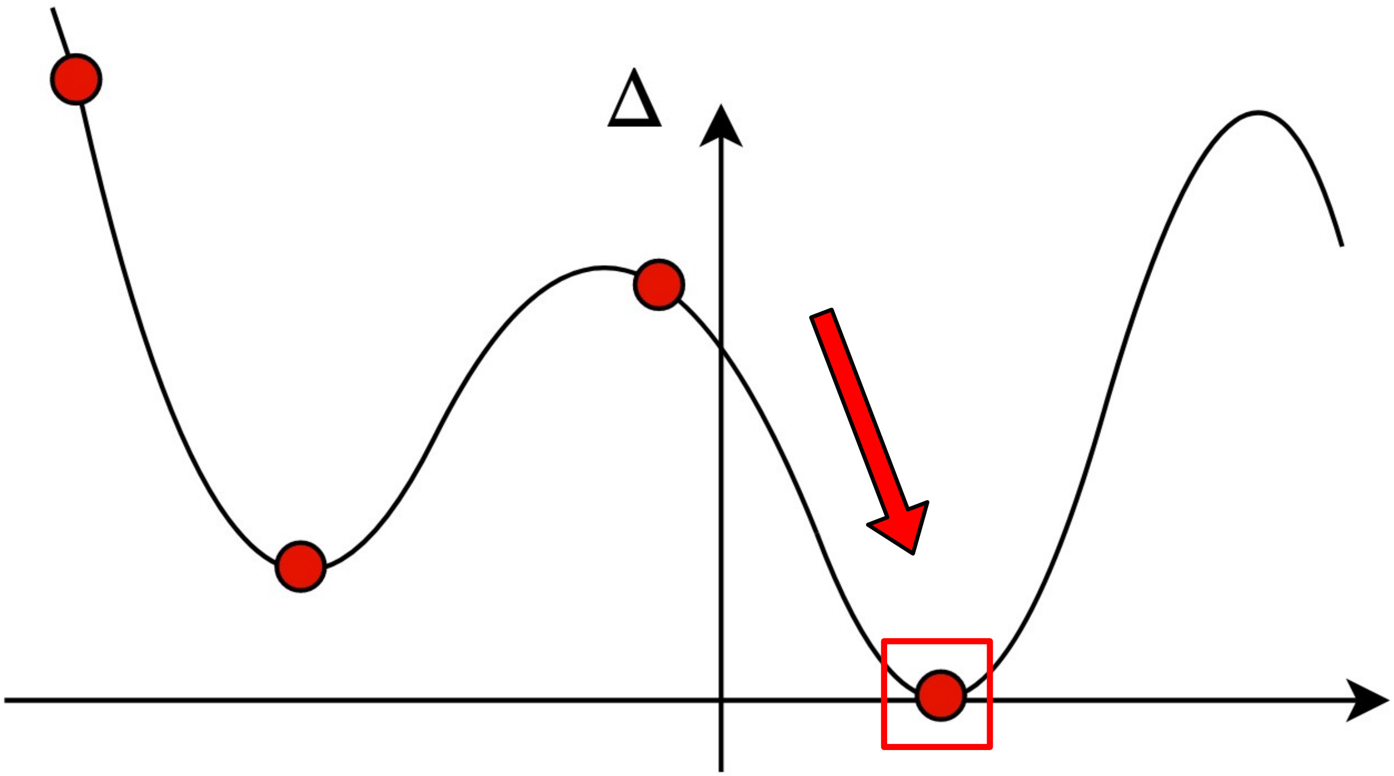
S-metaheuristics

- **Solve optimization problems**



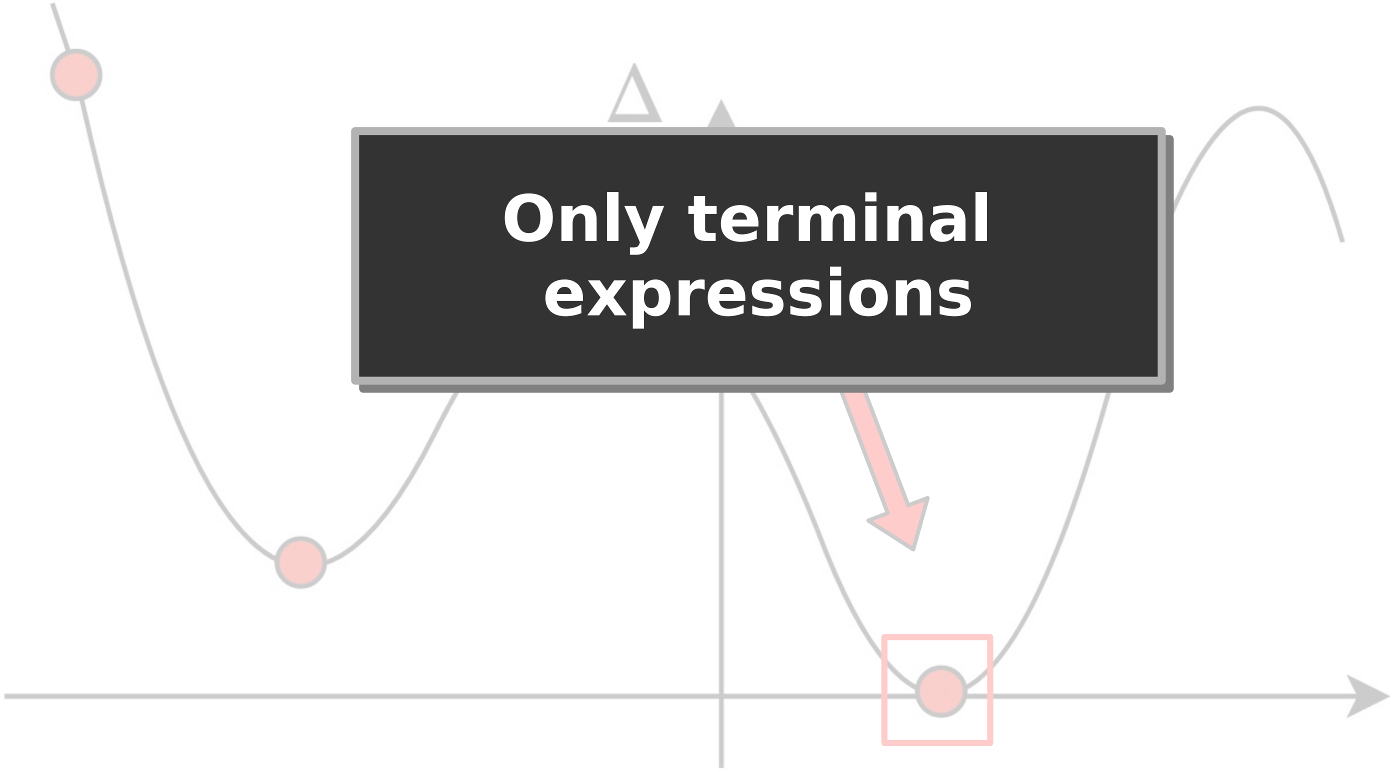
S-metaheuristics

- **Solve optimization problems**



S-metaheuristics

- Solve optimization problems



New prototype: Xyntia

Xyntia

S-metaheuristics

Can choose between:

→ Hill Climbing
→ Simulated annealing
→ Metropolis Hasting
→ **Iterated Local Search**

\> **MCTS**

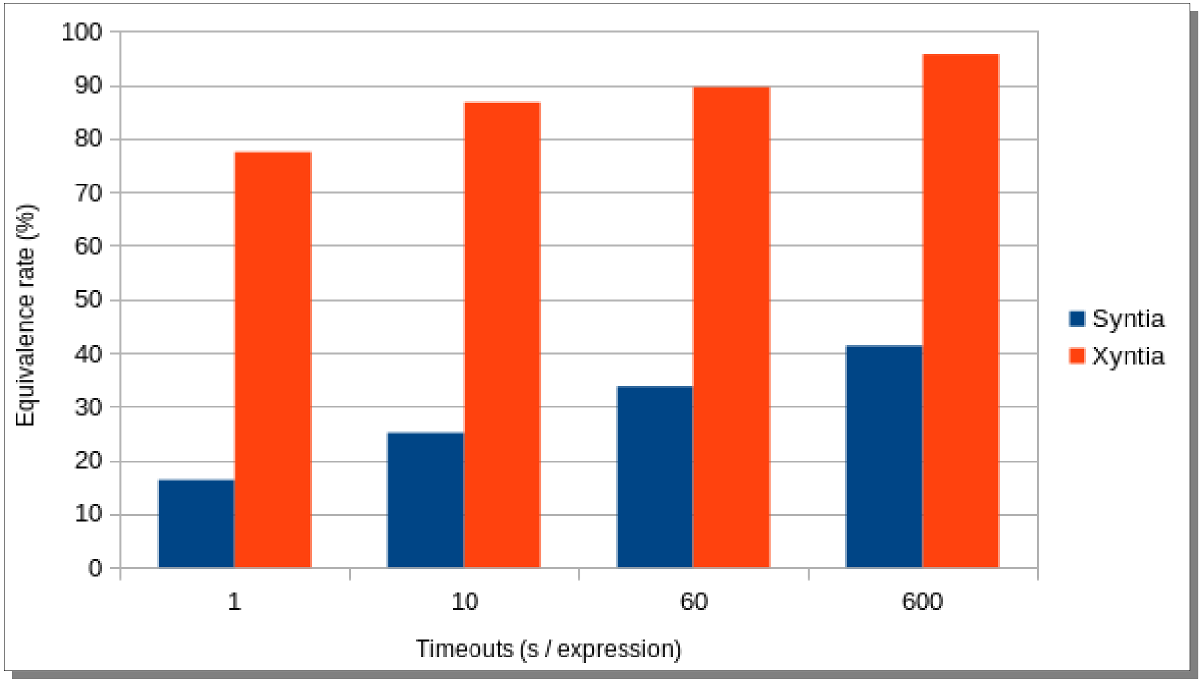
Xyntia vs Syntia

B1 (Syntia)

- **100 %** success rate in **1 s/expr.**

Syntia: 75% in 60 s/expr.

B2 (Ours)

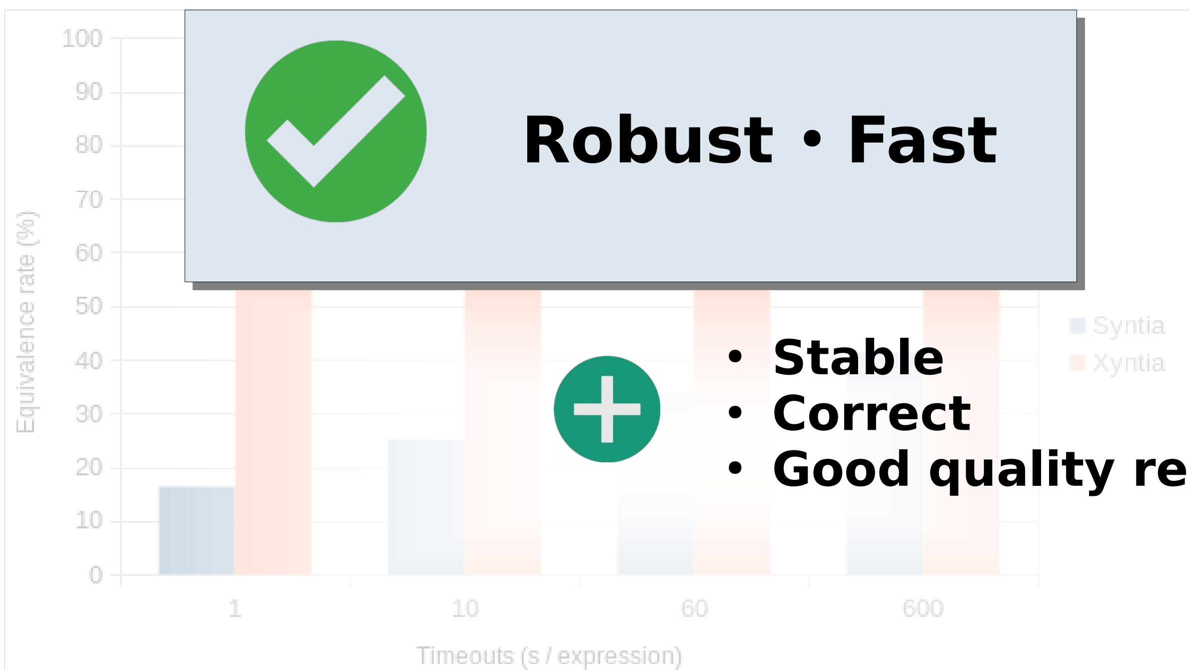


Xyntia vs Syntia

B1 (Syntia)

- **100 %** success rate in **1 s/expr.**

B2 (Ours)



Robust • Fast

- **Stable**
- **Correct**
- **Good quality results**

Other experiments

- Xyntia against QSynth
- Xyntia against “compiler like simplifications”
- Xyntia against program synthesizer **CVC4**
- Xyntia against superoptimizer **STOKE**

404

Not Found

The resource requested could not be found on this server!

- Use-cases:
  - State-of-the-art protections
  - **VM-based obfuscation**

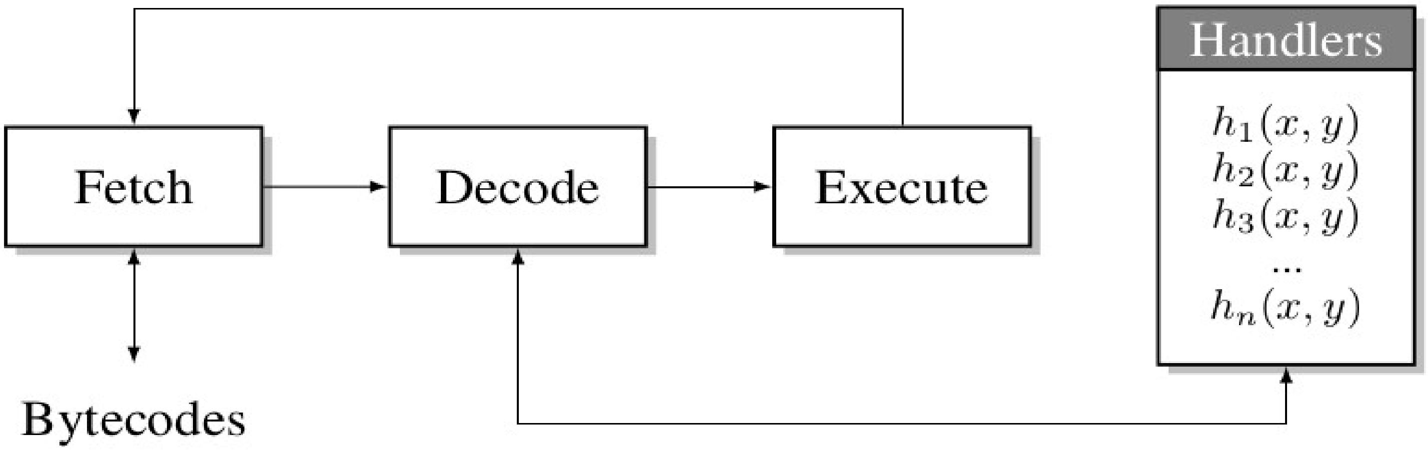
What’s next?


PROTECTIONS ARE BROKEN!


DON'T WORRY...
I HAVE A PLAN
memegenerator.net

Mitigate

Context : Virtualization



Proved to be sensitive to blackbox deobfuscation

Why VM-based obf. is vulnerable ?

- Handlers are too semantically simple:
  → e.g. $+$, $-$, $\times$, $\wedge$, $\vee$

- Obfuscation increase syntactic complexity
  → **Blackbox deobf. is not impacted**

We need to move ...

From syntactic to semantic complexity

Semantically complex expressions

- **Goal:**
  - Increase the semantic complexity of each handlers
  - Keep a Turing complete set of handlers

- **Example:**

$$
\begin{array}{rrcl}
 & h_0 & = & (x+y) + -((a - x^2) - (xy)) \\
+ & h_1 & = & (a - x^2) - xy + (-(y - (a \wedge x)) \times (y \otimes x)) \\
+ & h_2 & = & (y - (a \wedge x)) \times (y \otimes x) \\
\hline
 & h & = & x + y
\end{array}
$$

Merged handlers

- **Goal:**
  - Increase semantic + sampling complexity

- **Example:**

$$h_1(x,y) = x + y \quad \text{and} \quad h_2(x,y) = x \wedge y$$

$$\rightarrow \quad h(x,y,c) = \text{if } (c = cst) \text{ then } h_1(x,y) \text{ else } h_2(x,y)$$

- **Need to hide conditionals:**

```
int32_t h(int32_t a, int32_t b, int32_t c) {
    // if (c == cst) then h1(a,b,c) else h2(a,b,c);
    int32_t res = c - cst ;
    int32_t s = res >> 31;
    res = (-((res ^ s) -s) >> 31) & 1;
    return h1(a, b, c)*(1 - res) + res*h2(a, b, c);
}
```

Semantically complex handlers: results

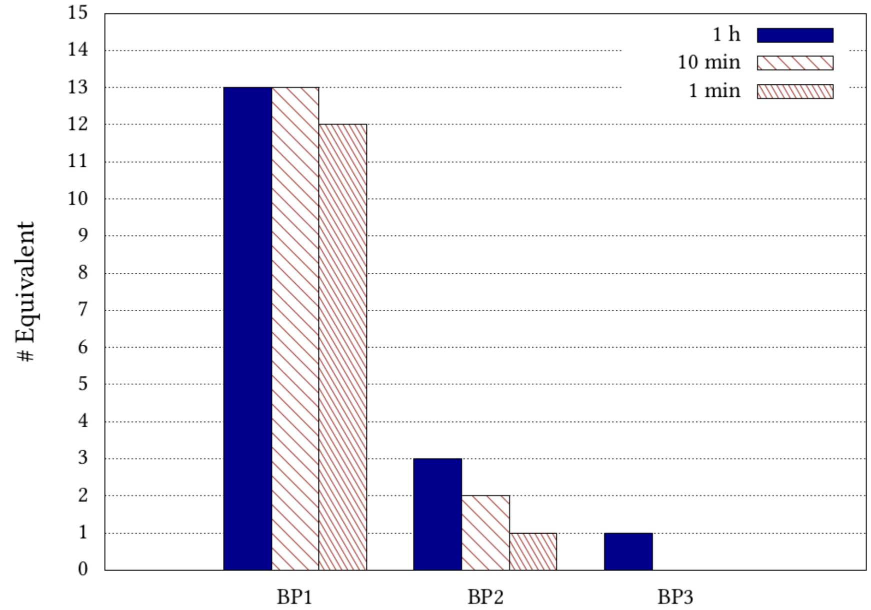


Figure 8: Xyntia on BP1,2, 3

More results:

- Syntia with 12h/exprs. → 1/15 on BP1

Merged handlers: results

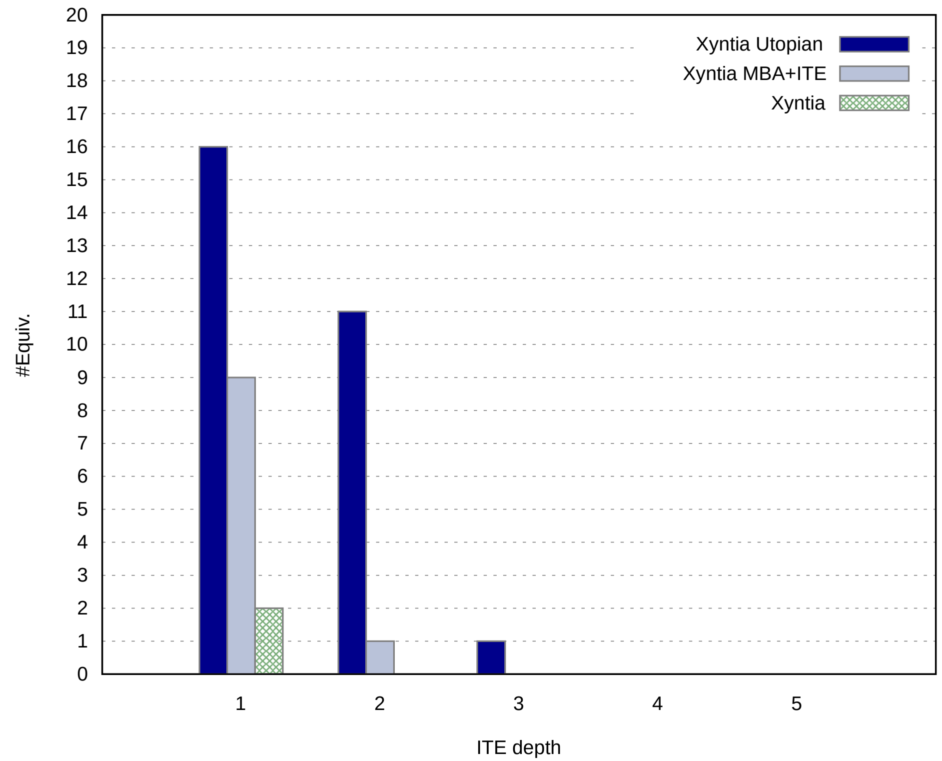


Figure 10: Merged handlers: Xyntia (timeout=60s)

More results:

- Syntia finds nothing for ≥ 2 nested ITE

Conclusion

MCTS is not appropriate for blackbox deobfuscation
→ Search space too unstable
→ Estimation of non terminal expressions pertinence is misleading

S-metaheuristics yields a significant improvement
→ More robust
→ Much Faster

Moving for syntactic to semantic complexity
→ 2 efficient methods to protect against blackbox deobfuscation

Thank you for your attention