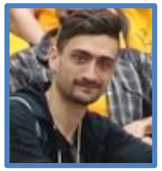
Journey Beyond Full Abstraction:

Exploring Robust Property Preservation for Secure Compilation


Carmine Abate
Inria Paris

Rob Blanco
Inria Paris

Deepak Garg
MPI-SWS

Cătălin Hrițcu
Inria Paris

Jérémy Thibault
Inria Paris

Marco Patrignani
Stanford & CISPA

Good programming languages provide helpful abstractions for writing more secure code

Good programming languages provide helpful abstractions for writing more secure code

- structured control flow, procedures, modules, interfaces, correctness and security specifications, ...

Good programming languages provide helpful abstractions for writing more secure code

- structured control flow, procedures, modules, interfaces, correctness and security specifications, ...

abstractions not enforced when compiling and linking with adversarial low-level code

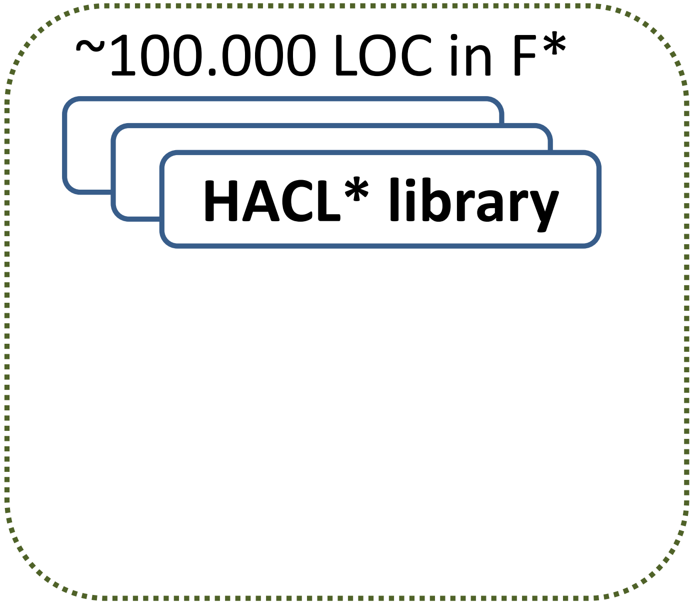
Good programming languages provide helpful abstractions for writing more secure code

- structured control flow, procedures, modules, interfaces, correctness and security specifications, ...

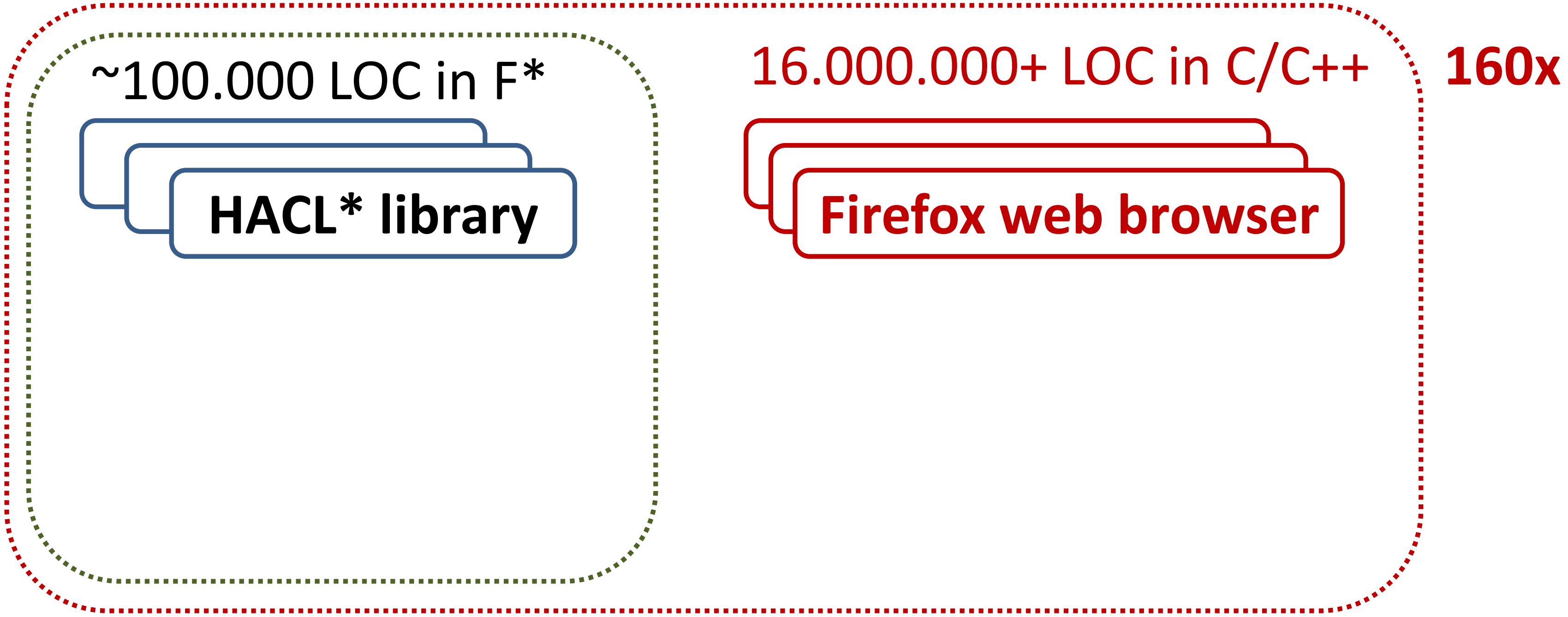
abstractions not enforced when compiling and linking with adversarial low-level code

- all source-level security guarantees are lost

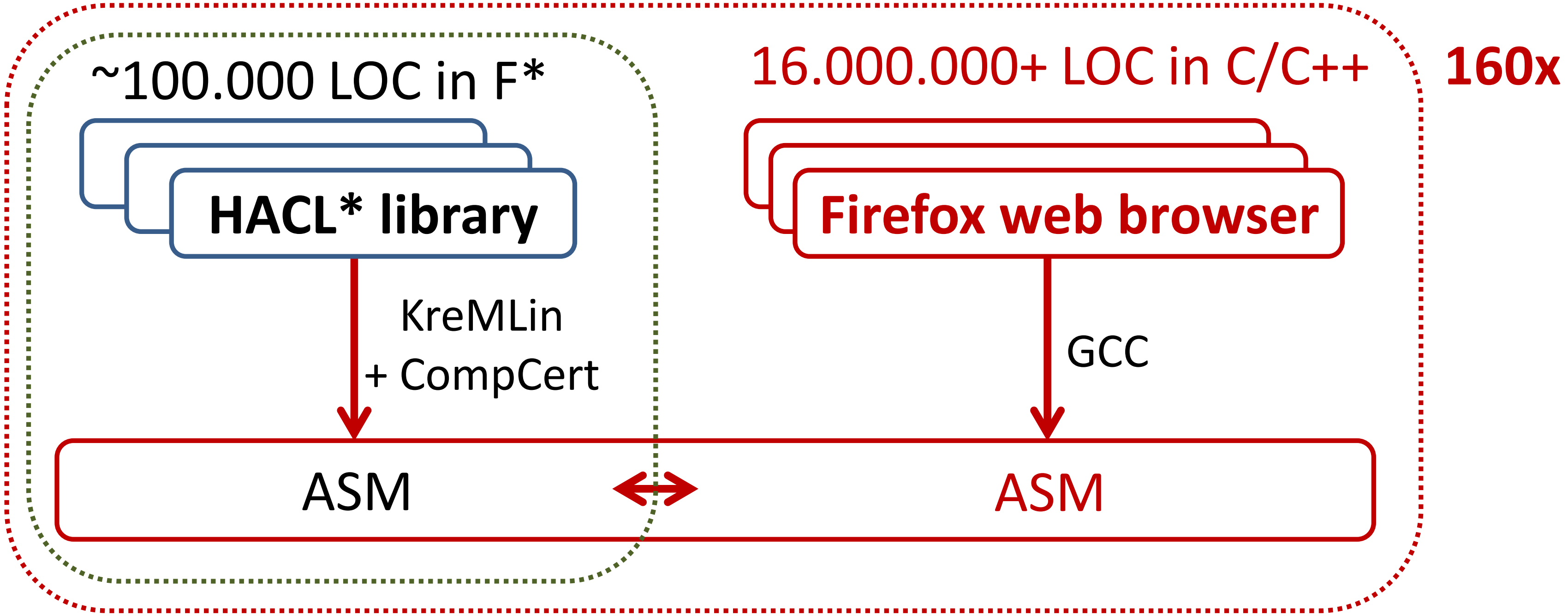
HACL* verified cryptographic library



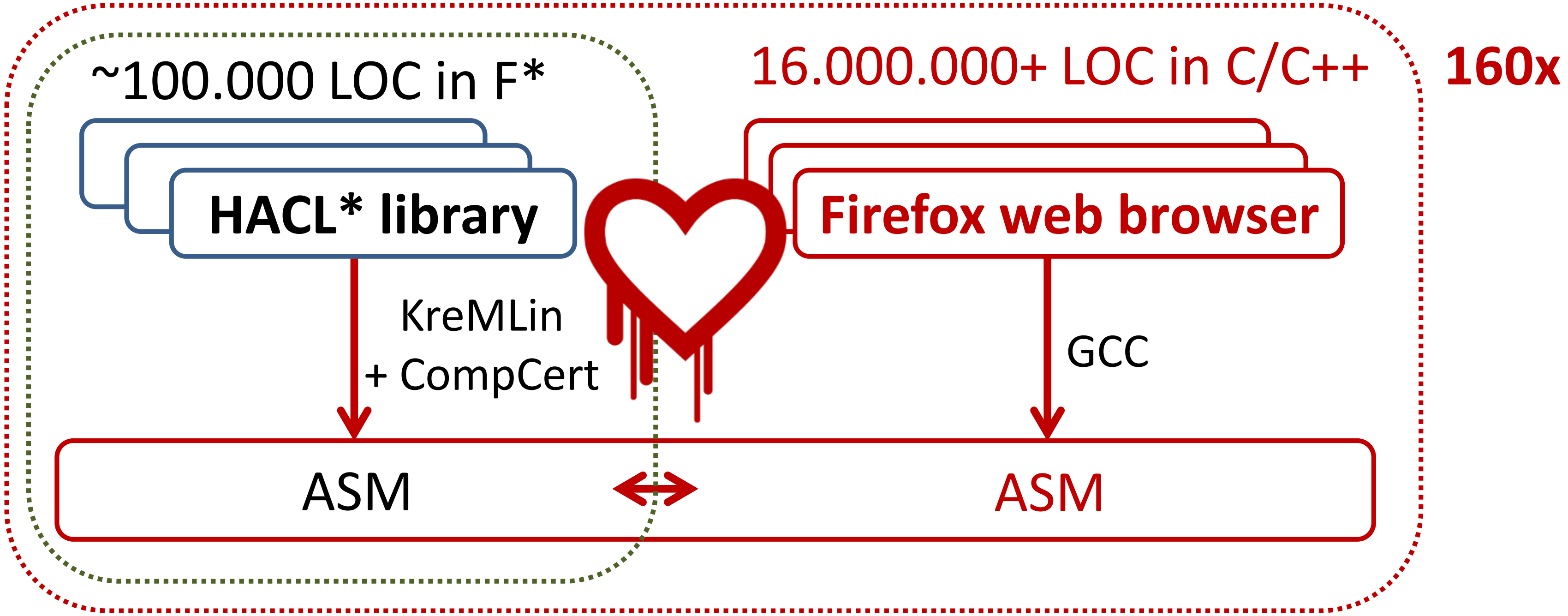
HACL* verified cryptographic library, in practice



HACL* verified cryptographic library, in practice



HACL* verified cryptographic library, in practice



Insecure interoperability: **linked code can read and write data and code, jump to arbitrary instructions, smash the stack, ...**

We need secure compilation chains

- Protect source-level abstractions
  even against linked adversarial low-level code

We need secure compilation chains

- Protect source-level abstractions
  even against linked adversarial low-level code
  - **various enforcement mechanisms**: processes, SFI, ...
  - shared responsibility: compiler, linker, loader, OS, HW

We need secure compilation chains

- Protect source-level abstractions
  even against linked adversarial low-level code
  - **various enforcement mechanisms**: processes, SFI, ...
  - shared responsibility: compiler, linker, loader, OS, HW
- Goal: enable source-level security reasoning

We need secure compilation chains

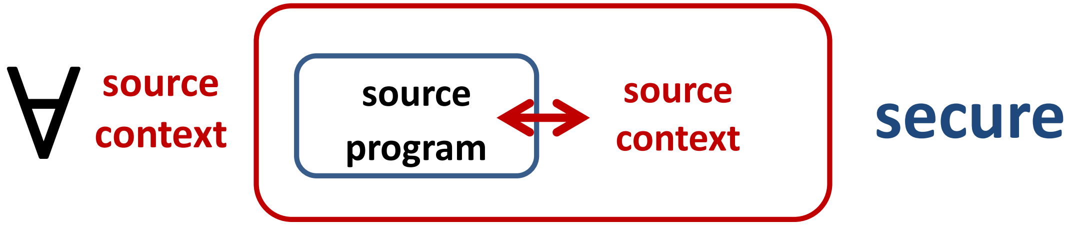
- Protect source-level abstractions
  even against linked adversarial low-level code
  - **various enforcement mechanisms**: processes, SFI, ...
  - shared responsibility: compiler, linker, loader, OS, HW
- Goal: enable source-level security reasoning
  - **linked adversarial target code cannot break the security of compiled program** any more than some linked source code

We need secure compilation chains

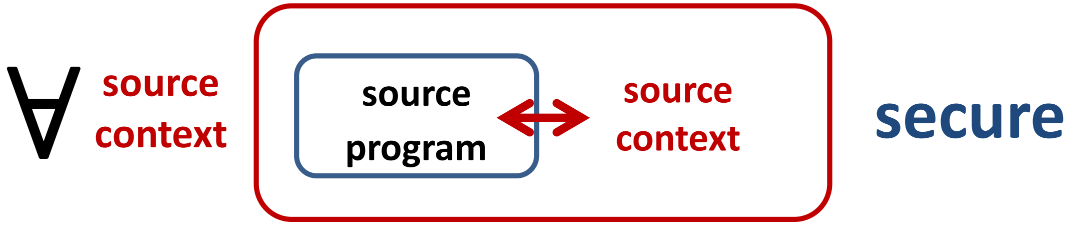
- **Protect source-level abstractions**
  even against linked adversarial low-level code
  - **various enforcement mechanisms**: processes, SFI, ...
  - shared responsibility: compiler, linker, loader, OS, HW
- **Goal: enable source-level security reasoning**
  - **linked adversarial target code cannot break the security of compiled program** any more than some linked source code
  - **no "low-level" attacks**

Robustly preserving security

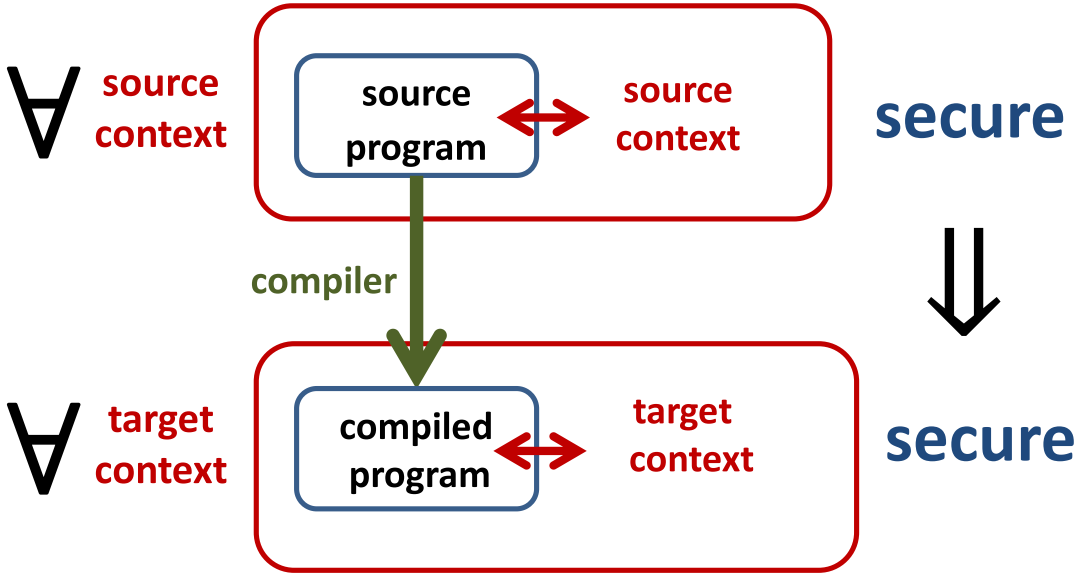
Robustly preserving security



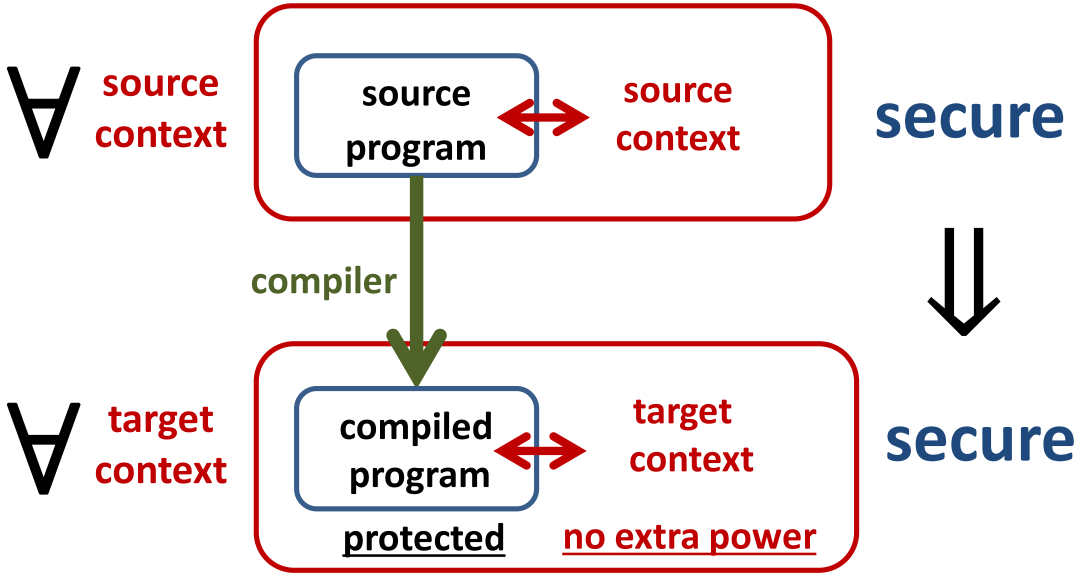
Robustly preserving security



Robustly preserving security



Robustly preserving security



But what should "secure" mean?

What properties should we robustly preserve?

What properties should we robustly preserve?

trace properties
(safety & liveness)

What properties should we robustly preserve?

hyperproperties
(noninterference)

trace properties
(safety & liveness)

What properties should we robustly preserve?

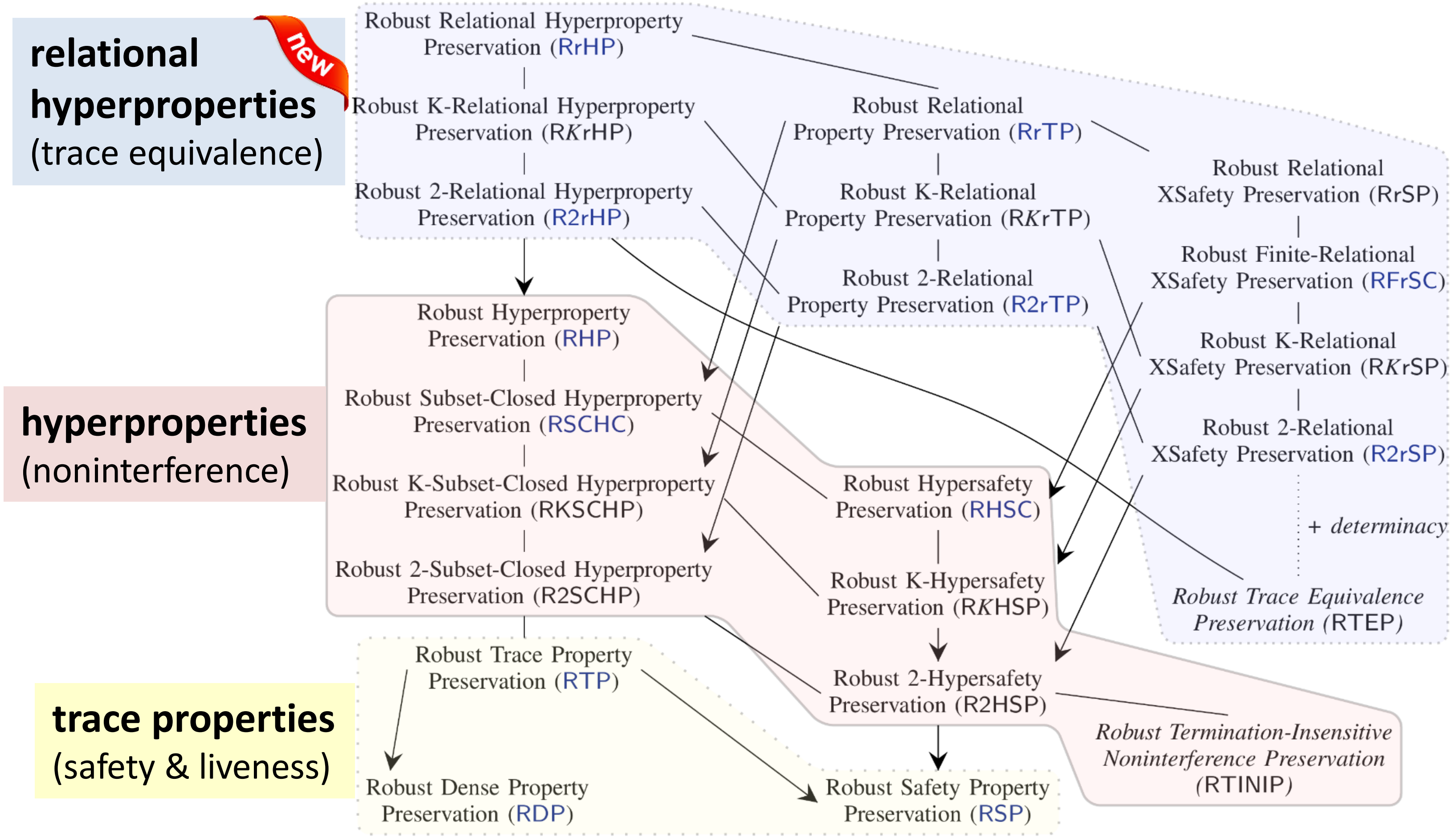
relational hyperproperties
(trace equivalence)

new

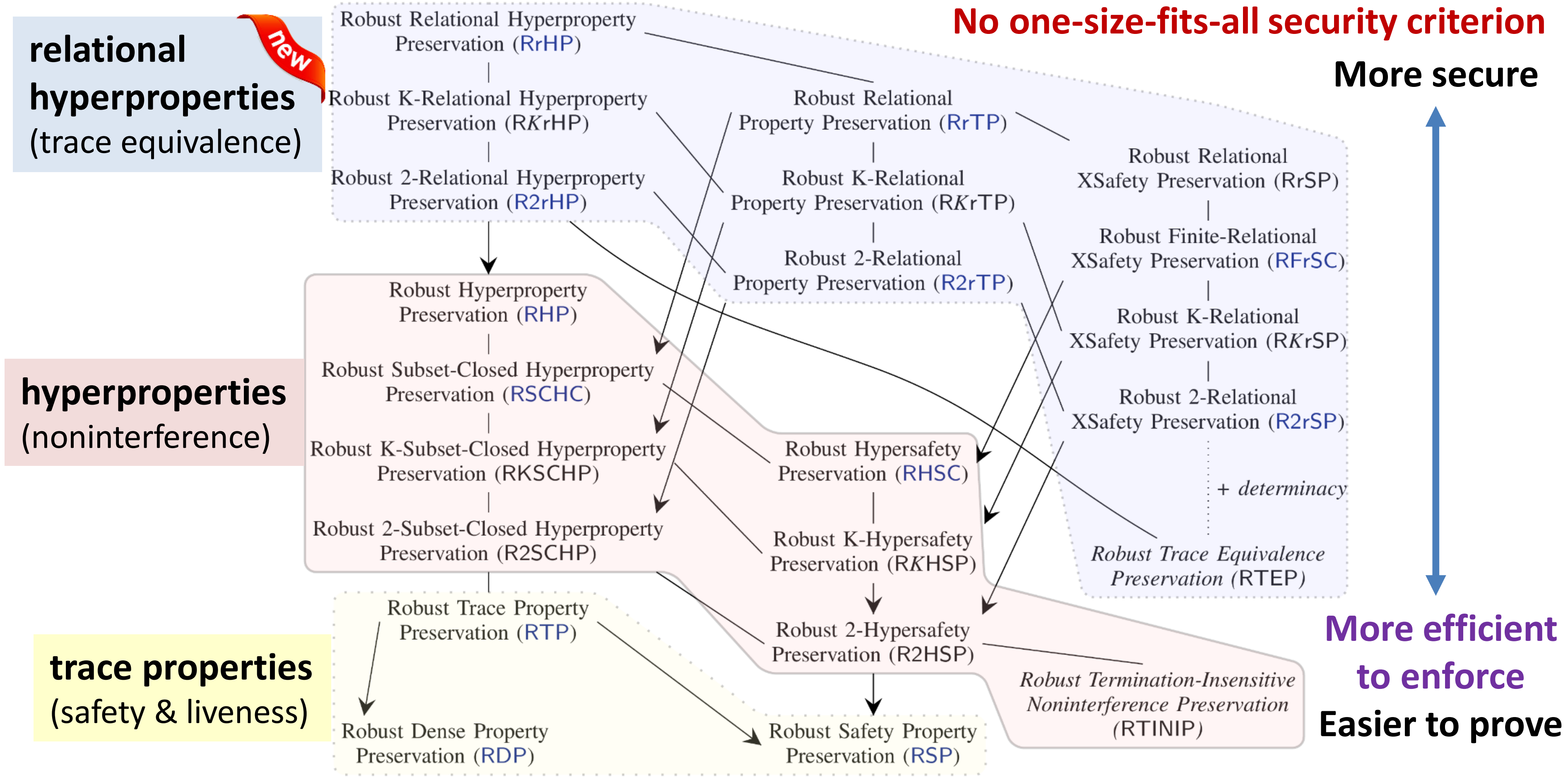
hyperproperties
(noninterference)

trace properties
(safety & liveness)

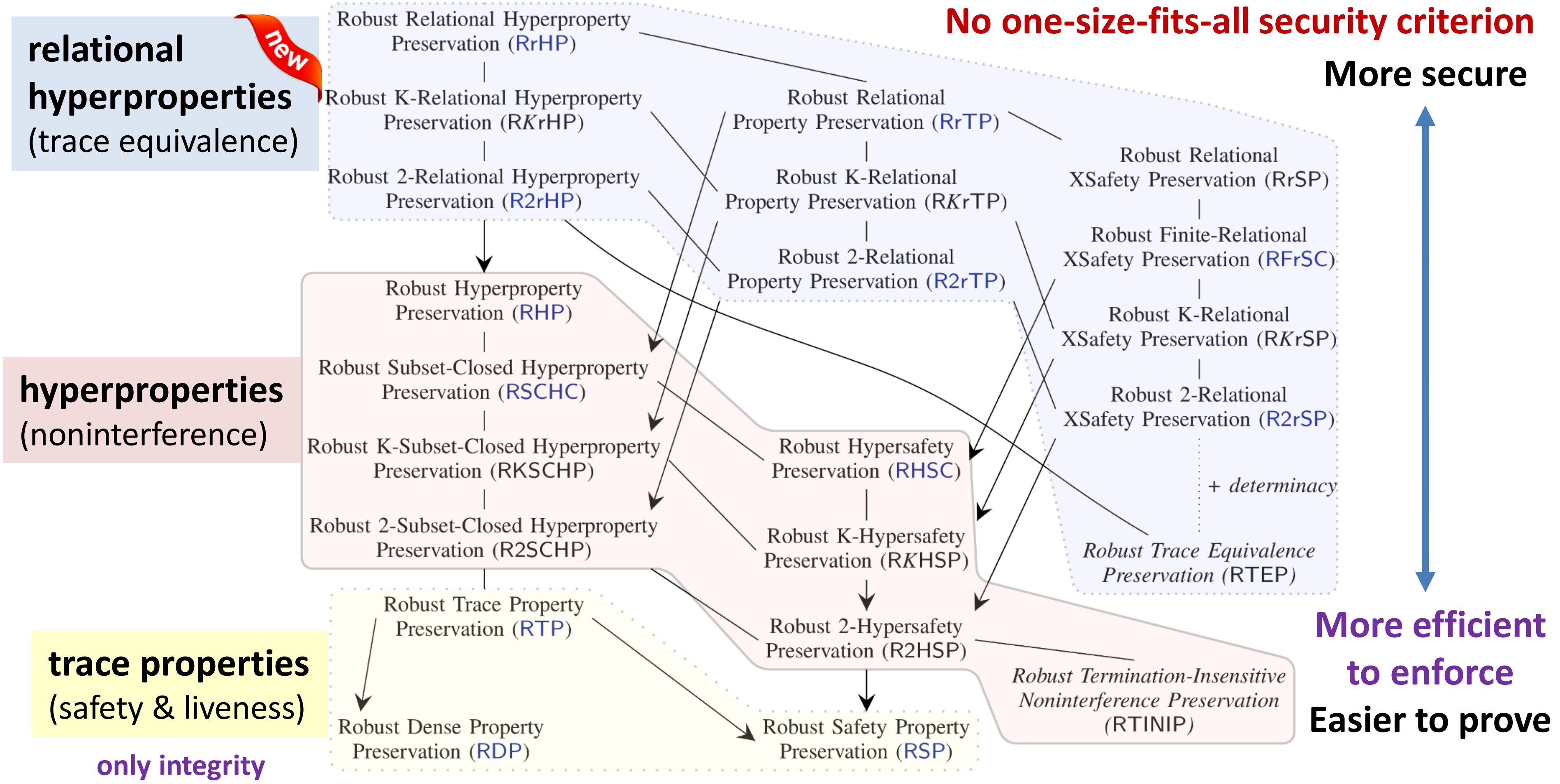
What properties should we robustly preserve?


relational hyperproperties (trace equivalence)
new
hyperproperties (noninterference)
trace properties (safety & liveness)
Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

What properties should we robustly preserve?

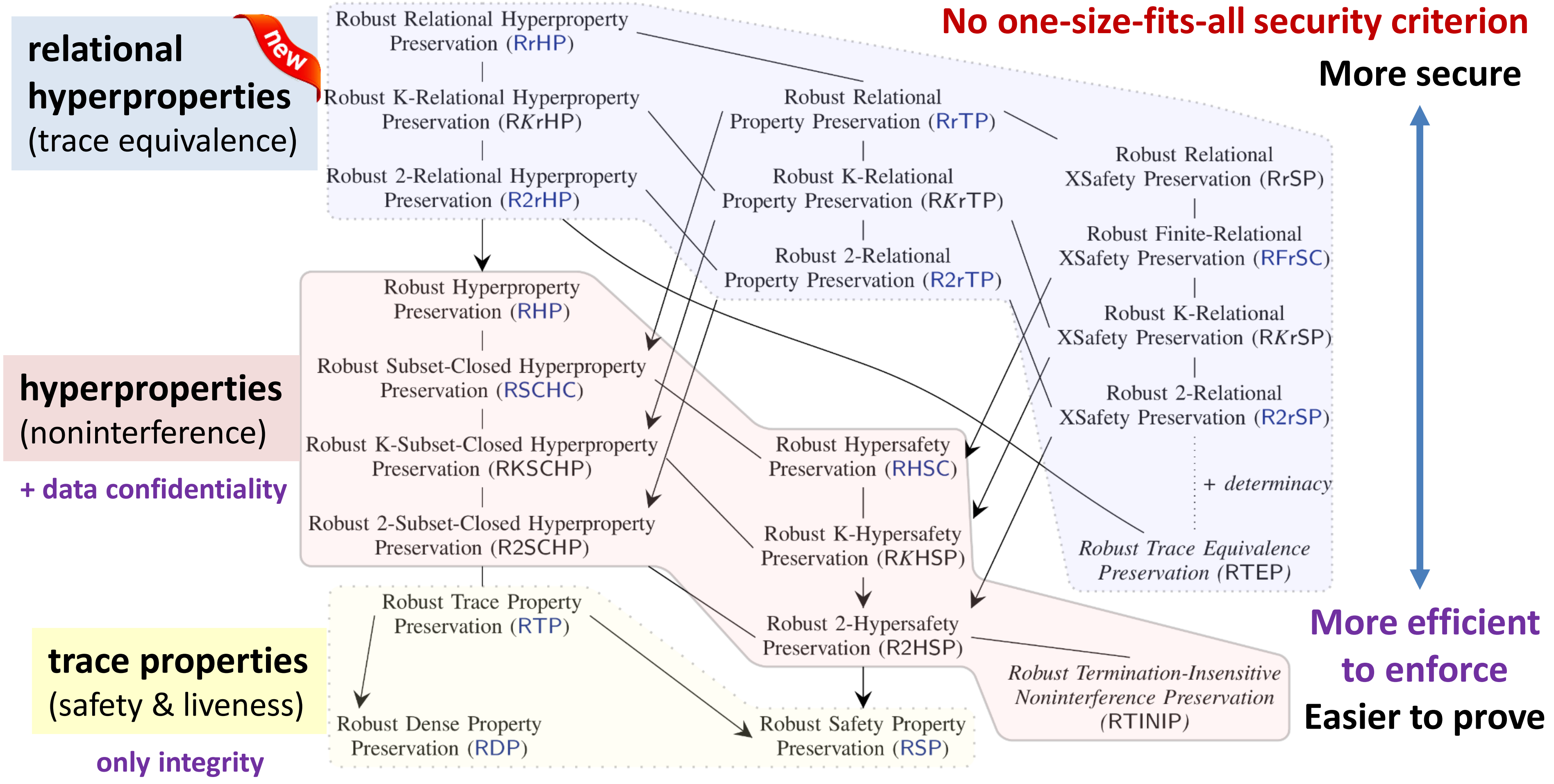

relational hyperproperties
(trace equivalence)
new
hyperproperties
(noninterference)
trace properties
(safety & liveness)
No one-size-fits-all security criterion
More secure
More efficient to enforce
Easier to prove
Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

What properties should we robustly preserve?



relational hyperproperties
(trace equivalence)
new
hyperproperties
(noninterference)
trace properties
(safety & liveness)
only integrity
No one-size-fits-all security criterion
More secure
More efficient to enforce
Easier to prove
Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

What properties should we robustly preserve?


relational hyperproperties
(trace equivalence)
new
hyperproperties
(noninterference)
+ data confidentiality
trace properties
(safety & liveness)
only integrity
Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)
No one-size-fits-all security criterion
More secure
More efficient to enforce
Easier to prove

What properties should we robustly preserve?

relational hyperproperties
(trace equivalence)

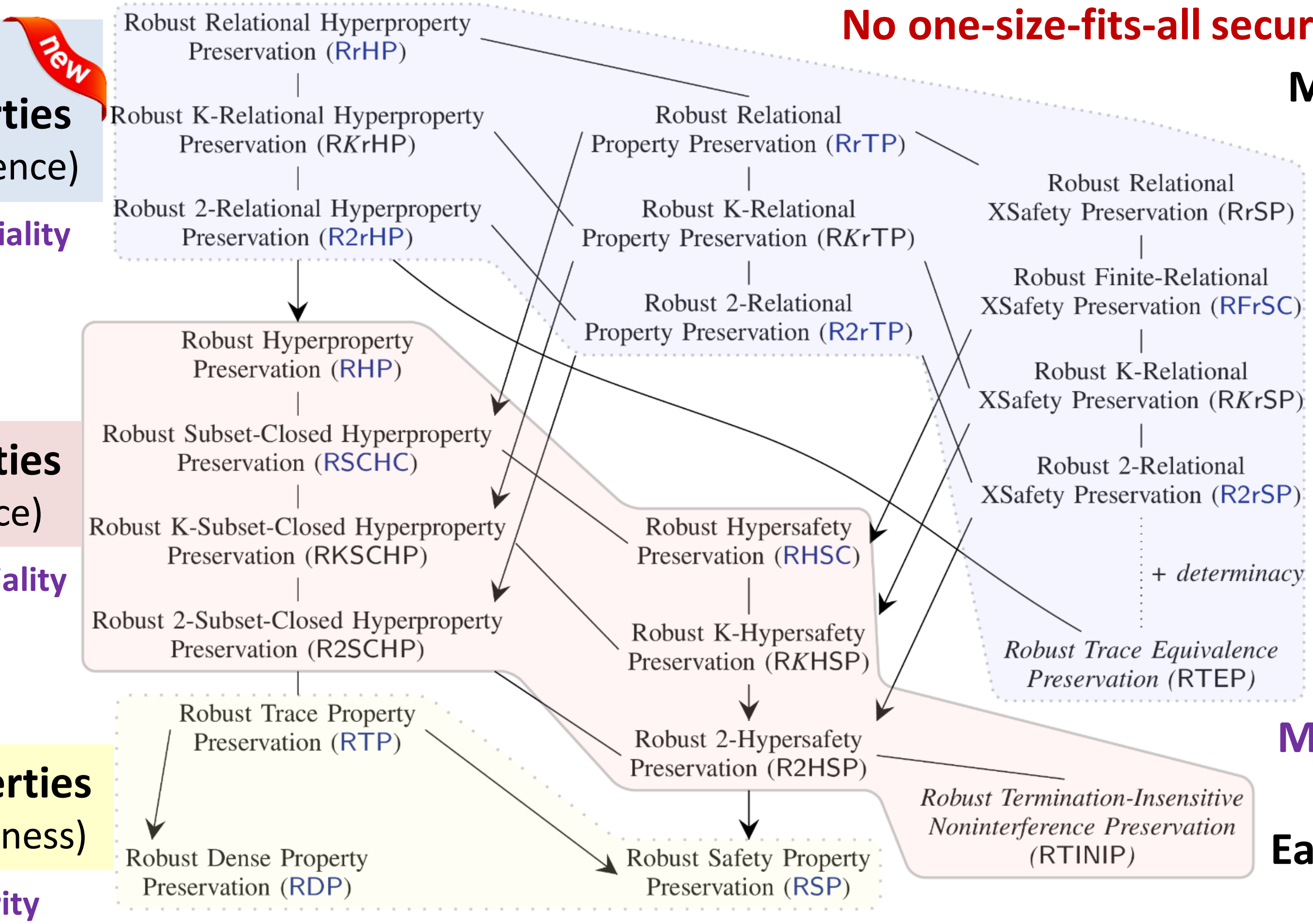
+ code confidentiality

hyperproperties
(noninterference)

+ data confidentiality

trace properties
(safety & liveness)

only integrity


new
Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

No one-size-fits-all security criterion

More secure

More efficient to enforce

Easier to prove

What properties should we robustly preserve?

relational hyperproperties
(trace equivalence)

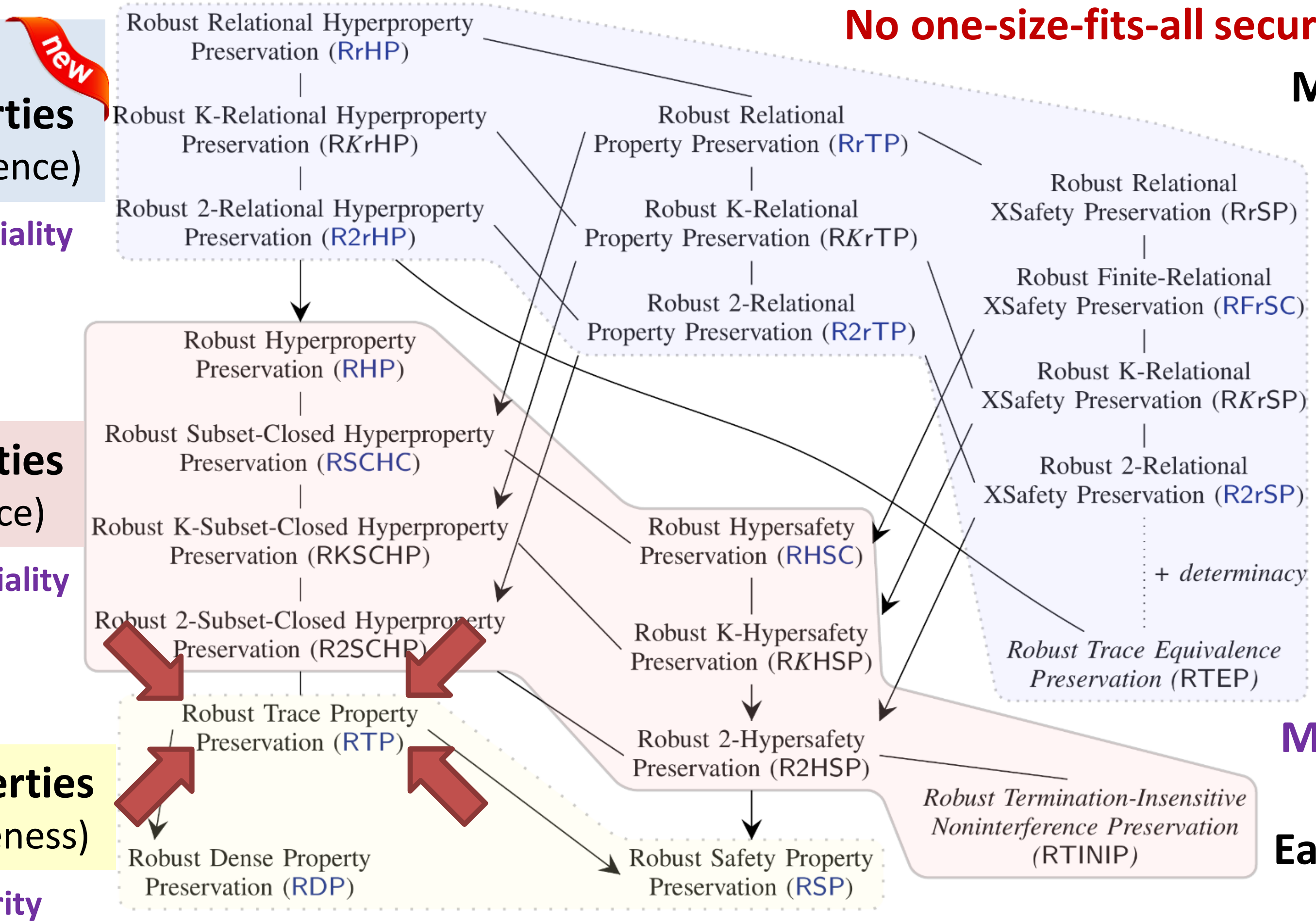
+ code confidentiality

hyperproperties
(noninterference)

+ data confidentiality

trace properties
(safety & liveness)

only integrity


new
Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

No one-size-fits-all security criterion

More secure

More efficient to enforce

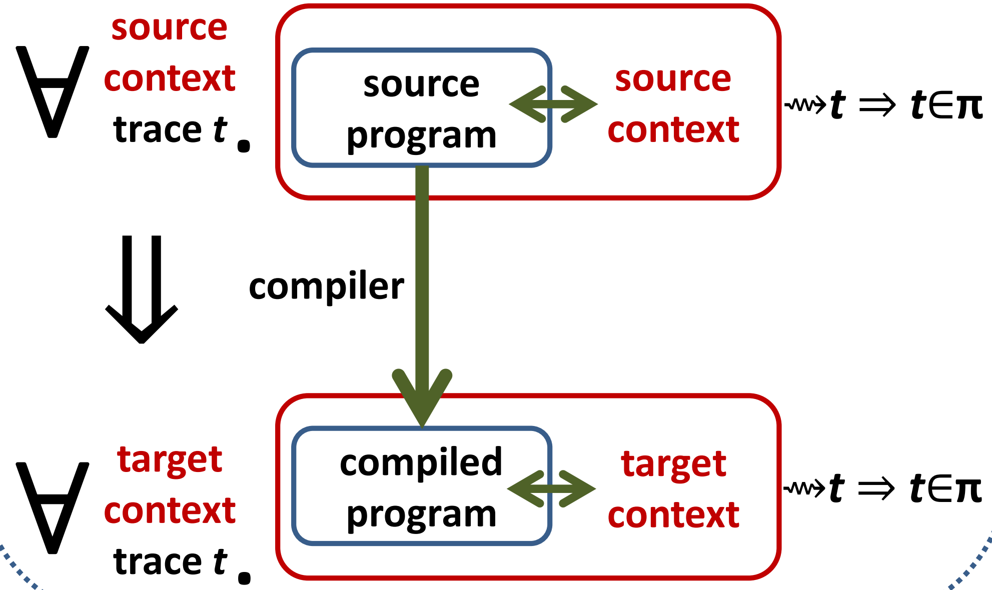
Easier to prove

Robust Trace Property Preservation

property-based characterization

∀source programs.

∀π trace property.



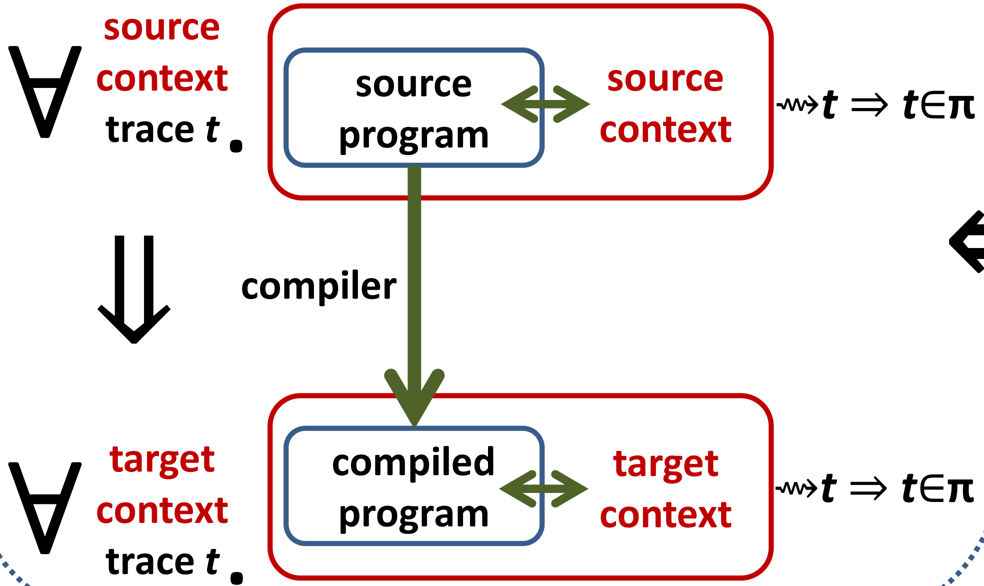
what one might want to achieve

Robust Trace Property Preservation

property-based characterization

∀source programs.

∀π trace property.



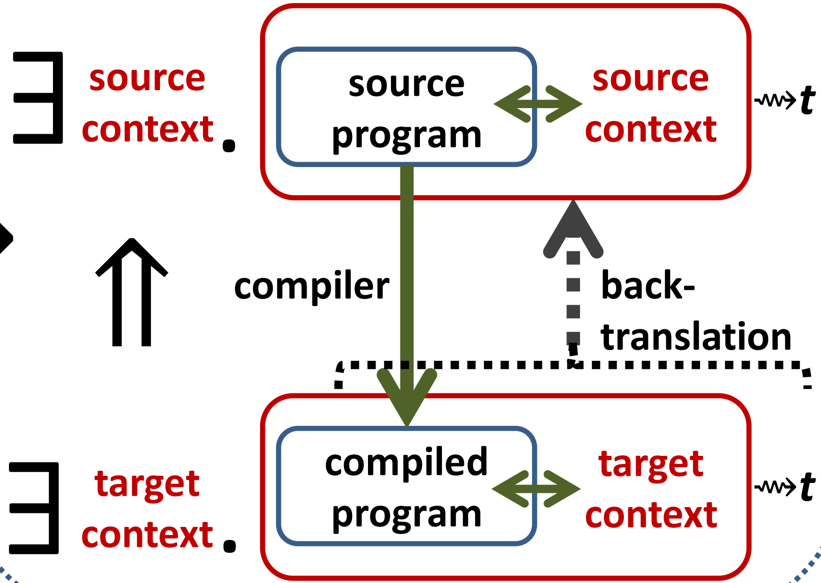
∀ source context trace t.
source program
source context
$\rightsquigarrow t \Rightarrow t \in \pi$
⇓
compiler
∀ target context trace t.
compiled program
target context
$\rightsquigarrow t \Rightarrow t \in \pi$

⇔

property-free characterization

∀source programs.

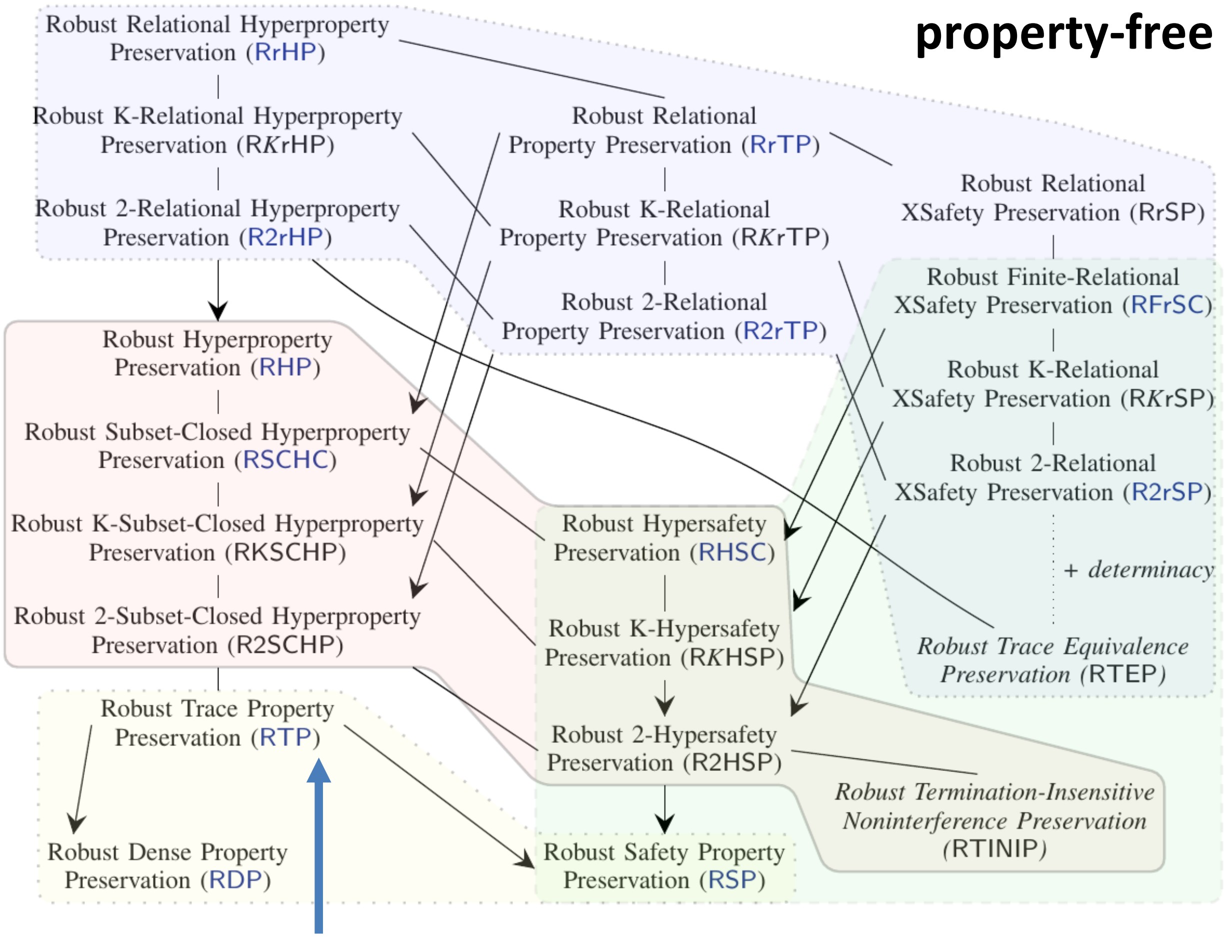
∀(bad/attack) trace *t*.



what one might want to achieve

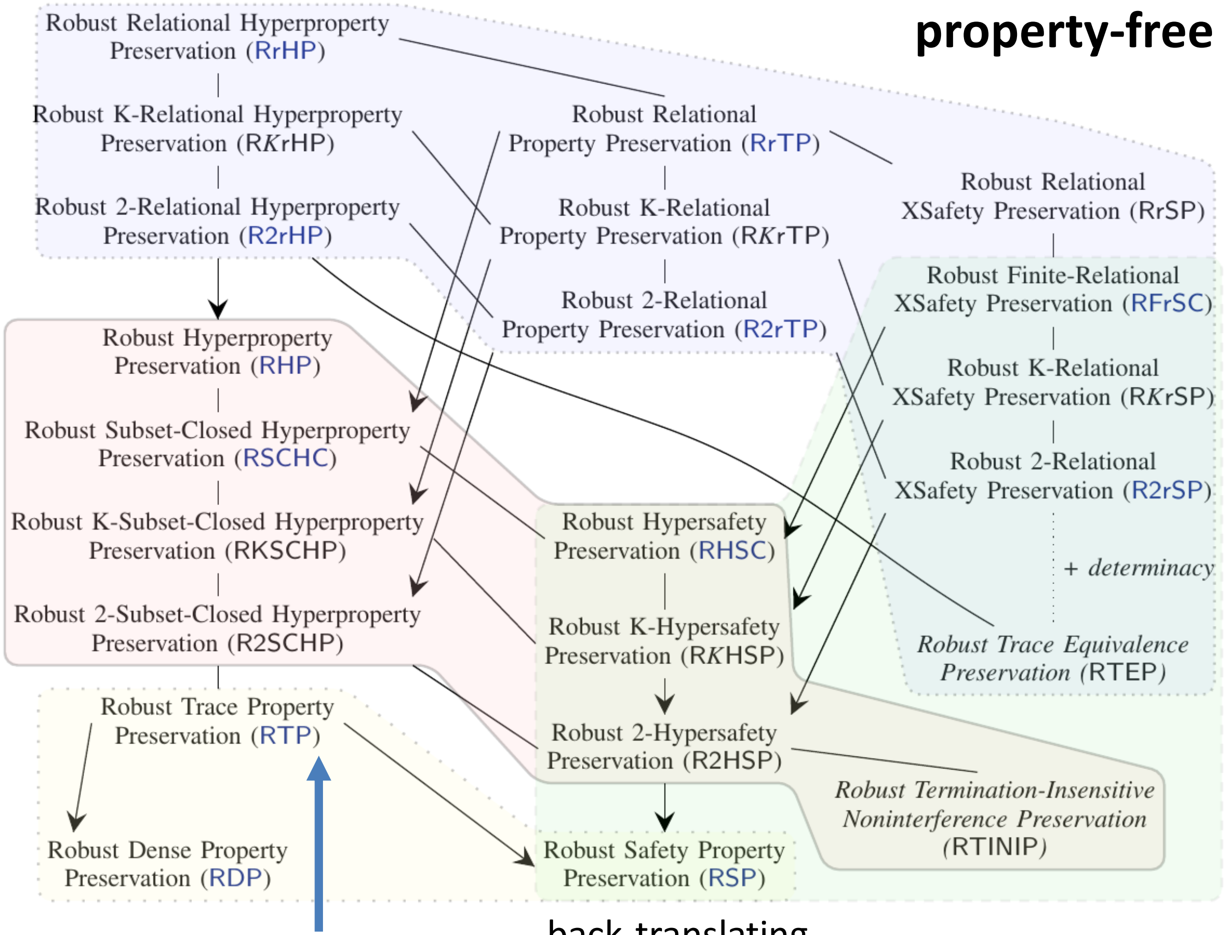
how one can prove it

Some of the proof difficulty is manifest in property-free characterization


Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

back-translating
prog & context & trace
$\forall P \forall C_T \forall t \exists C_S \ldots$

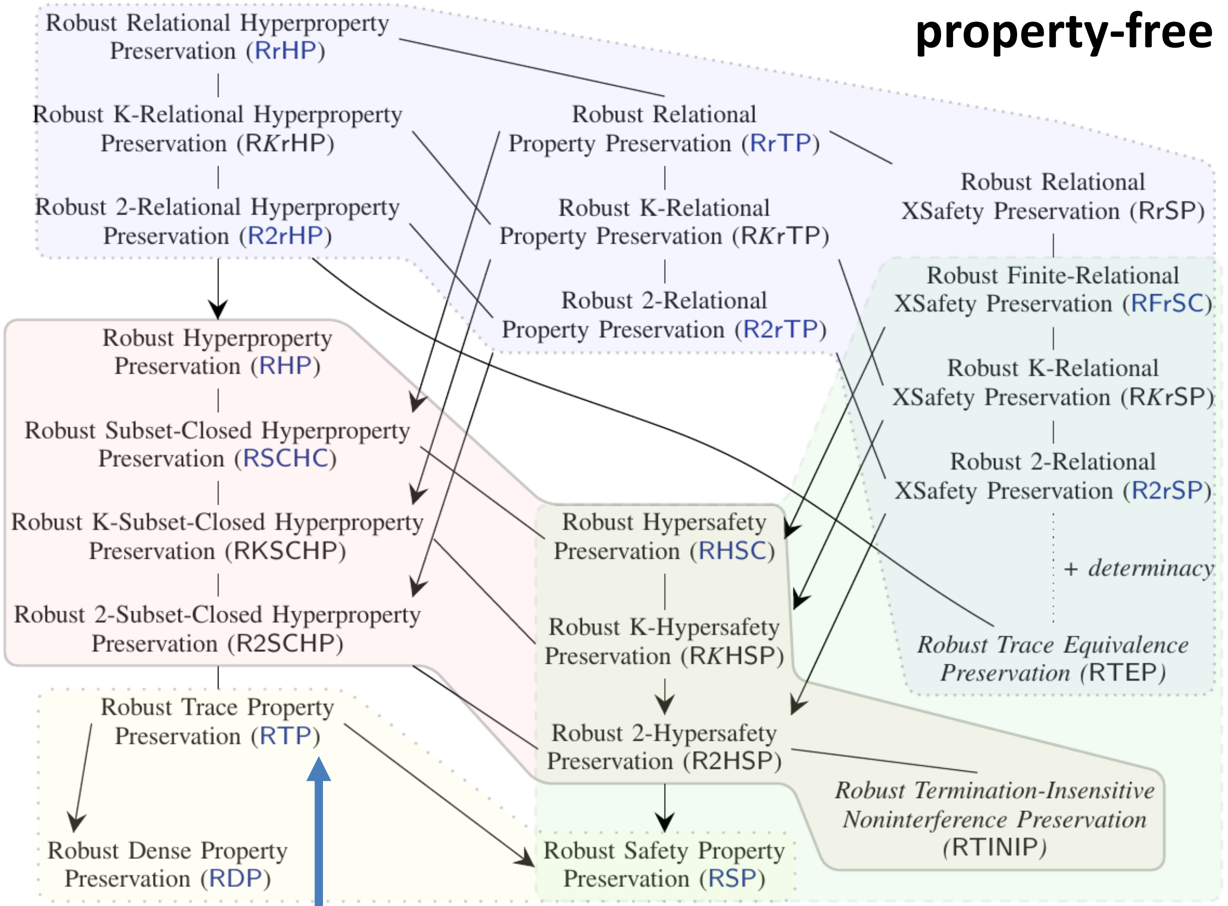
Some of the proof difficulty is manifest in property-free characterization


Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

back-translating
prog & context & trace
$\forall P \forall C_T \forall t \exists C_S \ldots$

back-translating
finite trace prefix
$\forall P \forall C_T \forall m \leq t \exists C_S \ldots$

Some of the proof difficulty is manifest in property-free characterization



Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

back-translating
finite set of
finite trace prefixes
$\forall k \forall P_1..P_k \forall C_T$
$\forall m_1..m_k \exists C_S$...

back-translating
prog & context & trace
$\forall P \forall C_T \forall t \exists C_S$...

back-translating
finite trace prefix
$\forall P \forall C_T \forall m \leq t \exists C_S$...

Some of the proof difficulty is manifest in property-free characterization

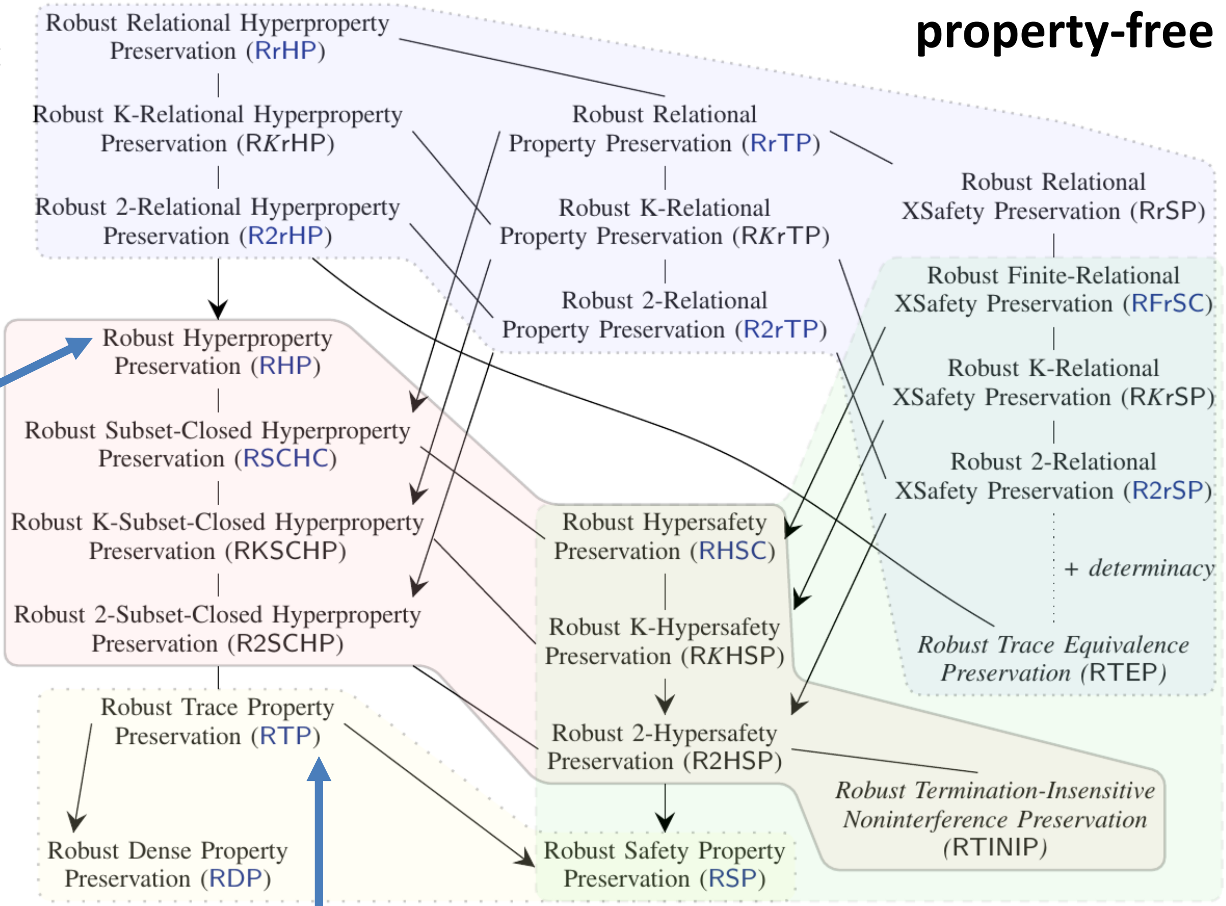
back-translating context
$\forall C_T \exists C_S \forall P \forall t\ldots$

back-translating prog & context
$\forall P \forall C_T \exists C_S \forall t\ldots$

back-translating prog & context & trace
$\forall P \forall C_T \forall t \exists C_S\ldots$

back-translating finite trace prefix
$\forall P \forall C_T \forall m \leq t \exists C_S\ldots$

back-translating finite set of finite trace prefixes
$\forall k \forall P_1..P_k \forall C_T$
$\forall m_1..m_k \exists C_S\ldots$



Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

Journey Beyond Full Abstraction

- **First to explore space of secure compilation criteria** based on robust property preservation

Journey Beyond Full Abstraction

- **First to explore space of secure compilation criteria** based on robust property preservation
- **Carefully studied the criteria and their relations**
  - Property-free characterizations
  - implications, **collapses**, **separations results**

Journey Beyond Full Abstraction

- **First to explore space of secure compilation criteria** based on robust property preservation
- **Carefully studied the criteria and their relations**
  - Property-free characterizations
  - implications, **collapses**, **separations results**
- **Introduced relational (hyper)properties (new classes!)**

Journey Beyond Full Abstraction

- **First to explore space of secure compilation criteria** based on robust property preservation
- **Carefully studied the criteria and their relations**
  - Property-free characterizations
  - implications, **collapses**, **separations results**
- **Introduced relational (hyper)properties (new classes!)**
- **Clarified relation to full abstraction ...**

Journey Beyond Full Abstraction

- **First to explore space of secure compilation criteria** based on robust property preservation
- **Carefully studied the criteria and their relations**
  - Property-free characterizations
  - implications, **collapses**, **separations results**
- **Introduced relational (hyper)properties (new classes!)**
- **Clarified relation to full abstraction ...**
- **Embraced and extended proof techniques ...**

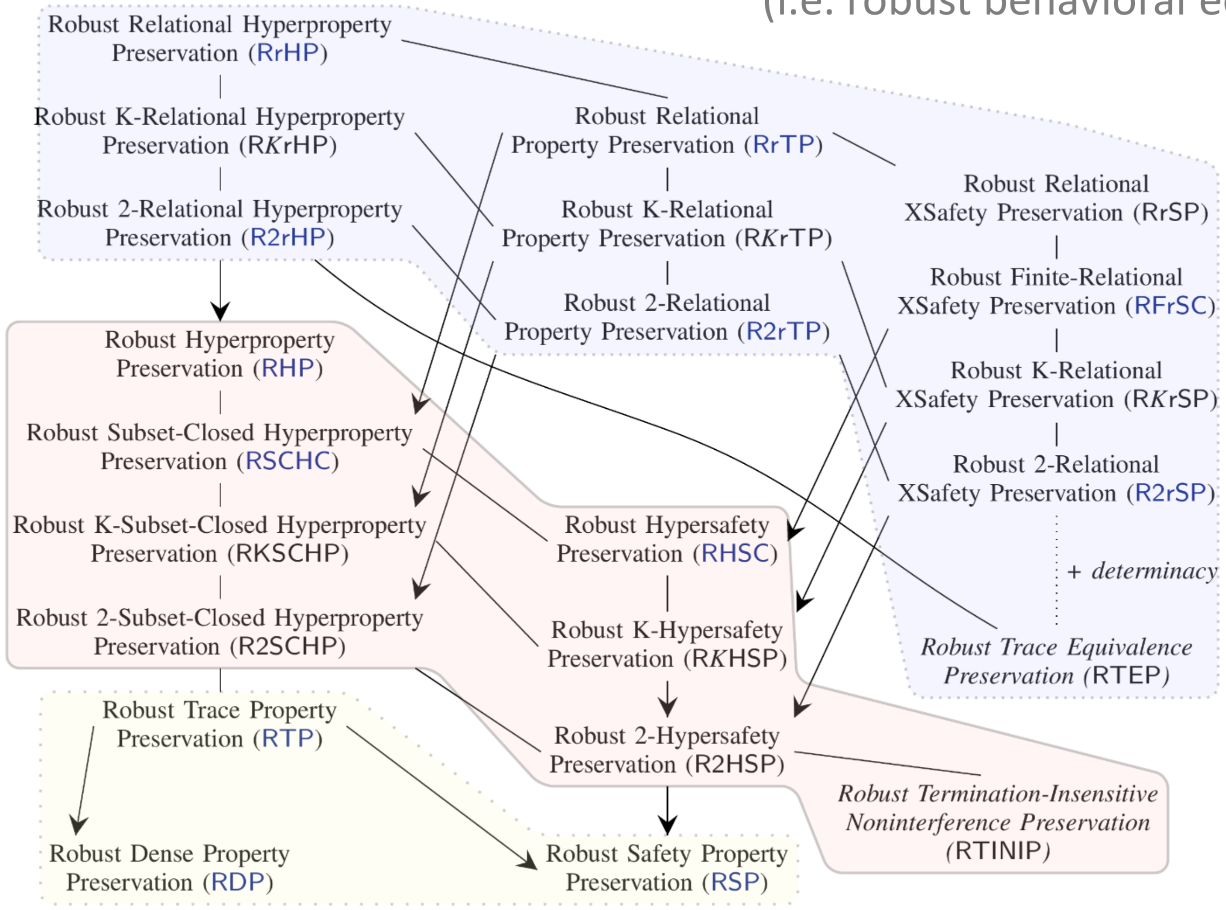
Journey Beyond Full Abstraction

- **First to explore space of secure compilation criteria** based on robust property preservation
- **Carefully studied the criteria and their relations**
  - Property-free characterizations
  - implications, **collapses**, **separations results**
- **Introduced relational (hyper)properties (new classes!)**
- **Clarified relation to full abstraction ...**
- **Embraced and extended proof techniques ...**

https://github.com/secure-compilation/exploring-robust-property-preservation

Where is Full Abstraction?

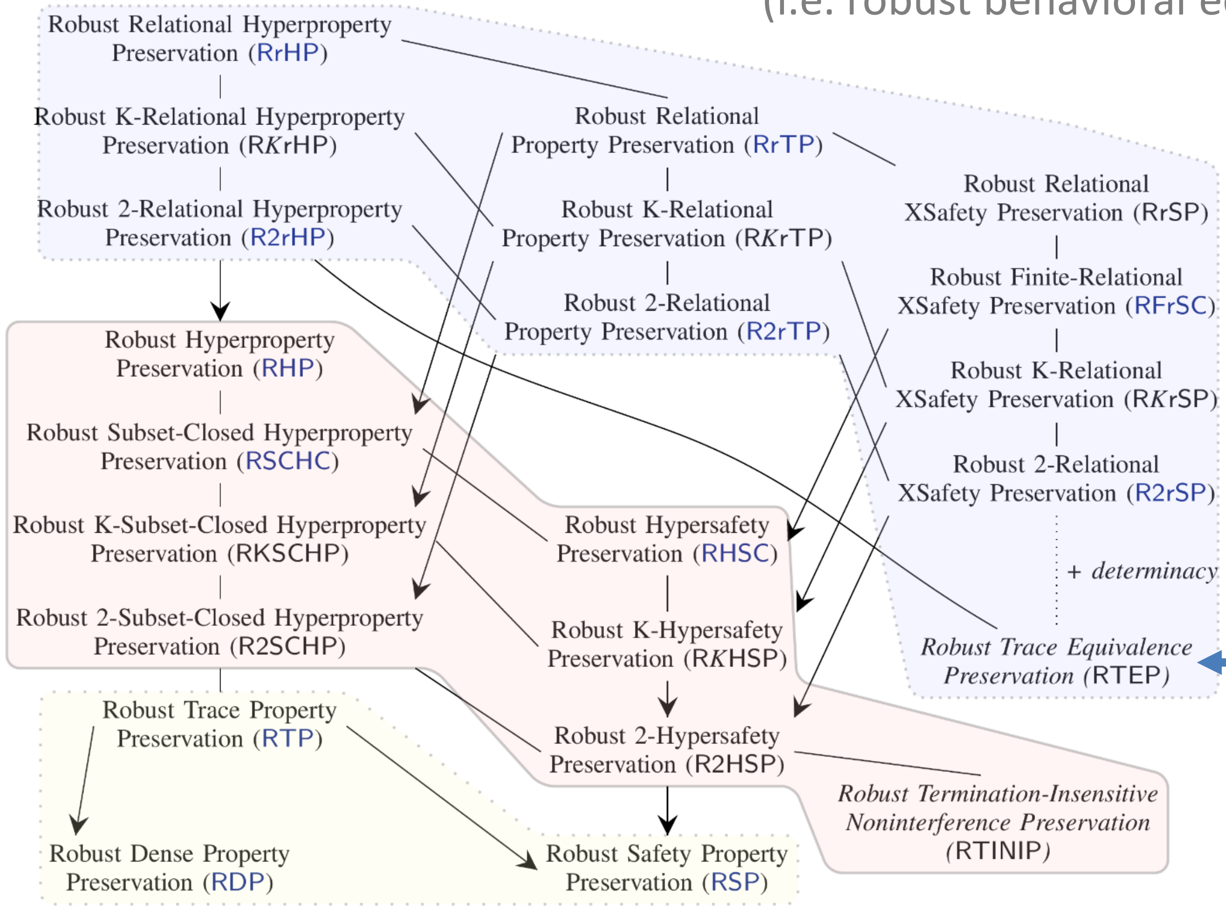
(i.e. robust behavioral equivalence preservation)



Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

Where is Full Abstraction?

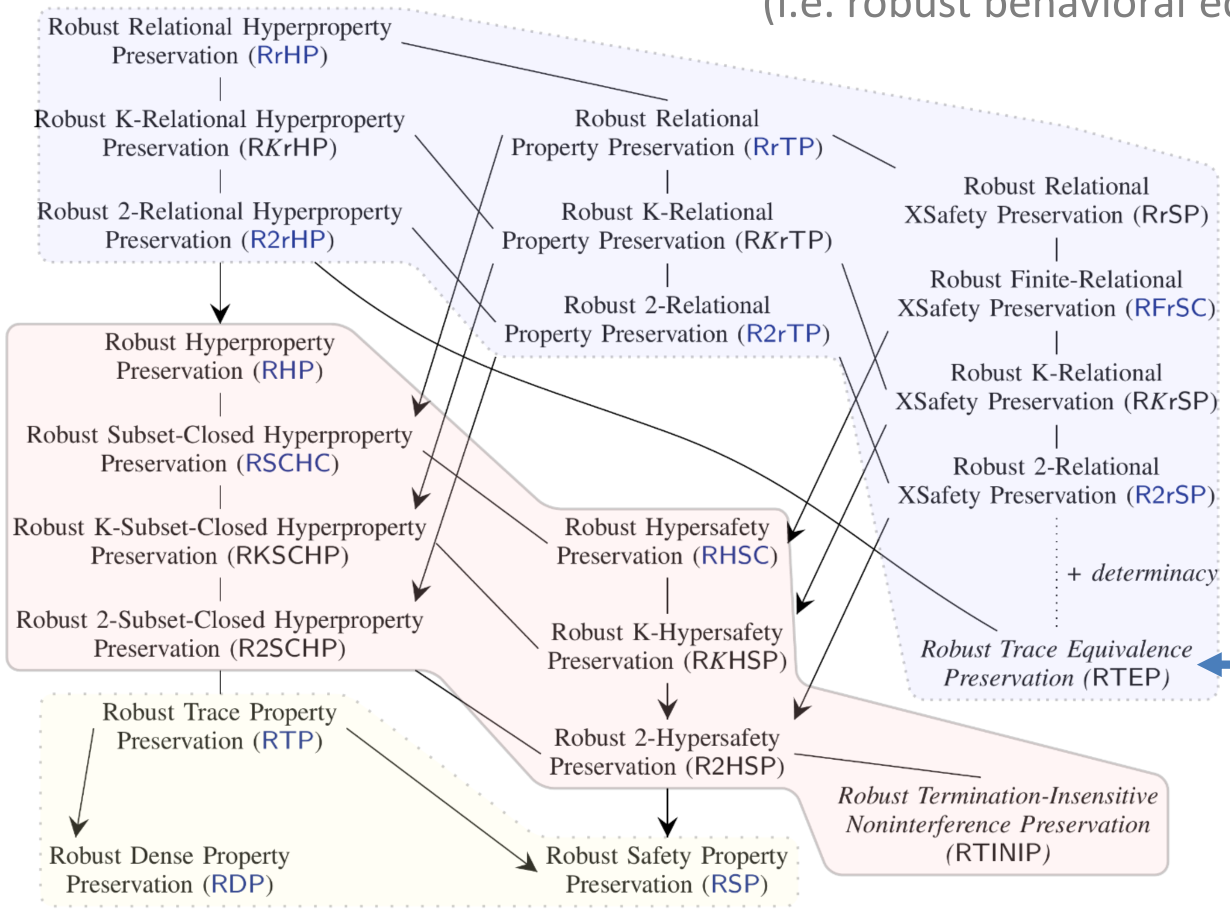
(i.e. robust behavioral equivalence preservation)


Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

without internal nondeterminism,
full abstraction is here

Where is Full Abstraction?

(i.e. robust behavioral equivalence preservation)



Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

without internal nondeterminism, **full abstraction is here**

doesn't imply any other criterion

Full abstraction does not imply any other criterion in our diagram

Full abstraction does not imply any other criterion in our diagram

- **Intuitive counterexample** adapted from Marco&Deepak [CSF'17]

Full abstraction **does not** imply any other criterion in our diagram

- **Intuitive counterexample** adapted from Marco&Deepak [CSF'17]
- **When context passes in bad input value** (e.g. ill-typed):

Full abstraction **does not** imply any other criterion in our diagram

- **Intuitive counterexample** adapted from Marco&Deepak [CSF'17]
- **When context passes in bad input value** (e.g. ill-typed):
  - **lunch the missiles** - breaks Robust Safety Preservation

Full abstraction **does not** imply any other criterion in our diagram

- **Intuitive counterexample** adapted from Marco&Deepak [CSF'17]
- **When context passes in bad input value** (e.g. ill-typed):
  - **lunch the missiles** - breaks Robust Safety Preservation
  - or **loop forever** - breaks Robust Liveness Preservation

Full abstraction **does not** imply any other criterion in our diagram

- **Intuitive counterexample** adapted from Marco&Deepak [CSF'17]
- **When context passes in bad input value** (e.g. ill-typed):
  - **lunch the missiles** - breaks Robust Safety Preservation
  - or **loop forever** - breaks Robust Liveness Preservation
  - or **leak secret inputs** - breaks Robust NI Preservation

Full abstraction does not imply any other criterion in our diagram

- **Intuitive counterexample** adapted from Marco&Deepak [CSF'17]
- **When context passes in bad input value** (e.g. ill-typed):
  - **lunch the missiles** - breaks Robust Safety Preservation
  - or **loop forever** - breaks Robust Liveness Preservation
  - or **leak secret inputs** - breaks Robust NI Preservation
- **Yet this doesn't break full abstraction** or **compiler correctness!**

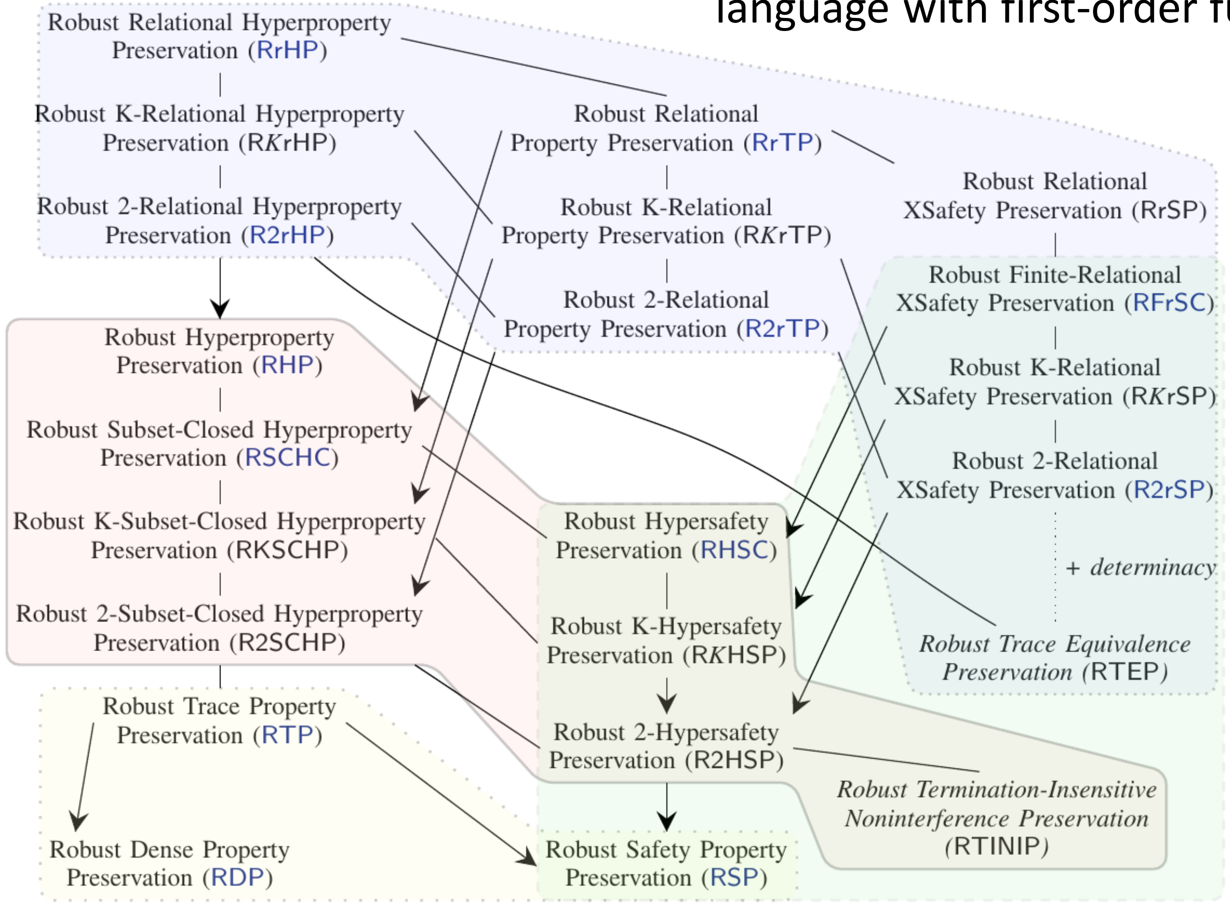
Full abstraction does not imply any other criterion in our diagram

- **Intuitive counterexample** adapted from Marco&Deepak [CSF'17]
- **When context passes in bad input value** (e.g. ill-typed):

  - **lunch the missiles** - breaks Robust Safety Preservation
  - or **loop forever** - breaks Robust Liveness Preservation
  - or **leak secret inputs** - breaks Robust NI Preservation
- **Yet this doesn't break full abstraction** or **compiler correctness!**
- Full abstraction only ensures **code confidentiality**
  - **no** integrity, **no** safety, **no** data confidentiality, ...

Embraced and extended™ proof techniques

for simple translation from statically to dynamically typed language with first-order functions and I/O



Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

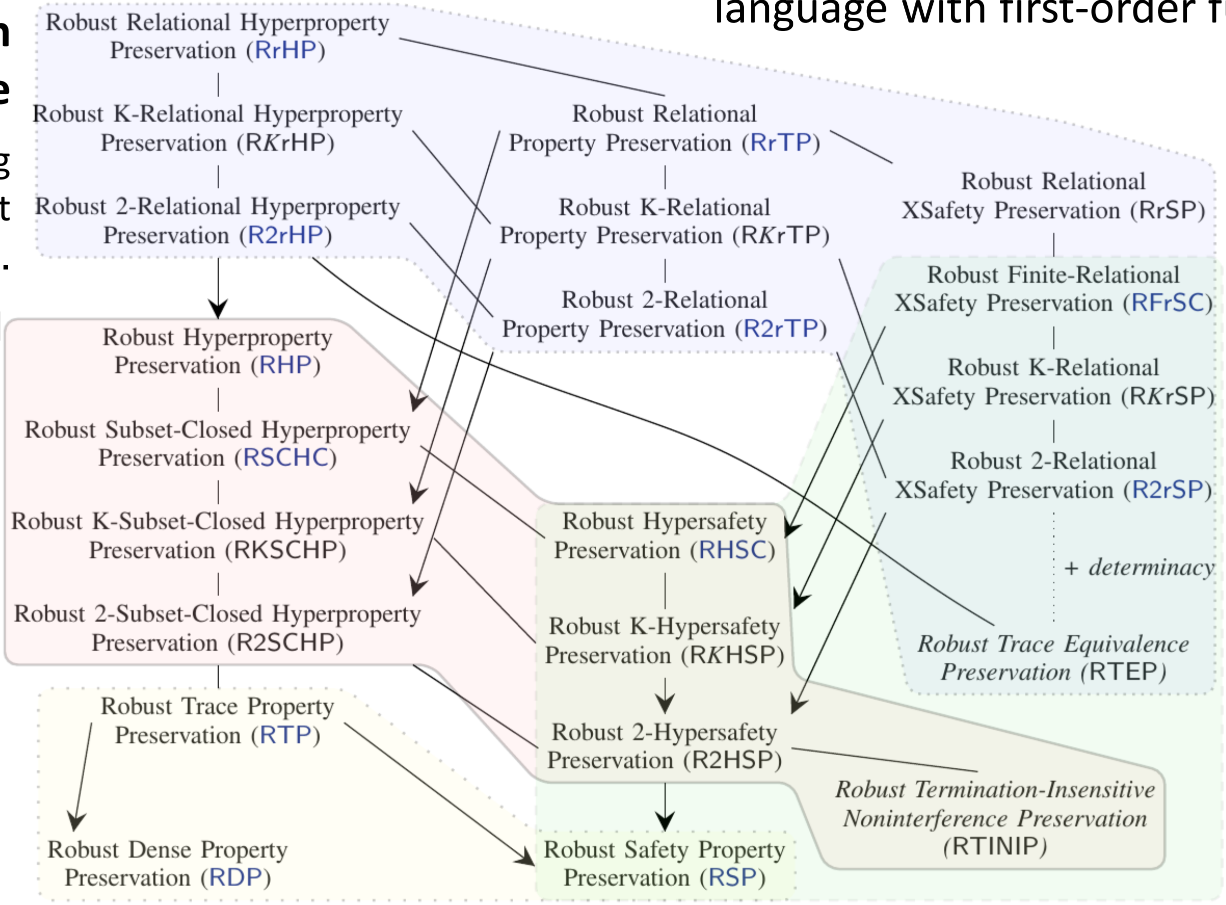
Embraced and extended™ proof techniques

for simple translation from statically to dynamically typed language with first-order functions and I/O

strongest criterion achievable

back-translating context $\forall C_T \exists C_S \forall P \forall t \ldots$

[New et al,ICFP'16]


Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

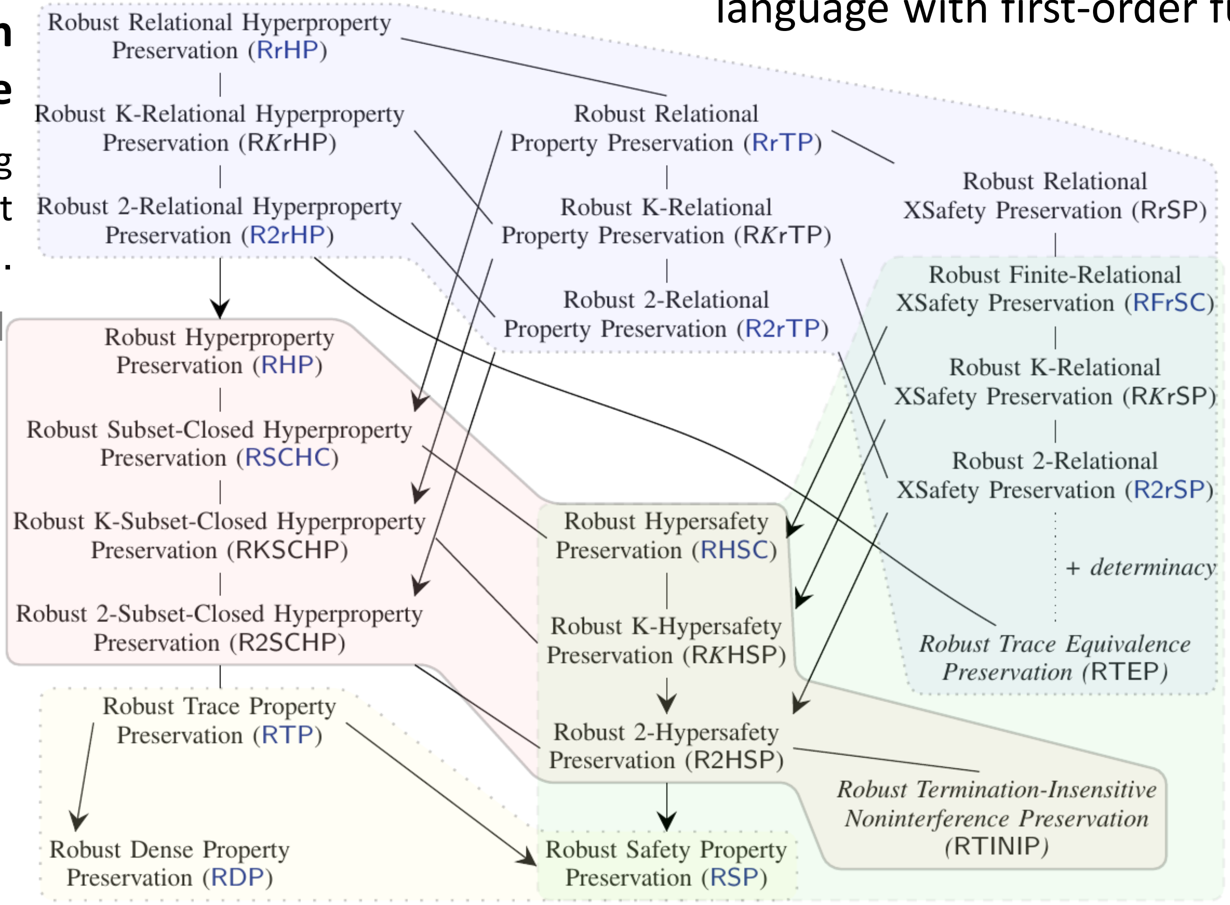
Embraced and extended™ proof techniques

for simple translation from statically to dynamically typed language with first-order functions and I/O

strongest criterion achievable

back-translating context $\forall C_T \exists C_S \forall P \forall t$...

[New et al,ICFP'16]



Robust Relational Hyperproperty Preservation (RrHP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust Relational Property Preservation (RrTP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Relational XSafety Preservation (RrSP)
Robust Finite-Relational XSafety Preservation (RFrSC)
Robust K-Relational XSafety Preservation (RKrSP)
Robust 2-Relational XSafety Preservation (R2rSP)
+ determinacy
Robust Trace Equivalence Preservation (RTEP)
Robust Hyperproperty Preservation (RHP)
Robust Subset-Closed Hyperproperty Preservation (RSCHC)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust Hypersafety Preservation (RHSC)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Trace Property Preservation (RTP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

generic technique applicable

back-translating finite set of finite trace prefixes $\forall k \forall P_1..P_k \forall C_T \forall m_1..m_k \exists C_S$...

[Jeffrey & Rathke, ESOP'05]
[Patrignani et al,TOPLAS'15]

Some open problems

- **Practically achieving**
  secure interoperability with lower-level code
  - more realistic languages and compilation chains

Some open problems

- **Practically achieving**
  secure interoperability with lower-level code
  – more realistic languages and compilation chains
- **Verifying robust satisfaction for source programs**
  – program logics, logical relations, partial semantics, ...

Some open problems

- **Practically achieving**
  secure interoperability with lower-level code
  - more realistic languages and compilation chains
- **Verifying robust satisfaction for source programs**
  - program logics, logical relations, partial semantics, ...
- **Different traces for source and target semantics**
  - connected by some arbitrary relation
  - mappings between source and target properties
  - interesting even for correct compilation

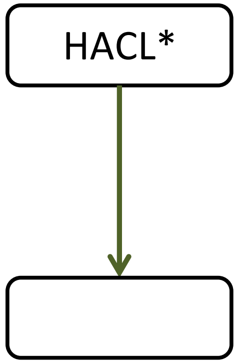
My dream: secure compilation at scale

language

HACL*

My dream: secure compilation at scale

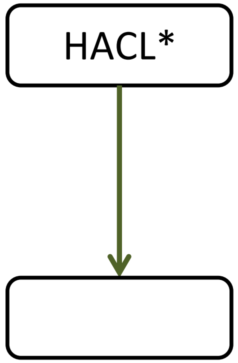
language



C language
+ components
+ memory safety

My dream: secure compilation at scale

language



C language

+ components

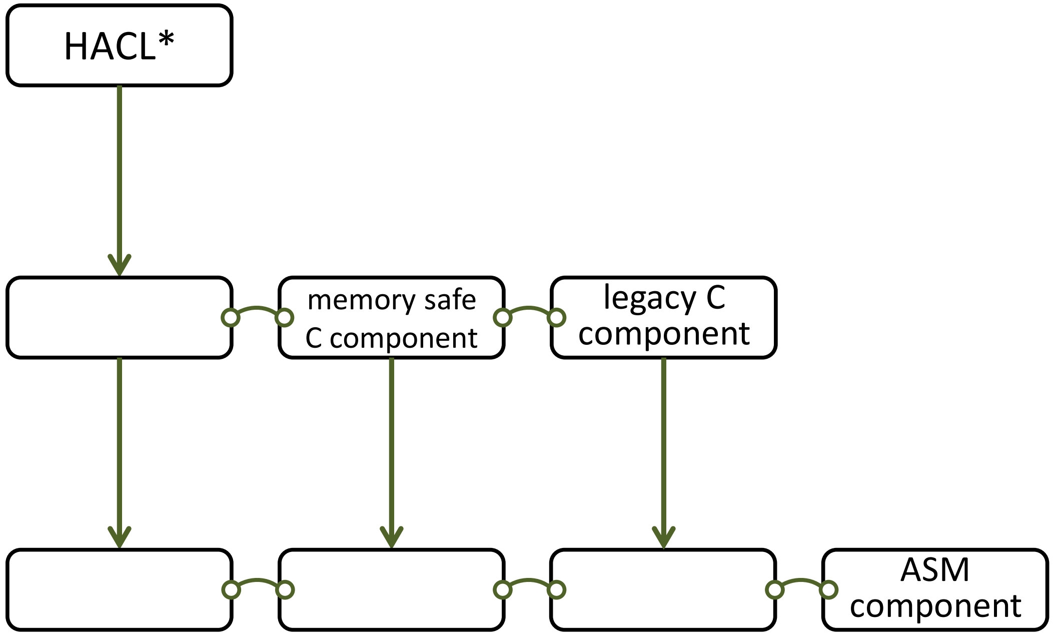
+ memory safety

My dream: secure compilation at scale

language

C language
+ components
+ memory safety

ASM language
(RISC-V + micro-policies)



My dream: secure compilation at scale

language

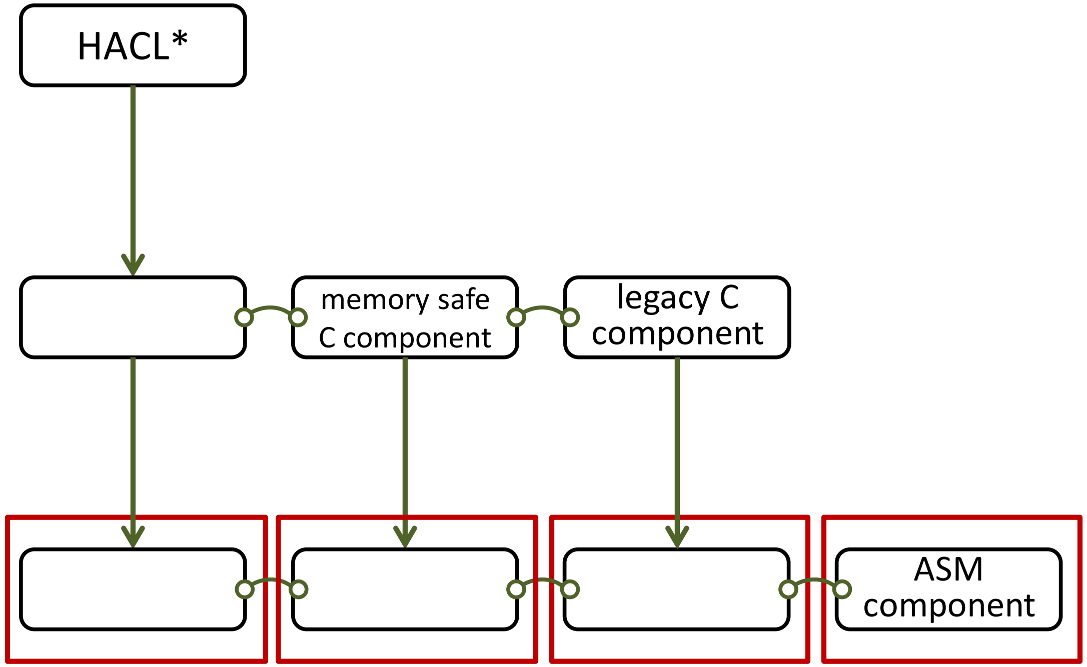
C language

+ components

+ memory safety

ASM language

(RISC-V + micro-policies)

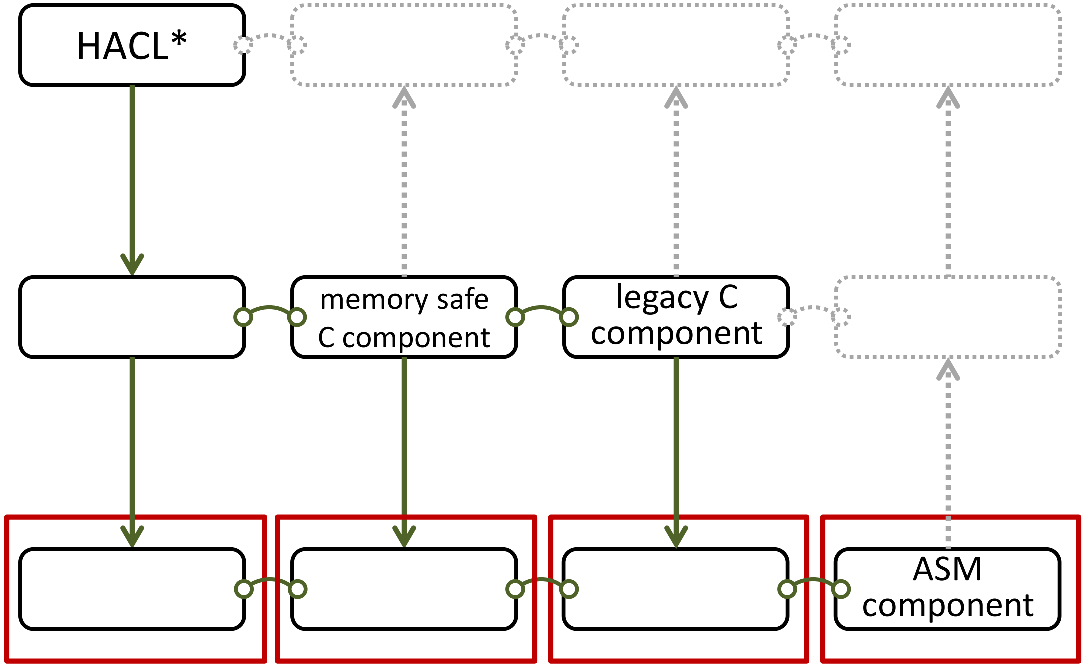


My dream: secure compilation at scale

language

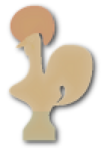
C language
+ components
+ memory safety

ASM language
(RISC-V + micro-policies)



Journey Beyond Full Abstraction

- **First to explore space of secure compilation criteria** based on robust property preservation
- **Carefully studied the criteria and their relations**
  - Property-free characterizations
  - implications, **collapses**, **separations results**
- **Introduced relational (hyper)properties (new classes!)**
- **Clarified relation to full abstraction ...**
- **Embraced and extended proof techniques ...**



https://github.com/secure-compilation/exploring-robust-property-preservation