Using the Robust Compilation Diagram

Marco Patrignani[†,‡]
[†] *Stanford University*
[‡] *CISPA Helmholtz Centre for Information Security*
`mp@cs.stanford.edu` or `squera.patrignani@gmail.com`

Abstract

Given that many researchers are using the diagram on secure compilation criteria from Abate et al. (2019), we provide a few guidelines on how to use it here, as well as the original LaTeX sources.

1 Introduction

If you need the image, the resources file mentioned in Section 1.3 contains some popular choices in good resolution. If you need to customise the diagram, then you likely need to compile your own version and then cut that picture from the pdf that LaTeX generates. Keep reading to know how to do that and how to customise it.

The resources file of Section 1.3 contains the whole diagram (file `lattice.tex`). To compile it, you must also include `lattice-preamble.tex` as well as (possibly) the `cleveref` package to compile. If you just import those files and do not touch anything, you will get a diagram as shown in Figure 1, provided you supply references for acronym names (see Section 1.2 for an explanation).

1.1 Fitting in a Page

Please note that if you need to make it fit in a single-column format, you will likely need to:

- either move it towards the left margin with a combination of negative hspace and a minipage:[1]

  ```
  {
  \hspace*{-13em}
  \begin{minipage}{\textwidth}
  \input{lattice}
  \end{minipage}
  }
  ```

- or scale it down (say to 75%) by adding this code to the tikzpicture command optional parameters (between square brackets):

  ```
  scale=.75, every node/.style={transform shape}
  ```

[1] This is the code used in this paper, which is a LaTeX "article" class. Other classes may have different margins, adjust the hspace accordingly.

1.2 Acronym Names and Links

If you do not want acronym names to be displayed, you can either manually erase them, or uncomment the code below in the `lattice-preamble` file. That sets the acronyms flag to false; normally it is set to true.

```
\newif\ifacr\acrfalse
```

If you want acronym names, the current code is set up in a way that most acronyms are actually *links* to the text where the formal definition is. Thus, you have the choice of keeping them as links or not.

If you want these acronyms to be links, the preamble defines the following commands for you, given a criterion CX:

- `\CXcomp` : this is the name you want displayed for the criterion. The rest of the macros already add a 'P' at the end of this name (or 'C' for property-free criteria);
- `\CX` : this is the command to refer to the criterion, if the link flag is off (see below), this calls `\CXcomp`;
- `\CXref` : this is an alias for the previous command;
- `\CXdef` : this command must be inserted where you define the criterion, it ensures that any usage of `\CX` points to the place where this command is used.

You should insert in your text all occurrences of `\CXdef` so that links from the diagram point to locations in your text. In this document, we added said commands in the list below; links in the diagram point to that list.

- RHP; RSCHP; RHSP; RHLP; RKHSP; R2HSP; RSP; RDP; RrHP; RKrHP; R2rHP; RrSP; RrXP; RrTP; RKrXP; RKrTP; RKrSP; R2rXP; R2rTP; R2rSP; RFrSP; RFrXP; RrHSP; RKSCHP; R2SCHP; RTP; RTEP; RTINIP.

If you do not want the acronyms to be links, then you need to uncomment the line below in your `lattice-preamble` file. That sets the links flag to false; normally it is set to true. Since they are not links, you do not need to add other text in your LaTeX.

```
\newif\iflink\linkfalse
```

1.3 File Sources

The diagram and the preamble files can be found at the following url:

- http://theory.stanford.edu/~mp/mp/Publications_files/rc-diagram.zip

Please note that people change institutions and if these links do not work, check our new institution pages.

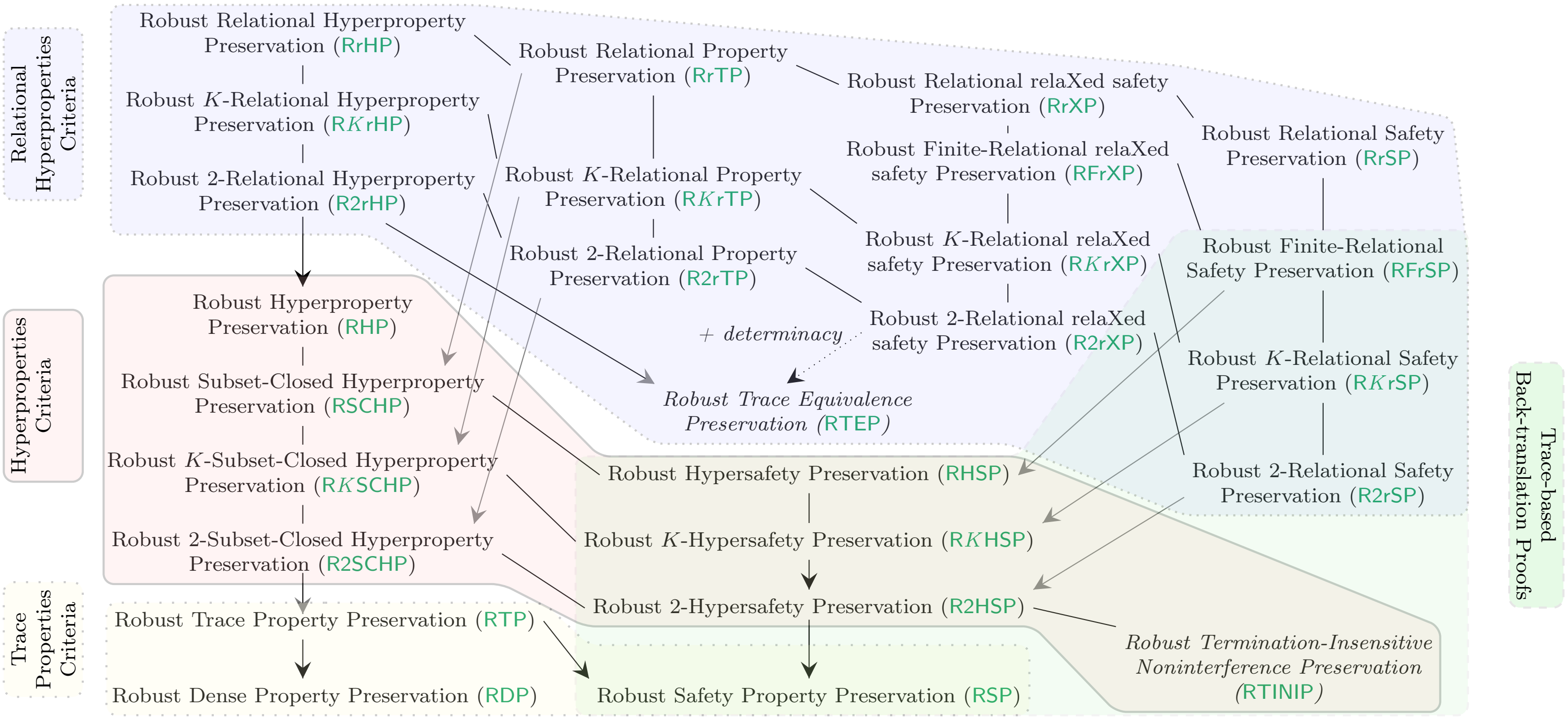
Relational Hyperproperties Criteria
Hyperproperties Criteria
Trace Properties Criteria
Trace-based Back-translation Proofs
Robust Relational Hyperproperty Preservation (RrHP)
Robust Relational Property Preservation (RrTP)
Robust Relational relaXed safety Preservation (RrXP)
Robust Relational Safety Preservation (RrSP)
Robust K-Relational Hyperproperty Preservation (RKrHP)
Robust Finite-Relational relaXed safety Preservation (RFrXP)
Robust K-Relational Property Preservation (RKrTP)
Robust 2-Relational Hyperproperty Preservation (R2rHP)
Robust K-Relational relaXed safety Preservation (RKrXP)
Robust Finite-Relational Safety Preservation (RFrSP)
Robust 2-Relational Property Preservation (R2rTP)
Robust Hyperproperty Preservation (RHP)
+ determinacy
Robust 2-Relational relaXed safety Preservation (R2rXP)
Robust K-Relational Safety Preservation (RKrSP)
Robust Subset-Closed Hyperproperty Preservation (RSCHP)
Robust Trace Equivalence Preservation (RTEP)
Robust K-Subset-Closed Hyperproperty Preservation (RKSCHP)
Robust Hypersafety Preservation (RHSP)
Robust 2-Relational Safety Preservation (R2rSP)
Robust K-Hypersafety Preservation (RKHSP)
Robust 2-Subset-Closed Hyperproperty Preservation (R2SCHP)
Robust 2-Hypersafety Preservation (R2HSP)
Robust Trace Property Preservation (RTP)
Robust Termination-Insensitive Noninterference Preservation (RTINIP)
Robust Dense Property Preservation (RDP)
Robust Safety Property Preservation (RSP)

Figure 1: Diagram for robust compilation criteria.

References

Abate, Carmine, Roberto Blanco, Deepak Garg, Cătălin Hriţcu, Marco Patrignani, and Jérémy Thibault (2019). “Journey Beyond Full Abstraction: Exploring Robust Property Preservation for Secure Compilation”. In: *2019 IEEE 32th Computer Security Foundations Symposium*. CSF 2019.