UNIVERSITY
OF BERN

Temporal Logics Meet Real-World Software Requirements: A Reality Check

13th International Conference on Formal Methods in Software Engineering (FormaliSE)

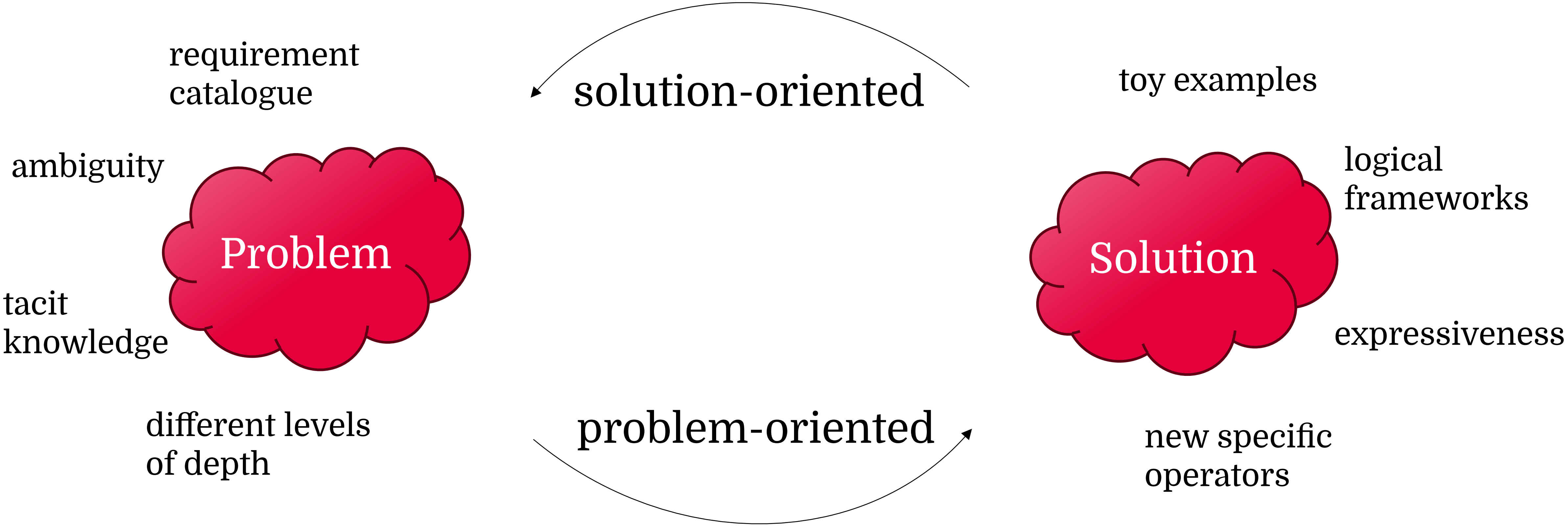
2025-04-27, Ottawa CA

Roman Bögli [*] · Atefeh Rohani [*] · Thomas Studer [*] · Christos Tsigkanos [◇] · Timo Kehrer [*]

[*] University of Bern, Switzerland · [◇] University of Athens, Greece

Introduction

Formalizing Software Requirements



requirement catalogue

solution-oriented

toy examples

ambiguity

Problem

Solution

logical frameworks

tacit knowledge

expressiveness

different levels of depth

problem-oriented

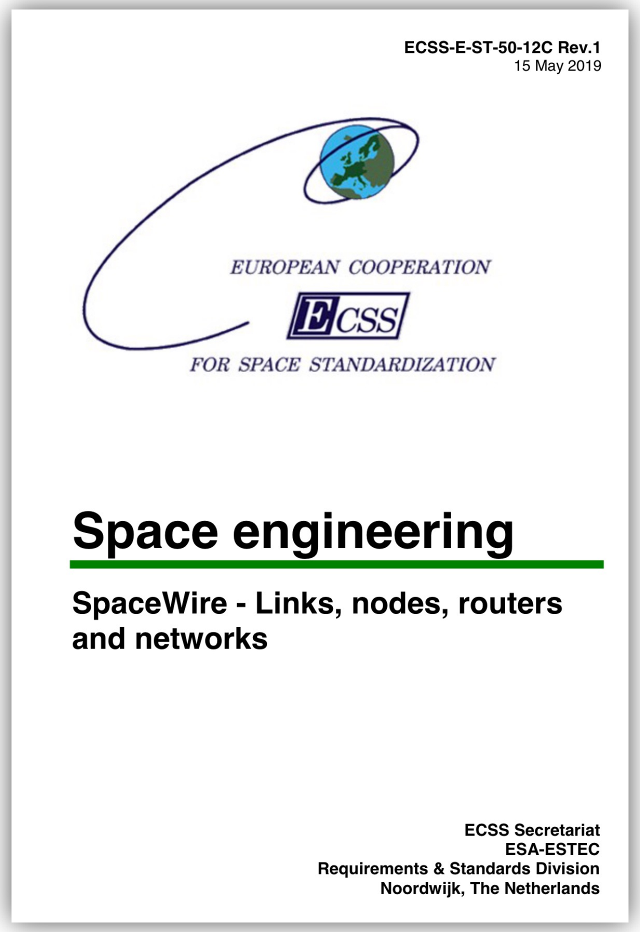
new specific operators

Our Study Scope: Temporal Logics

- Jungle of logics

Our Study Subject: SpaceWire Protocol

- Standard specification for a data handling network (e.g. on spacecrafts)



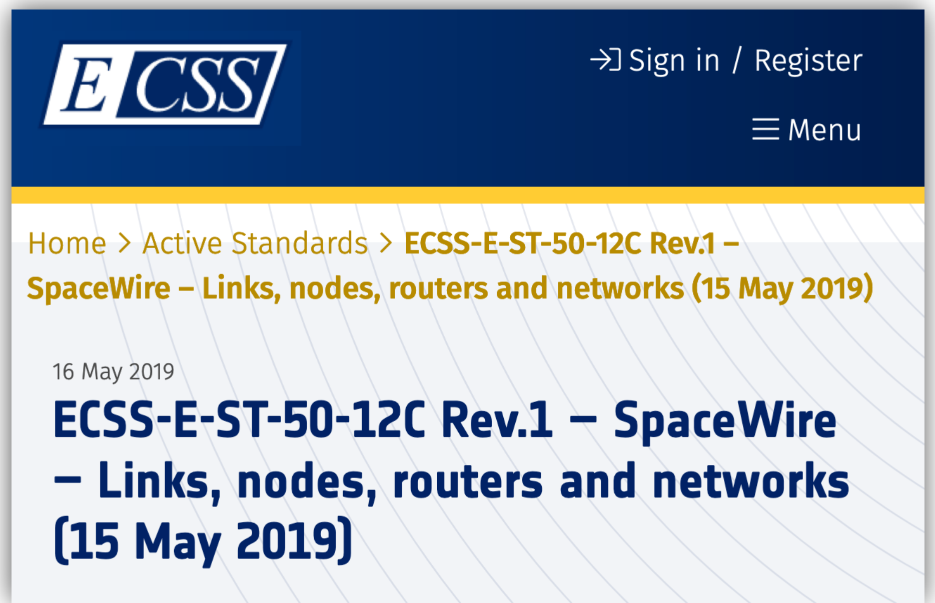
ECSS-E-ST-50-12C Rev.1
15 May 2019

EUROPEAN COOPERATION
ECSS
FOR SPACE STANDARDIZATION

Space engineering

SpaceWire - Links, nodes, routers and networks

ECSS Secretariat
ESA-ESTEC
Requirements & Standards Division
Noordwijk, The Netherlands



Home > Active Standards > ECSS-E-ST-50-12C Rev.1 – SpaceWire – Links, nodes, routers and networks (15 May 2019)

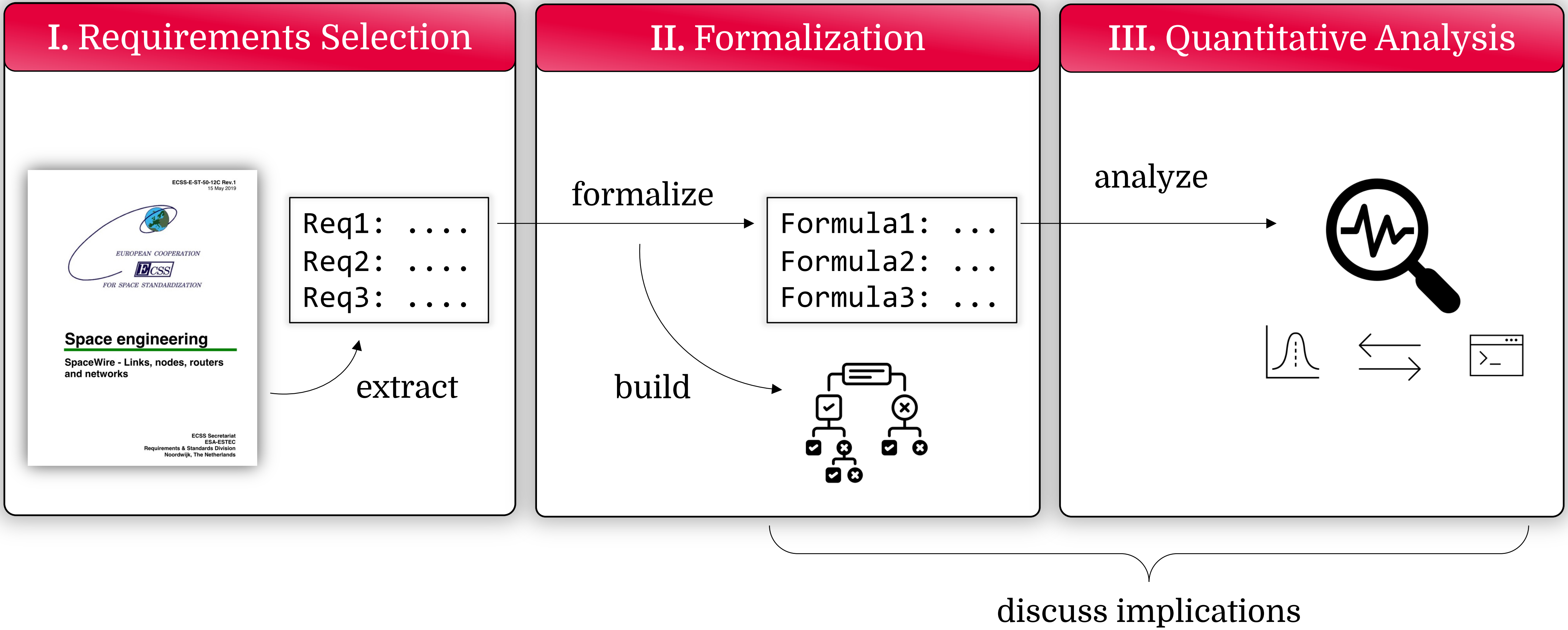
16 May 2019

ECSS-E-ST-50-12C Rev.1 – SpaceWire – Links, nodes, routers and networks (15 May 2019)

Methodology

Methodology

Overview



I. Requirements Selection

ECSS-E-ST-50-12C Rev.1
15 May 2019

EUROPEAN COOPERATION
ECSS
FOR SPACE STANDARDIZATION

Space engineering

SpaceWire - Links, nodes, routers and networks

ECSS Secretariat
ESA-ESTEC
Requirements & Standards Division
Noordwijk, The Netherlands

extract

Req1: ....
Req2: ....
Req3: ....

II. Formalization

formalize

Formula1: ...
Formula2: ...
Formula3: ...

build

III. Quantitative Analysis

analyze

discuss implications

Methodology

I. Requirement Selection

- Functional software requirements with notion of temporal behavior.

- Examples

✅ “Null detection shall be enabled whenever the receiver is enabled.”

⛔ “The line receiver shall maintain correct operation for differential input voltages of up to 600 mV magnitude.”

non-functional

⛔ “Zero or more data characters at the front of a packet shall form a destination address.”

no temporal notion

Methodology

II. Formalization

- Striving for natural formalizations
- Criteria
  - Solely based on temporal operators present in the requirement.
  - Used logic is minimal, i.e., just expressive enough to capture the requirement.
  - Compact formulizations are favored over longer ones.

- Example "*Between now and n, it should always be A.*"
  - LTL

$$A \wedge \mathcal{X} A \wedge \mathcal{X}(\mathcal{X} A) \wedge \cdots \underbrace{\mathcal{X}(\mathcal{X}(\mathcal{X}(\cdots \mathcal{X}}_{n \text{ times}} A)))$$

  - MITL

more natural

$$\Box_{(0,n)} A$$

Methodology

III. Quantitative Analysis – RQ1

What is the distribution of natural logics used for the transcribed SpaceWire requirements, and can they be mutually translated?

- Motivation
  - Prevalence, trends, outliers
  - Framework restrictions, tool support

- Example
  - LTL
  - MITL

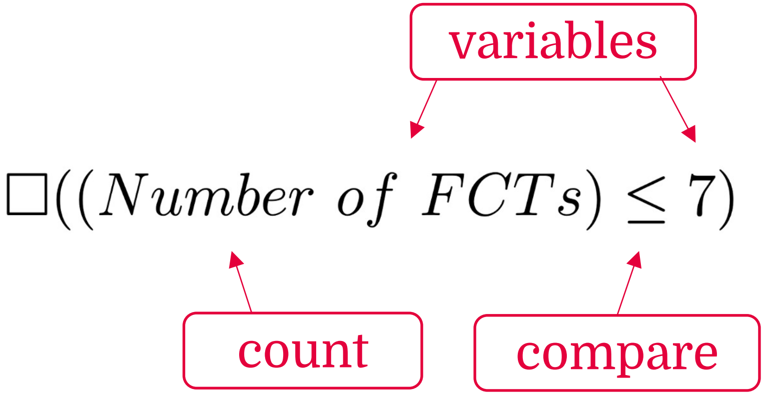
$$A \wedge \mathcal{X} A \wedge \mathcal{X}(\mathcal{X} A) \wedge \cdots \underbrace{\mathcal{X}(\mathcal{X}(\mathcal{X}(\cdots \mathcal{X}}_{n \text{ times}} A))) \xleftrightarrow{\boxed{\text{translates}}} \Box_{(0,n)} A$$

Methodology

III. Quantitative Analysis – RQ2

What is the engineering complexity of the natural formulae for a transcribed SpaceWire requirement, and does it differ among the logics?

- Motivation



variables

$\Box((Number\ of\ FCTs) \leq 7)$

count

compare

- Included metrics
  - AST height (ASTH)
  - # atomic propositions (APs)
  - # comparison operators (COPs)
  - # logical operators (LOPs)
  - # temporal operators (TOPs)
  - Shannon entropy $H(v) = -\sum_i p_i \log_2(p_i)$

Methodology

III. Quantitative Analysis – RQ2

What is the engineering complexity of the natural formulae for a transcribed SpaceWire requirement, and does it differ among the logics?

- Included metrics
  - ASTH ⇒ 5
  - # APs ⇒ { y, u_eq_9, i_gt_3} = 3
  - # COPs ⇒ {=, <} = 2
  - # LOPs ⇒ {∧, ∨, →, ¬} = 4
  - # TOPs ⇒ {□, ◊} = 2
  - Shannon entropy ≈ 2.585

$$H(v) = -\sum_i p_i \log_2(p_i)$$

- Example

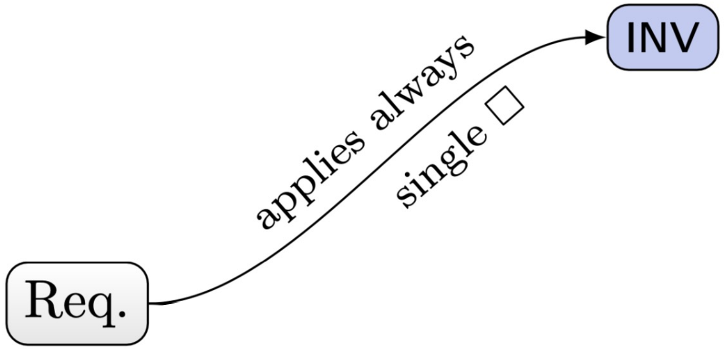
$$\Box (y \land (u = 9) \rightarrow \Diamond (\neg y \lor i < 3))$$

Results

Results

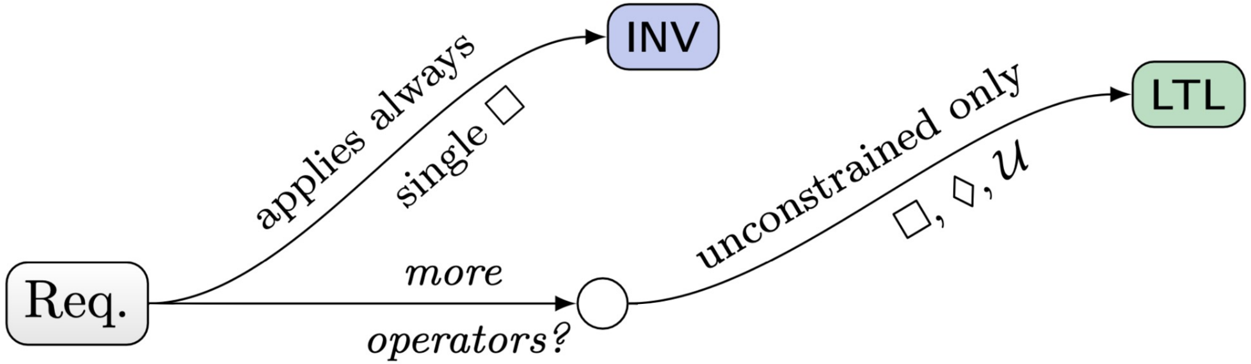
Formalized Requirements

Ref	[Requirement ID] Requirement Text	Operators	Logic	Formalization
R1	[1006] Null detection shall be enabled **whenever** the receiver is enabled.	□	INV	$\Box((\mathit{receiver\ enabled}) \rightarrow (\mathit{Null\ detection\ enabled}))$



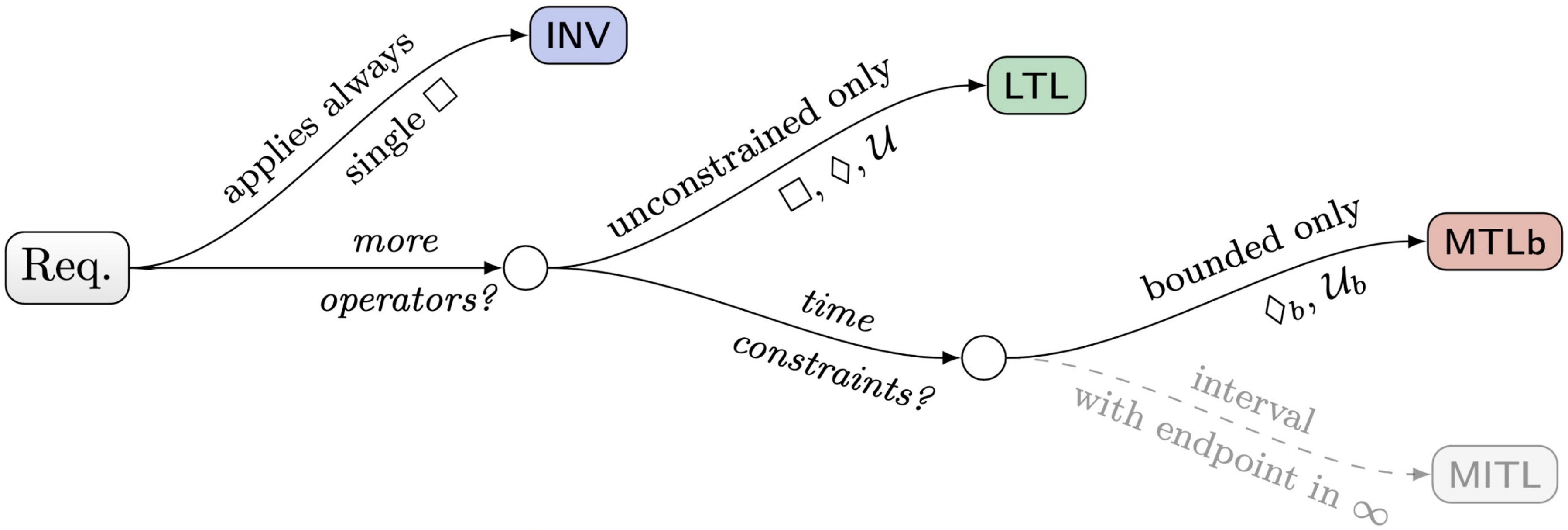
Formalized Requirements

Ref	[Requirement ID] Requirement Text	Operators	Logic	Formalization
R3	[2013] **When** the link is initialised or re-initialised, one FCT shall be sent for **every** eight N-Chars that can be held in the receive FIFO up to the maximum of seven FCTs.	$\Box, \mathcal{X}, \mathcal{U}$	LTL	$\Box((\mathit{link\ state} : (\mathit{initialised} \lor \mathit{reinitialised})) \rightarrow (((8\ \mathit{NChar\ held}) \rightarrow \mathcal{X}(\mathit{one\ FCT\ sent}))\ \mathcal{U}\ (\mathit{Num\ sent\ FCT} \leq 7)))$



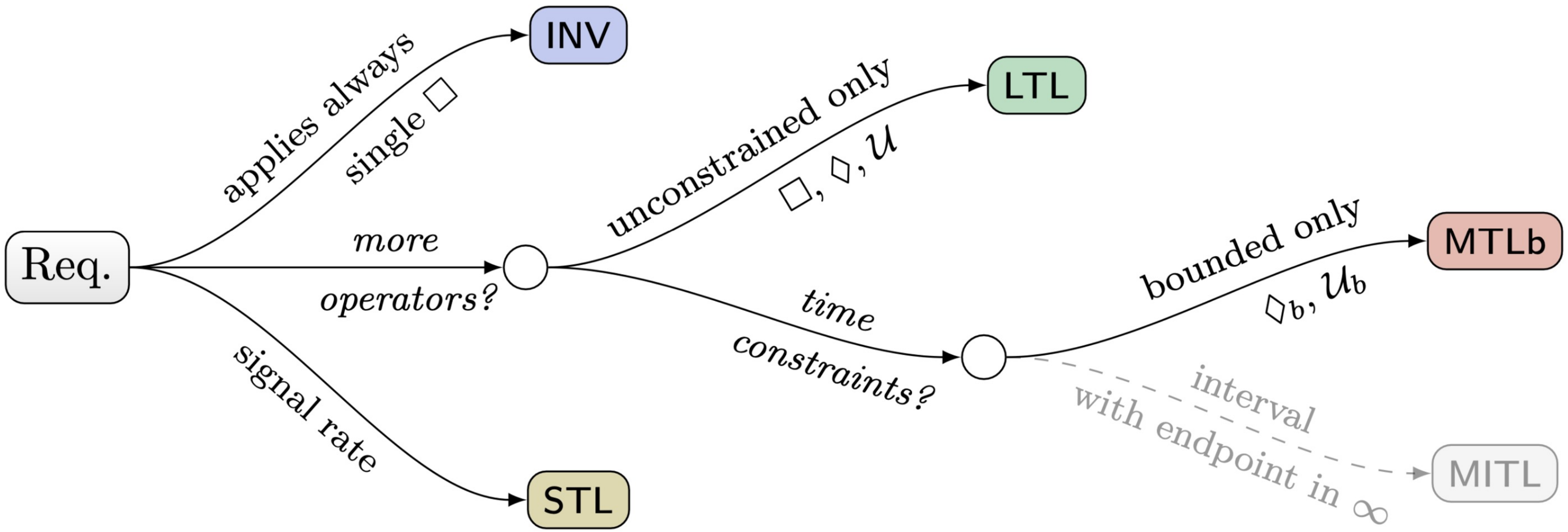
Formalized Requirements

Ref	[Requirement ID] Requirement Text	Operators	Logic	Formalization
R6	[3014] The **delay between** the interrupt code arriving and the interrupt acknowledgement being generated shall be less than the maximum time determined for a node to generate an interrupt acknowledgement code.	$\Box, \Diamond_I,$ $I = interval$	MTLb	$\Box((interrupt\ code\ arriving) \rightarrow$ $\Diamond_{(0,t)}(interrupt\ ack)\ generated),$ $t \leq max\ interrupt\ ack\ time$



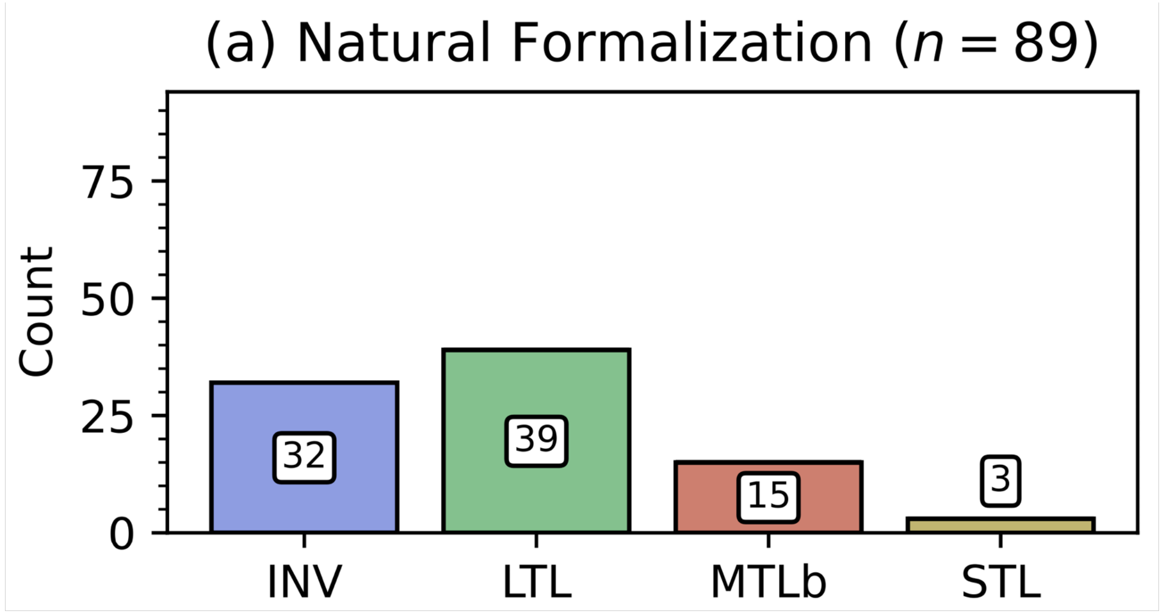
Formalized Requirements

Ref	[Requirement ID] Requirement Text	Operators	Logic	Formalization
R9	[4002] The SpaceWire output port shall operate at 10 ±1 Mb/s **until** set to operate at a different data signaling rate.	$\Box, \mathcal{U}$	STL	$\Box((9 \leq S_{data}(t) \leq 11)$ $\mathcal{U}$ $(\mathit{set\ different\ rate}))$



RQ1: Distribution and Mutual Translatability

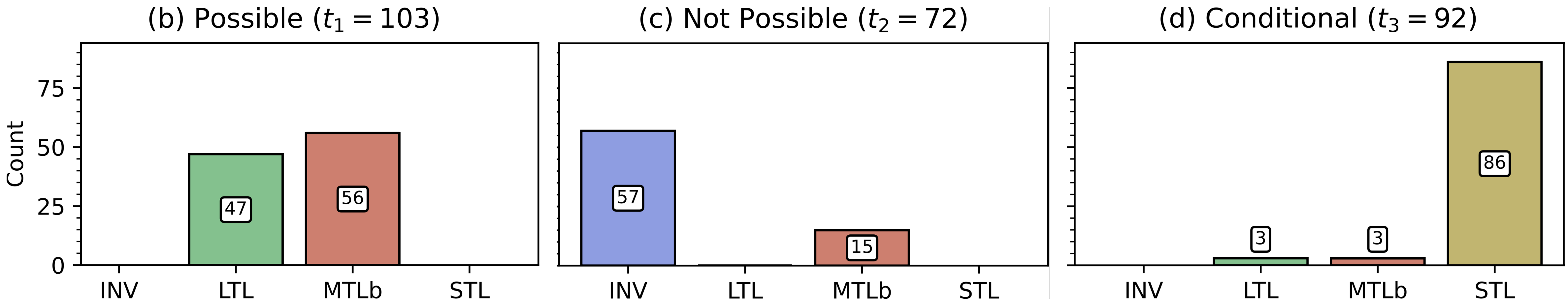
Part 1 / 2



Results

RQ1: Distribution and Mutual Translatability

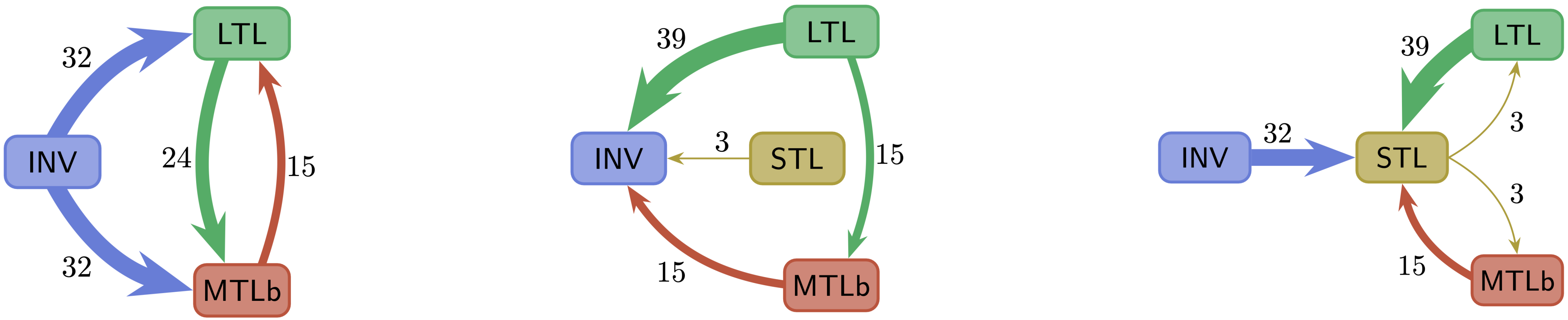
Part 2 / 2



(b) Possible ($t_1 = 103$)

(c) Not Possible ($t_2 = 72$)

(d) Conditional ($t_3 = 92$)



(a) Possible

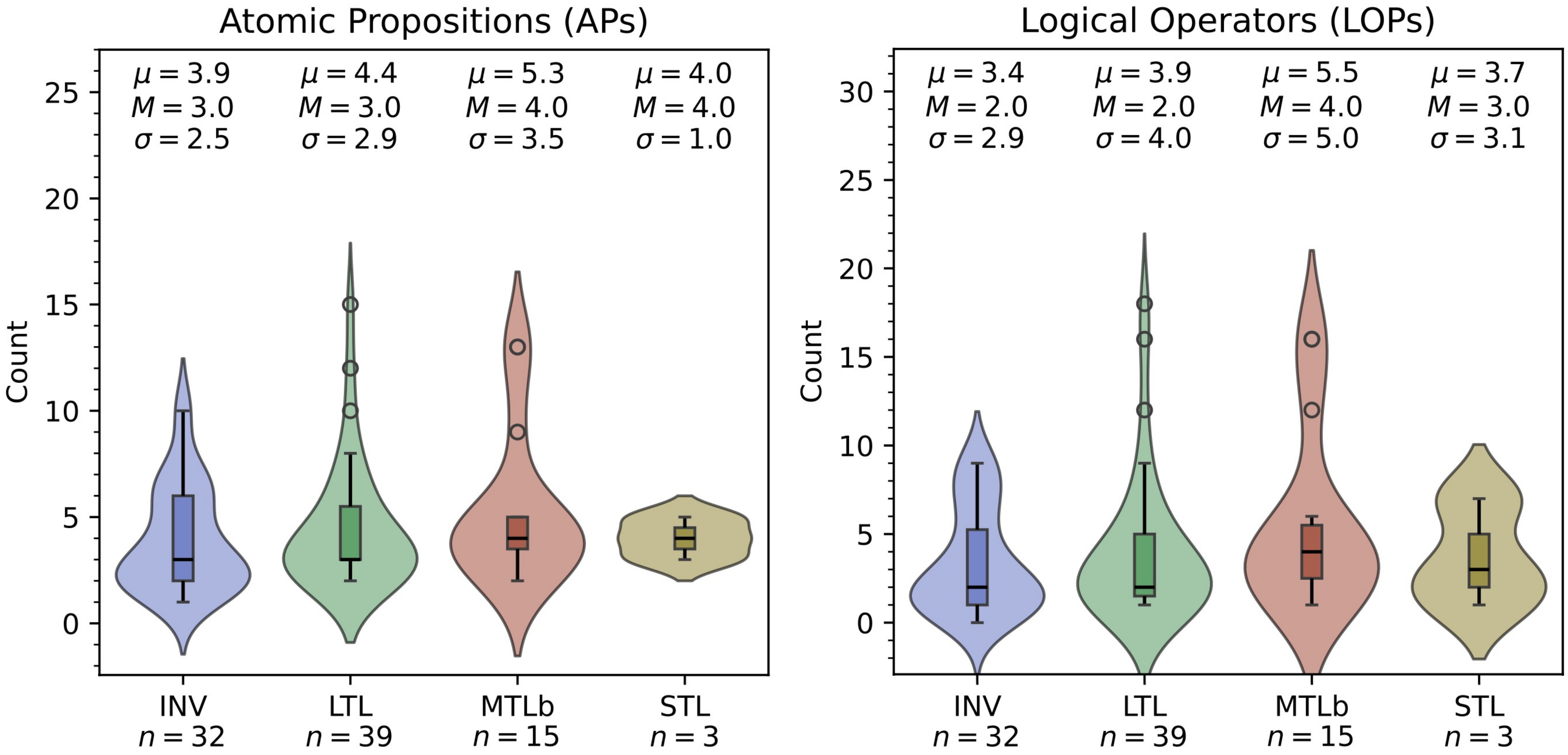
(b) Not Possible

(c) Conditionally Possible

Results

RQ2: Engineering Complexity

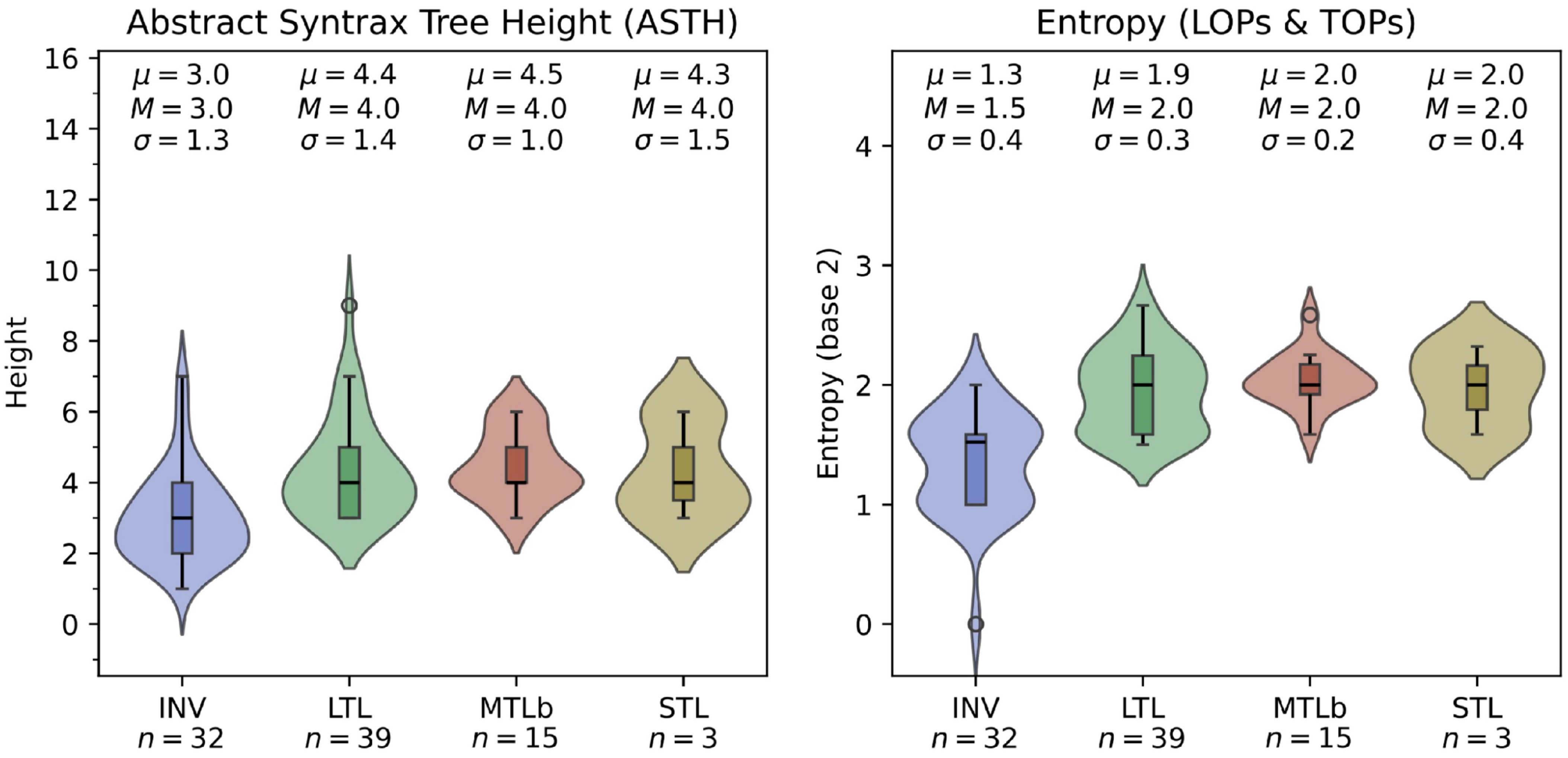
Part 1 / 2



Results

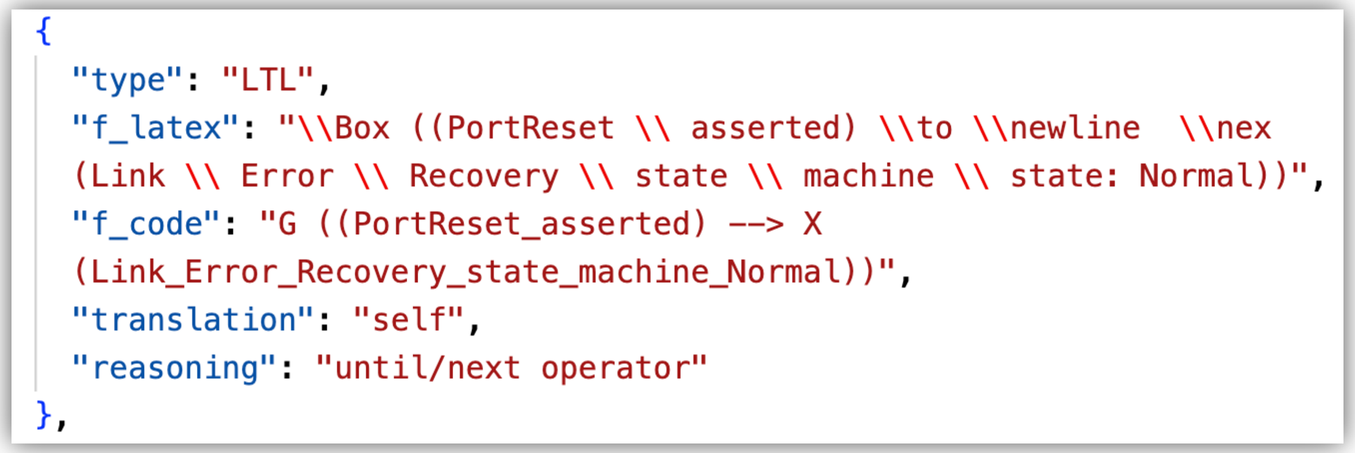
RQ2: Engineering Complexity

Part 2 / 2

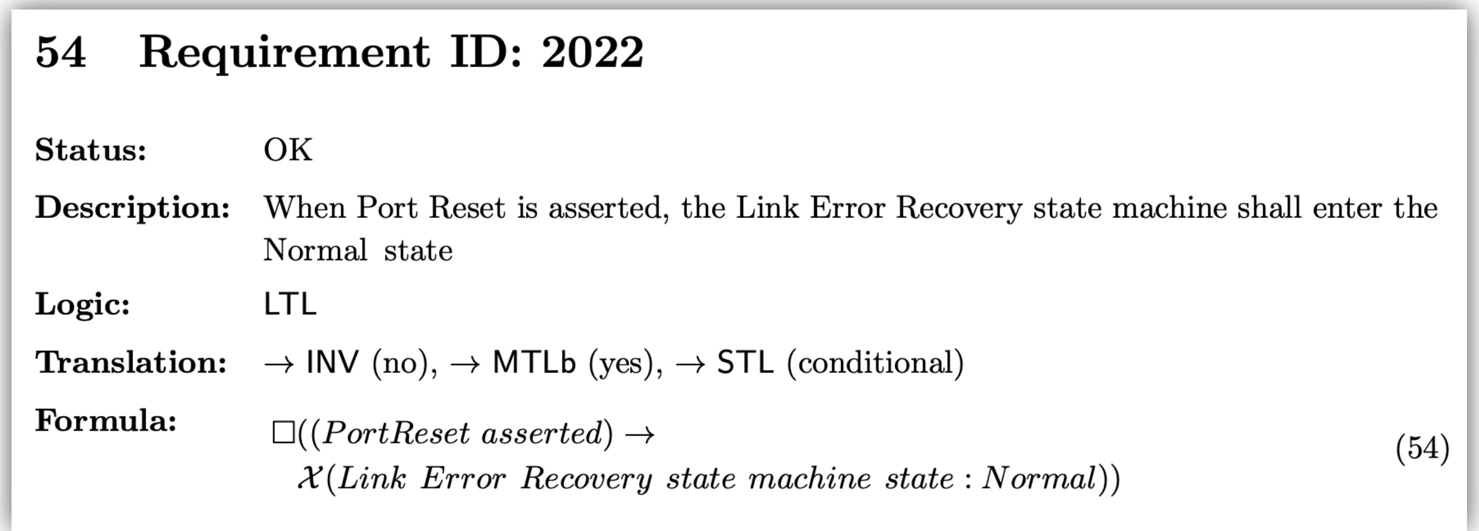


Results

Tool Support and Dataset



```
{
  "type": "LTL",
  "f_latex": "\\Box ((PortReset \\ asserted) \\to \\newline  \\nex
  (Link \\ Error \\ Recovery \\ state \\ machine \\ state: Normal))",
  "f_code": "G ((PortReset_asserted) --> X
  (Link_Error_Recovery_state_machine_Normal))",
  "translation": "self",
  "reasoning": "until/next operator"
},
```



54 Requirement ID: 2022

Status: OK

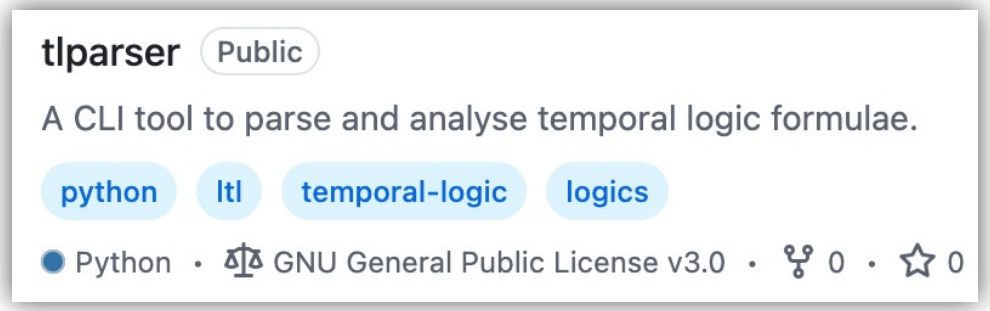
Description: When Port Reset is asserted, the Link Error Recovery state machine shall enter the Normal state

Logic: LTL

Translation: → INV (no), → MTLb (yes), → STL (conditional)

Formula:

$$\Box((PortReset\ asserted) \rightarrow \mathcal{X}(Link\ Error\ Recovery\ state\ machine\ state : Normal)) \tag{54}$$



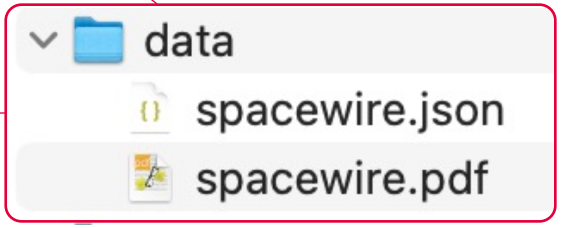
tlparser Public

A CLI tool to parse and analyse temporal logic formulae.

python ltl temporal-logic logics

● Python · GNU General Public License v3.0 · 0 · ☆ 0

https://zenodo.org/records/14810693



Discussion

Discussion

Potential Implication

- Practitioners
  - Substantial amount of invariants
  - Jungle of tool support
  - Engineering complexity enables fingerprinting

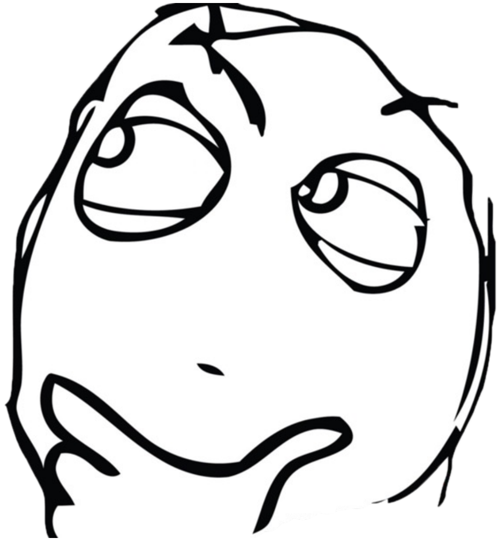
- Researchers
  - Observed Pareto principle
  - Specialized unified subsets of existing logics

Discussion

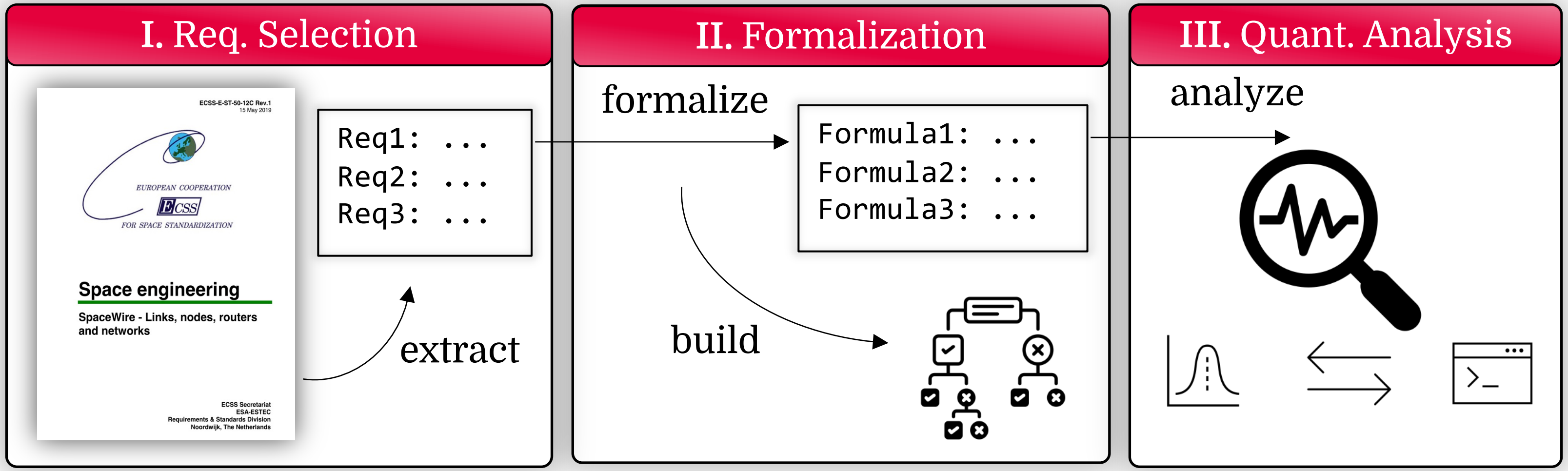
Future Work

- Theoretic aspects
  - Investigating monitorability
  - Extend to other requirement documents

- Engineering aspects
  - Extend notion of fingerprint
  - Interface with other tools and DSLs
  - Leverage dataset for GPT models



Summary



I. Req. Selection

ECSS-E-ST-50-12C Rev.1
15 May 2019

EUROPEAN COOPERATION
ECSS
FOR SPACE STANDARDIZATION

Space engineering

SpaceWire - Links, nodes, routers and networks

ECSS Secretariat
ESA-ESTEC
Requirements & Standards Division
Noordwijk, The Netherlands

```
Req1: ...
Req2: ...
Req3: ...
```

extract

II. Formalization

formalize

```
Formula1: ...
Formula2: ...
Formula3: ...
```

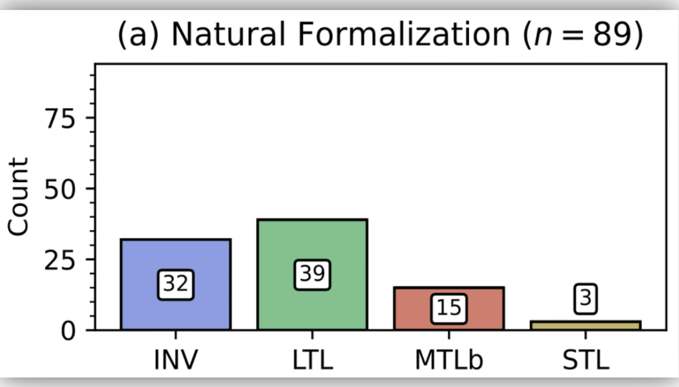
build

III. Quant. Analysis

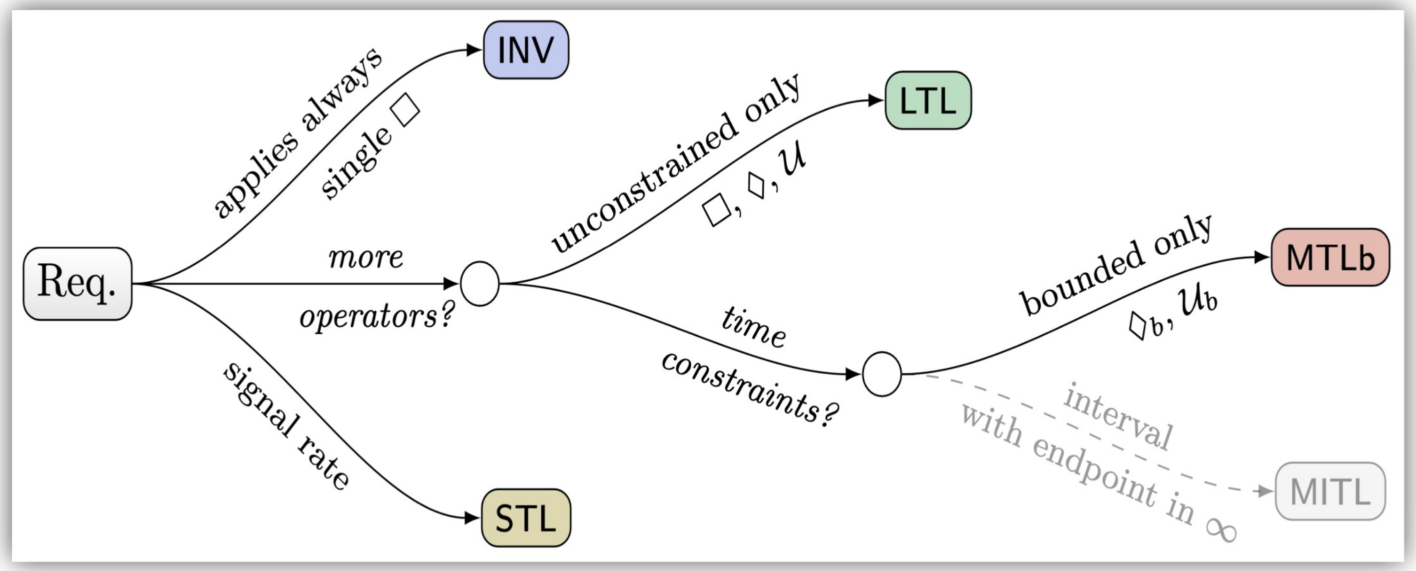
analyze

tlparser Public

A CLI tool to parse and analyse temporal logic formulae.



(a) Natural Formalization ($n = 89$)



Req.

applies always

single $\Box$

INV

more operators?

unconstrained only

$\Box, \Diamond, \mathcal{U}$

LTL

time constraints?

bounded only

$\Diamond_b, \mathcal{U}_b$

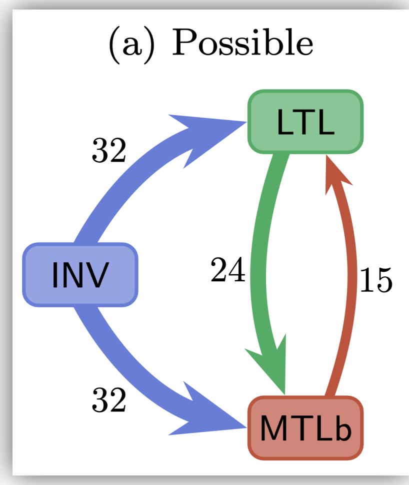
MTLb

interval with endpoint in ∞

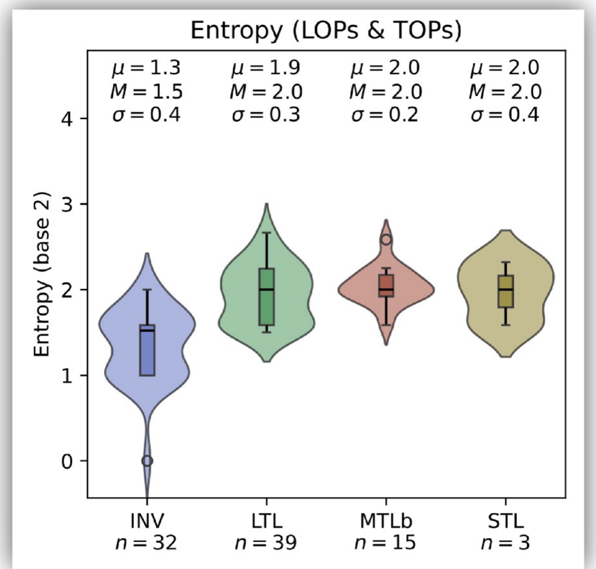
MITL

signal rate

STL



(a) Possible



Entropy (LOPs & TOPs)

Thank you

Happy to chat: **roman.boegli@unibe.ch**

Appendix

Appendix

Atomic Propositions

- Pragmatic approach
  - striving finest granularity possible
  - while maintaining the necessary level of coarseness

- Example
  - “The gotNull.indication primitive shall be passed to the data link layer, when the first Null is received without any errors after the receiver has been enabled.”

$$\Box(\mathit{receiverEnabled} \rightarrow \mathcal{X}(\Box(\mathit{firstNullReceivedWithoutError})) \rightarrow (\mathit{gotNullPassed}))$$

separating into **(firstNullReceived) ∧ (¬ error)** introduces a problem:

if the if the first ‚Null‘ is received with an error, ‚firstNullReceived‘ would never hold again as subsequent nulls would no longer be the first one (i.e. unsatisfiable)

Appendix

Threats to Validity

- Internal
  - Subjectivity in natural formalization
  - AP granularity

- Explicit declared pragmatism
- Systematic of decision tree

- External
  - Single case (SpaceWire)

- Applicability of oerall methodology remains
- Tool support (tlparser)

- Construct
  - Engineering complexity ignores semantic, algorithm complexity, or a system’s broader context.

- Practical value for problem-oriented approaches