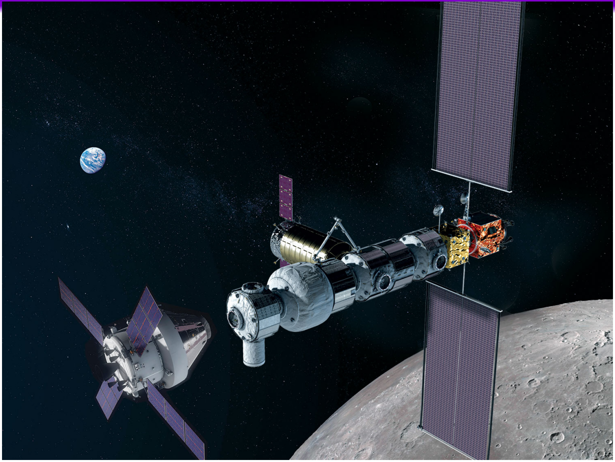
MLTL Multi-type (MLTLM): A Logic for Reasoning About Signals of Different Types

Gokul Hariharan, Brian Kempa,
Tichakorn Wongpiromsarn, Phillip H. Jones, **Kristin Y. Rozier**
Iowa State University

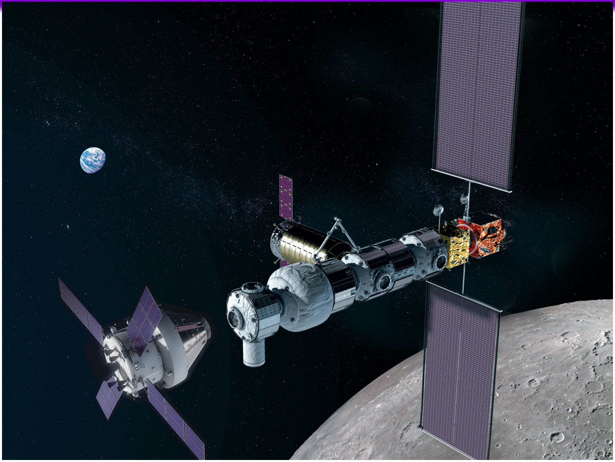
Numerical Software Verification (NSV)
August 11, 2022

NASA Lunar Gateway: Assume-Guarantee Contracts[1]



[1] Dabney, James B., Julia M. Badger, and Pavan Rajagopal. "Adding a Verification View for an Autonomous Real-Time System Architecture." In AIAA Scitech 2021 Forum, p. 0566. 2021.

NASA Lunar Gateway: Assume-Guarantee Contracts[1]



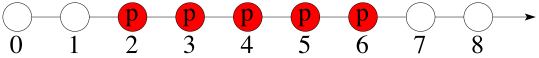
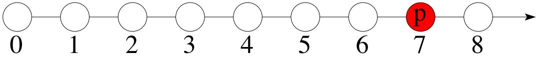
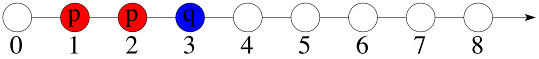
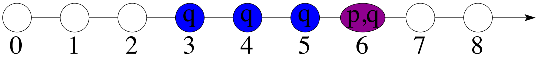
$$(CMD == START) \rightarrow (\Box_{[0,5]}(ActionHappens \& \Box_{[0,2]}(CMD = END)))$$

[1] Dabney, James B., Julia M. Badger, and Pavan Rajagopal. "Adding a Verification View for an Autonomous Real-Time System Architecture." In AIAA Scitech 2021 Forum, p. 0566. 2021.

Encoding Finite Timelines

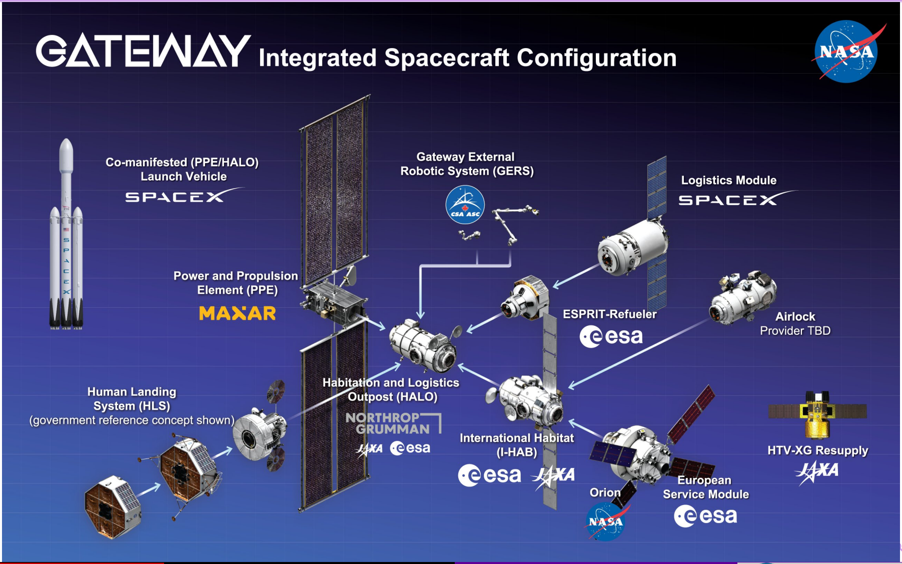
Mission-time LTL (MLTL) reasons about *bounded* timelines:

- finite set of atomic propositions {p q}
- Boolean connectives: ¬, ∧, ∨, and →
- temporal connectives *with time bounds*:

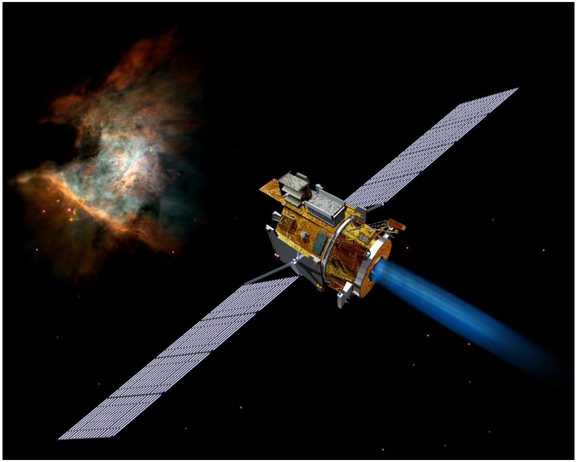
Symbol	Operator	Timeline
$\Box_{[2,6]} p$	ALWAYS$_{[2,6]}$	
$\Diamond_{[0,7]} p$	EVENTUALLY$_{[0,7]}$	
$p\,\mathcal{U}_{[1,5]}\,q$	UNTIL$_{[1,5]}$	
$p\,\mathcal{R}_{[3,8]}\,q$	RELEASE$_{[3,8]}$	

Mission-bounded LTL is an over-approximation for mission time τ

NASA Lunar Gateway: A System of (Mix-Typed) Systems!


GATEWAY Integrated Spacecraft Configuration
NASA
Co-manifested (PPE/HALO) Launch Vehicle
SPACEX
Gateway External Robotic System (GERS)
CSA ASC
Logistics Module
SPACEX
Power and Propulsion Element (PPE)
MAXAR
ESPRIT-Refueler
esa
Airlock Provider TBD
Human Landing System (HLS) (government reference concept shown)
Habitation and Logistics Outpost (HALO)
NORTHROP GRUMMAN
JAXA esa
International Habitat (I-HAB)
esa JAXA
HTV-XG Resupply
JAXA
Orion
NASA
European Service Module
esa

A Typical Deep Space Mission



- *monthly* course corrections
- *nanosecond* precise sensor adjustments
- *system-level* requirements

The Question

Existing logics reason over signals of the *same type*:

$\pi = \{\sigma_0, \cdots, \sigma_n\}$ is a set of signals populating $p_0, \cdots, p_n \in \mathcal{AP}$

The Question

Existing logics reason over signals of the *same type*:

$\pi = \{\sigma_0, \cdots, \sigma_n\}$ is a set of signals populating $p_0, \cdots, p_n \in \mathcal{AP}$

What happens when signals have different types?

We Need A Logic For This!

Like MLTL but . . .

- includes type conversions different types in one formula

[2] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

We Need A Logic For This!

Like MLTL but . . .

- includes type conversions different types in one formula
- allows for changing type conversions like MLTL allows for changing time intervals

[2] K.Y.Rozier. “Specification: The Biggest Bottleneck in Formal Methods and Autonomy.” VSTTE, 2016.

We Need A Logic For This!

Like MLTL but . . .

- includes type conversions different types in one formula
- allows for changing type conversions like MLTL allows for changing time intervals
- cleanly separates type conversions from logic, enabling efficient analysis

[2] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

We Need A Logic For This!

Like MLTL but . . .

- includes type conversions different types in one formula
- allows for changing type conversions like MLTL allows for changing time intervals
- cleanly separates type conversions from logic, enabling efficient analysis
- converts to MLTL or similar to reduce complexity

[2] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

We Need A Logic For This!

Like MLTL but . . .

- includes type conversions different types in one formula
- allows for changing type conversions like MLTL allows for changing time intervals
- cleanly separates type conversions from logic, enabling efficient analysis
- converts to MLTL or similar to reduce complexity
- still easy to validate [2]

[2] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

We Need A Logic For This!

Like MLTL but . . .

- includes type conversions different types in one formula
- allows for changing type conversions like MLTL allows for changing time intervals
- cleanly separates type conversions from logic, enabling efficient analysis
- converts to MLTL or similar to reduce complexity
- still easy to validate [2]

We need MLTL *for mixed types*

[2] K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

Why Don't We Use (Existing) More Expressive Logic?

- **modularity**: clean separation of type conversions from MLTL structure

Why Don't We Use (Existing) More Expressive Logic?

- **modularity**: clean separation of type conversions from MLTL structure
- **complexity**: fit in limited resources of embedded systems

Why Don't We Use (Existing) More Expressive Logic?

- **modularity**: clean separation of type conversions from MLTL structure
- **complexity**: fit in limited resources of embedded systems
- **validation**: use the right tool for the job, not a kludge

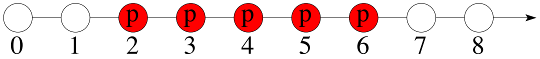
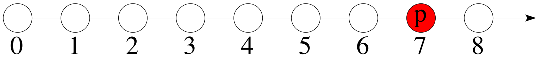
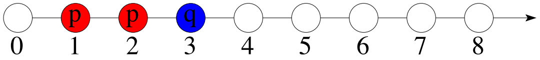
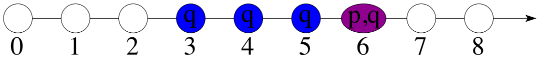
Why Don't We Use (Existing) More Expressive Logic?

- **modularity**: clean separation of type conversions from MLTL structure
- **complexity**: fit in limited resources of embedded systems
- **validation**: use the right tool for the job, not a kludge
- **extensibility**: retain type conversions to enable optimization
  - store more information in one formula

Encoding Finite Trajectories Over Signals of Mixed Types

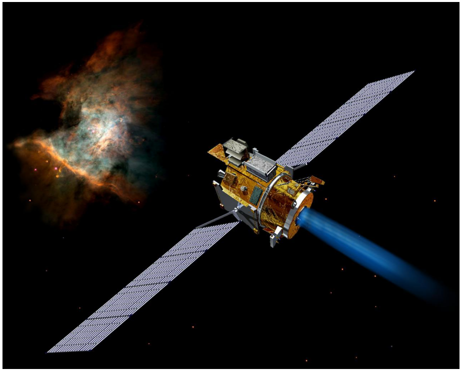
MLTL Multi-type (MLTLM) reasons about formulas over signals of *mixed types*:

- finite set of atomic propositions {p q}
- Boolean connectives: $\neg$, $\wedge$, $\vee$, and $\rightarrow$
- temporal connectives *with time bounds*:

Symbol	Operator	Timeline
$\Box_{[2,6,\mathbb{A}]}p$	ALWAYS$_{[2,6,\mathbb{A}]}$	
$\Diamond_{[0,7,\mathbb{A}]}p$	EVENTUALLY$_{[0,7,\mathbb{A}]}$	
$p\mathcal{U}_{[1,5,\mathbb{A}]}q$	UNTIL$_{[1,5,\mathbb{A}]}$	
$p\mathcal{R}_{[3,8,\mathbb{A}]}q$	RELEASE$_{[3,8,\mathbb{A}]}$	

To resolve the formula, need a *projection*: $T^{\mathbb{B}}_{\mathbb{A}}(\sigma^{\mathbb{A}}) = \sigma^{\mathbb{B}}$

Example: Deep Space Mission



The spacecraft maintenance cycle runs at least once a month over the five-year mission.

Monthly course corrections never involve burning the thrusters more than 3 seconds at a time.

$$\Box_{[0,5,year]}[(\Diamond_{[0,30,day]} maintenance) \wedge (\neg\Box_{[0,3,sec]} thrusters)]$$

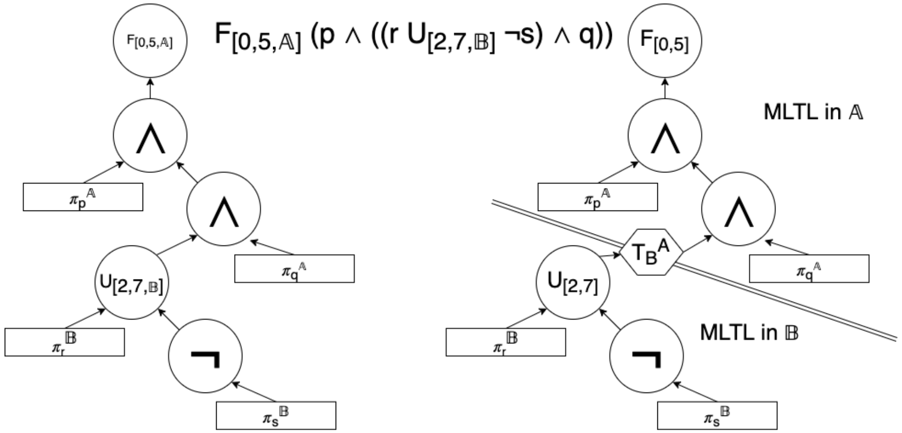
Relationship Between MLTL and MLTLM

Logical projection: a projection that can be expressed in MLTL

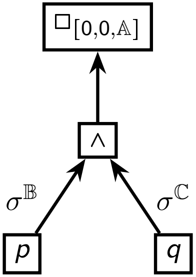
MLTLM with all logical projections is equivalent to MLTL

MLTLM ↔ MLTL

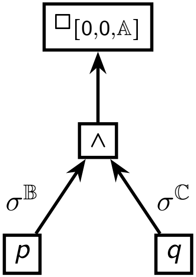
$F_{[0,5,\mathbb{A}]}(p \wedge ((r\ U_{[2,7,\mathbb{B}]} \neg s) \wedge q))$

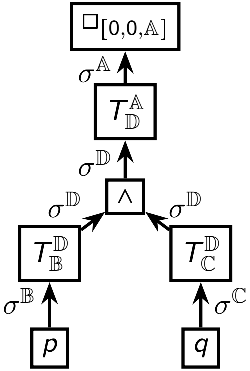


Projection Options & Implementation Patterns

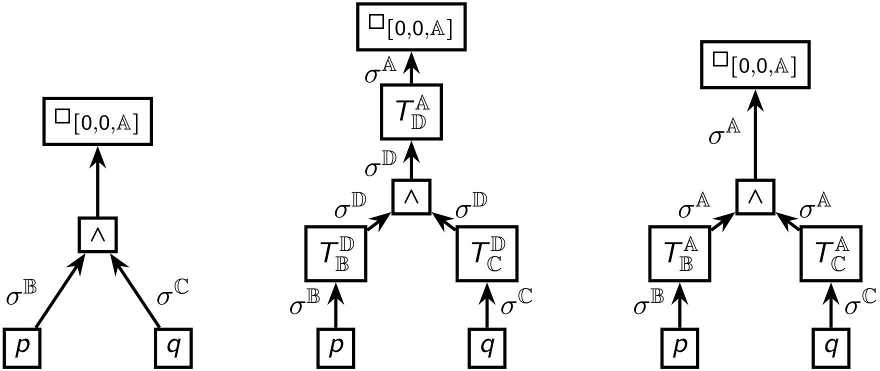


Projection Options & Implementation Patterns

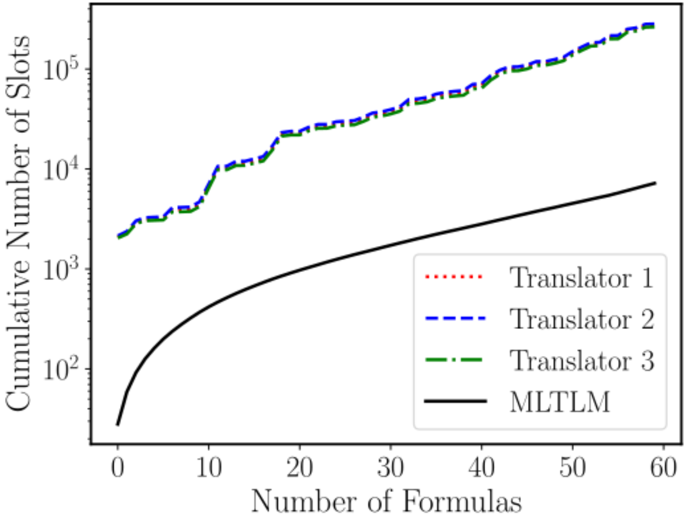




Projection Options & Implementation Patterns

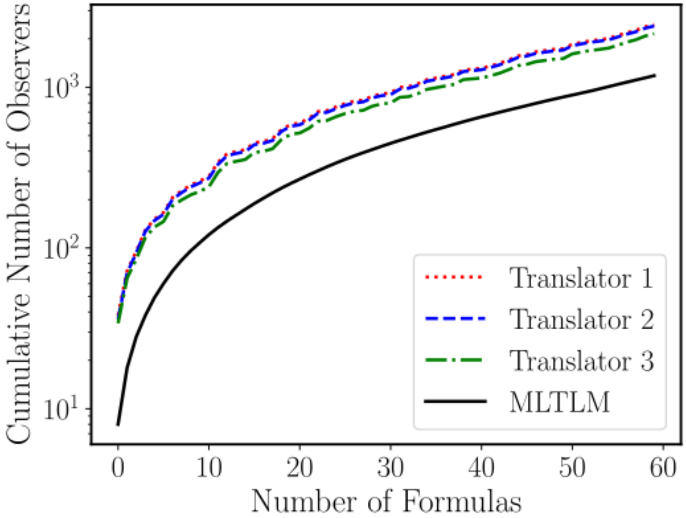


Direct Analysis of MLTLM Reduces Memory*



* versus translation to MLTL

Direct Analysis of MLTLM Reduces Time*



* versus translation to MLTL

Summary

- 3 translation algorithms: MLTLM w/logical projections $\rightarrow$ MLTL
- MLTLM RV algorithm & open-source implementation
- Direct MLTLM analysis saves space and time
- Preserve formula validation and modularity!

Mix your types with MLTLM!

http://temporallogic.org/research/NSV2022