From English Assume-Guarantee Contracts to Validated Temporal Logic Specifications

Kristin Yvonne Rozier
Iowa State University

SCHLOSS DAGSTUHL
Leibniz-Zentrum für Informatik

November 9, 2022

Projects: Formula Validation for Specification (by Humans)

A NASA Rocket Scientist was given this English specification:

"p oscillates every time step"

She wrote four possible LTL formulas:

$$\phi_1 = (\text{ALWAYS } ((p \vee \text{ NEXT } \neg p) \vee (\neg p \vee \text{NEXT} p)))$$
$$\phi_2 = (\text{ALWAYS } ((p \vee \text{ NEXT } \neg p) \wedge (\neg p \vee \text{NEXT} p)))$$
$$\phi_3 = (\text{ALWAYS } ((p \wedge \text{ NEXT } \neg p) \vee (\neg p \wedge \text{NEXT} p)))$$
$$\phi_4 = (\text{ALWAYS } ((p \wedge \text{ NEXT } \neg p) \wedge (\neg p \wedge \text{NEXT} p)))$$

LTL: Truth Table Validation Example

"p oscillates every time step"

How do we convincingly demonstrate which LTL formula is right? Here's one way:

$\phi_1 = (\text{ALWAYS } ((p \vee \text{ NEXT } \neg p) \vee (\neg p \vee \text{NEXT} p)))$
$\phi_2 = (\text{ALWAYS } ((p \vee \text{ NEXT } \neg p) \wedge (\neg p \vee \text{NEXT} p)))$
$\phi_3 = (\text{ALWAYS } ((p \wedge \text{ NEXT } \neg p) \vee (\neg p \wedge \text{NEXT} p)))$
$\phi_4 = (\text{ALWAYS } ((p \wedge \text{ NEXT } \neg p) \wedge (\neg p \wedge \text{NEXT} p)))$

p	$\mathcal{X}p$	$\neg p$	$\neg\mathcal{X}p$	$p \wedge \mathcal{X}\neg p$	$\neg p \wedge \mathcal{X}p$	ϕ_1	ϕ_2	ϕ_3	ϕ_4
T	T	F	F	F	F	T	T	F	F
T	F	F	T	T	F	T	F	T	F
F	T	T	F	F	T	T	F	T	F
F	F	T	T	F	F	T	T	F	F

Assume-Guarantee Contracts[1]

Contract Template Format

C_1:

- variables:
  - inputs: *values supplied to function*
  - outputs: *values returned from function*
- types: *data types (can be custom) of each input and output*
- assumptions: *conditions and constraints on inputs under which contract is valid*
- guarantees: conditions on outputs when contract is valid

[1] Dabney, Rajagopal, Badger. FSW 2022: Using Assume-Guarantee Contracts for Developmental Verification of Autonomous Spacecraft.

Generating Contracts: From Textual Requirements to MLTL[2]

1. When Executive receives a task command, Executive shall respond with accept/reject within 5 seconds
2. Executive shall provide task updates at 0.1 Hz
3. After accepting command, Executive shall respond with completion message within 10 seconds

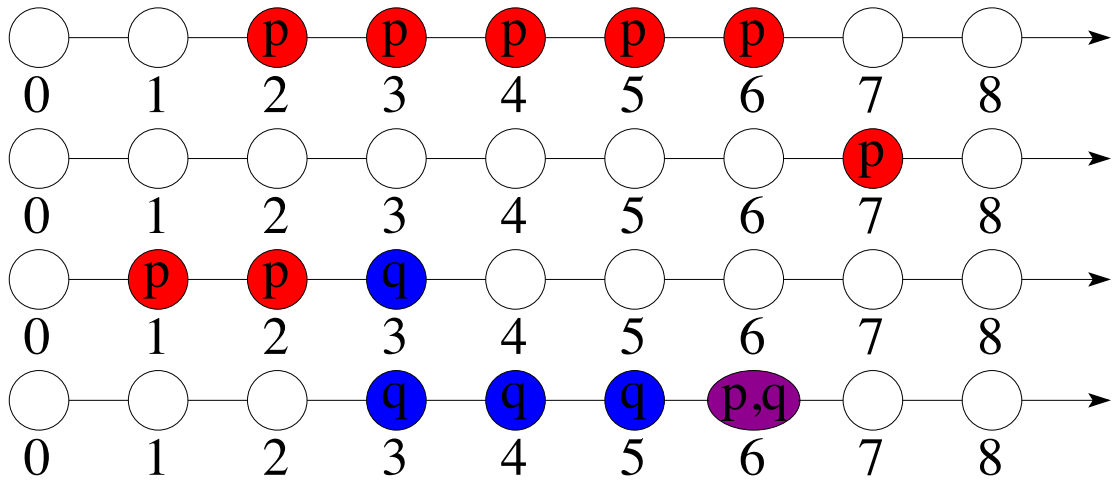
[2] FSW 2021: Using Assume-Guarantee Contracts In Autonomous Spacecraft - James Bruster Dabney

Encoding Finite Timelines

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

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

Symbol	Operator	Timeline
$\Box_{[2,6]} p$	$\textsc{Always}_{[2,6]}$	
$\Diamond_{[0,7]} p$	$\textsc{Eventually}_{[0,7]}$	
$p \mathcal{U}_{[1,5]} q$	$\textsc{Until}_{[1,5]}$	
$p \mathcal{R}_{[3,8]} q$	$\textsc{Release}_{[3,8]}$	



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

NASA Lunar Gateway: Assume-Guarantee Contracts[3]

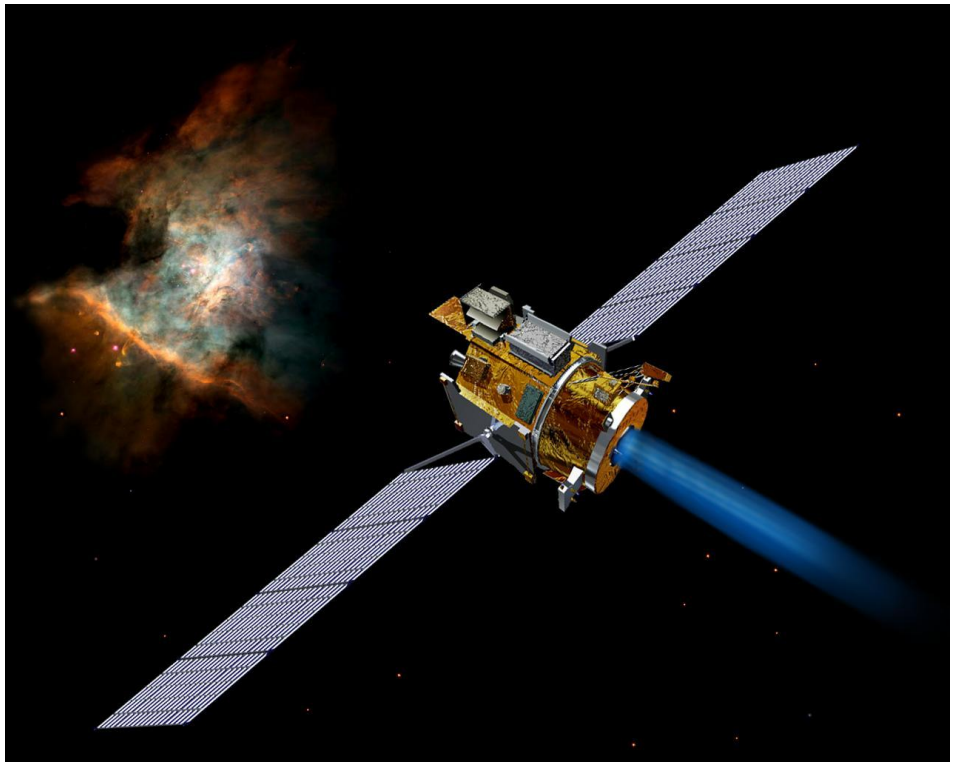
[3] Dabney, James B., Julia M. Badger, and Pavan Rajagopal. "Adding a Verification View for an Autonomous Real-Time System Architecture." In AIAA Scitech 2021 Forum, p. 0566. 2021.

NASA Lunar Gateway: Assume-Guarantee Contracts[3]

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

[3] 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.

MLTLM Example: Deep Space Mission [4]



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.

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

[4] Hariharan, Kempa, Wongpiromsarn, Jones, Rozier. NSV 2022.

Some Challenges

- Specification elicitation **methods** & **structures**
  - See Group 4 report . . .
- **More expressive** logics that are **not harder to validate**
- Automated **validation**
  - For LTL, MLTL, MLTLM, LTLf, . . .

BACKUP SLIDES