



INTELLIGENT
SYSTEMS
DIVISION



Capture, Analyze, Diagnose: Realizability Checking of Requirements in FRET

Andreas Katis¹, Anastasia Mavridou¹, Dimitra
Giannakopoulou², Thomas Pressburger², Johann Schumann¹

¹ Employed by KBR; NASA Ames Research Center, CA, USA

² NASA Ames Research Center, CA, USA

Realizability Checking in FRET (Live Demo)

captures + assists *explains*

FRETish

System Component *

FSM ☒ Compositional ☐ Monolithic

Timeout (seconds) 900 CHECK DIAGNOSE HELP

CC0 CC1 CC2

Counterexample for Conflict 3 SIMULATE

Variable name	Variable type	Step 0	Step 1	Step 2	Step 3	Step 4	Step 5
good	bool	true	true	true	true	true	
standby	bool	false	false	false	false	false	
state	real	2	2	2	2	2	
supported	bool	true	true	true	true	true	
STATE	real	1	4	5	6	7	

ID ↑	Summary
FSM006	FSM shall for 5 ticks satisfy (state = ap_maneuver_state & standby & good) => STATE = ap_standby_state
FSM007	FSM shall within 5 ticks satisfy (state = ap_maneuver_state & supported & good) => STATE = ap_transition_state
FSM002	FSM shall always satisfy (standby & state = ap_transition_state) => STATE = ap_standby_state

FSM007 FSM shall within 5 ticks satisfy (state = ap_maneuver_state & supported & good) => STATE = ap_transition_state

FSM002 FSM shall always satisfy (standby & state = ap_transition_state) => STATE = ap_standby_state

Realizability Checking

[FSM-006] *“The autopilot shall change **states** from MANEUVER to STANDBY when the pilot is in control (**standby**) and sensor data is **good** and stay in this state for the next 5 ticks.”*

[FSM-007] *“The autopilot shall, within the next 5 ticks, change **states** from MANEUVER to TRANSITION when the system is **supported** and sensor data is **good**.”*

- **Reactive Systems:** *Inputs* controlled by the environment; *outputs* controlled by system.

Inputs:

standby, supported, good : Bool

Outputs:

STATE : {MANEUVER, STANDBY, TRANSITION}

- Formal Analysis

Consistency: Are the requirements satisfiable?

Realizability: Are the requirements satisfiable for any input provided by the environment?

Realizability in Requirements Specification Tools

Tools	Finite State	Infinite State	Decomposition	Liveness	Unrealizable Cores
Spectra	✓	✗	✗	✓	✓
SpeAR	✓	✓	✗	✗	✗
AGREE	✓	✓	✗	✗	✗
RATSY	✓	✗	✗	✓	✓
EARS-CTRL	✓	✗	✗	✗	✗
FRET	✓	✓	✓	✗	✓

<https://github.com/NASA-SW-VnV/fret>

Tools	Finite State	Infinite State	Decomposition	Liveness	Unrealizable Cores	Algorithms	Backend	Other features
Spectra	✓	✗	✗	✓	✓	BDD-based fixpoint	CUDD + JTLV	Well-separation, Vacuity Checking, Counterstrategies
SpeaAR	✓	✓	✗	✗	✗	<i>k</i> -induction	JKind	N/A
AGREE	✓	✓	✗	✗	✗	<i>k</i> -induction	JKind	N/A
RATSY	✓	✗	✗	✓	✓	BDD-based fixpoint	CUDD + NuSMV	Counterstrategies
EARS-CTRL	✓	✗	✗	✗	✗	BDD-based fixpoint	autoCode4	N/A
FRET	✓	✓	✓	✗	✓	<i>k</i> -induction, SMT-based fixpoint	JKind, Kind2	Simulation of conflicting requirements