LTL semantics for Runtime Verification

Martin Leucker

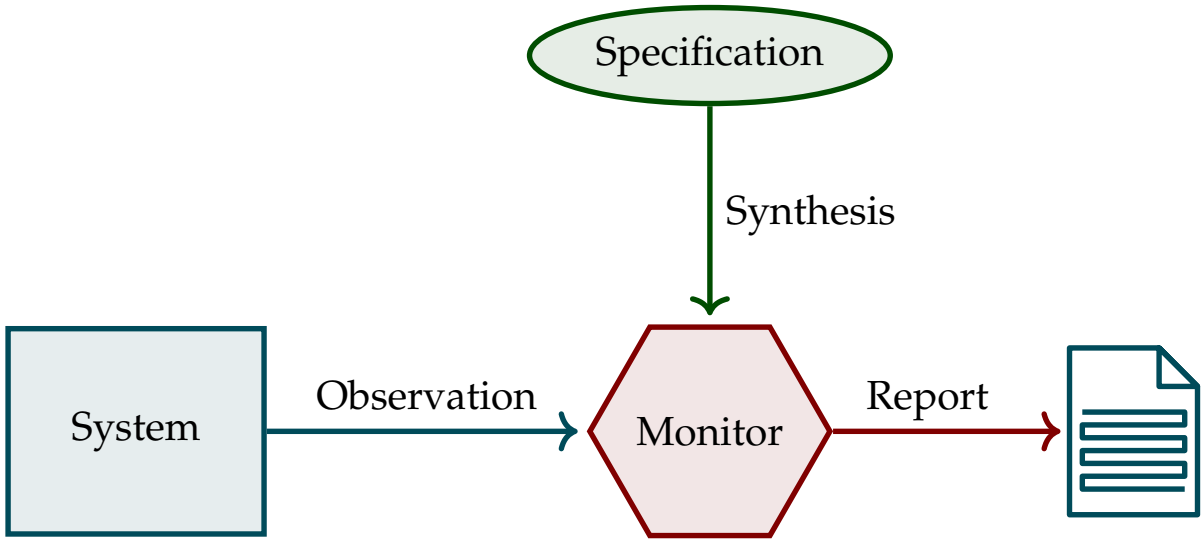
Based on

Andreas Bauer, Martin Leucker, Christian Schallhart,

Comparing LTL Semantics for Runtime Verification.

J. Log. Comput. 20(3): 651-674 (2010)

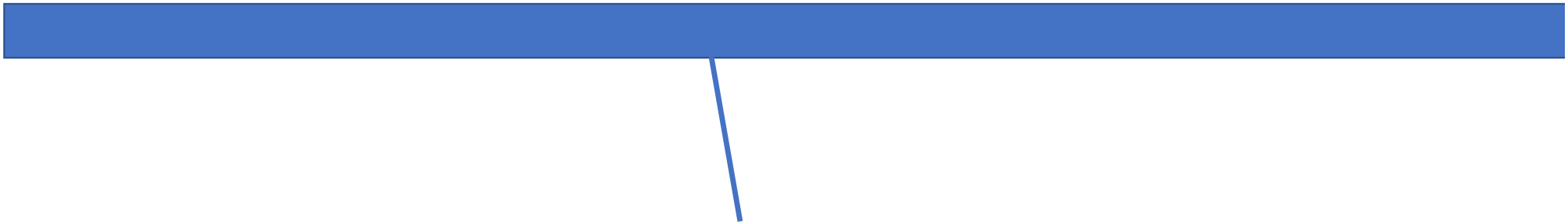
Runtime Verification



- Partial Verification
- Testing Temporal Assertions
- Test Cases as Input Sequences checked by Monitors
- Debugging
- Control?

The Ideas

Specification of Traces: LTL



Finite Trace - FLTL

Say yes or no!

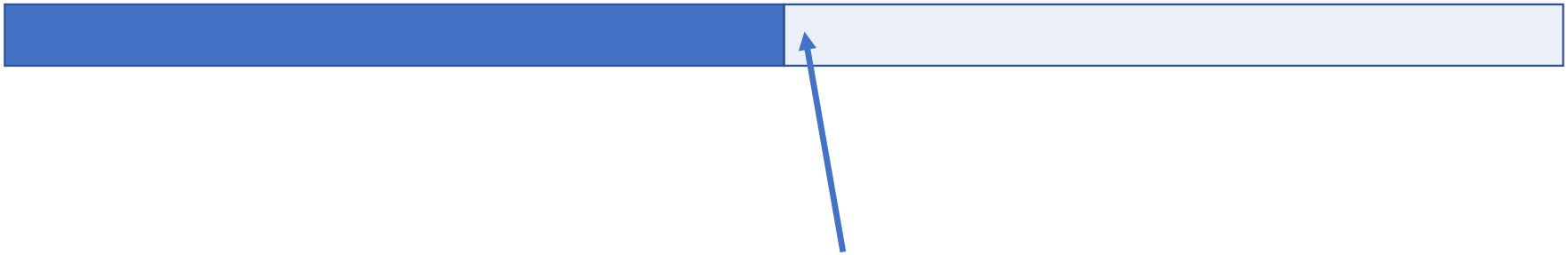
Zohar Manna and Amir Pnueli. Temporal Verification of Reactive Systems: Safety. Springer, New York, 1995.

Finite Trace – $\text{LTL}^{\mp}$

Have a strong view or weak view – for of the logic for the empty word.

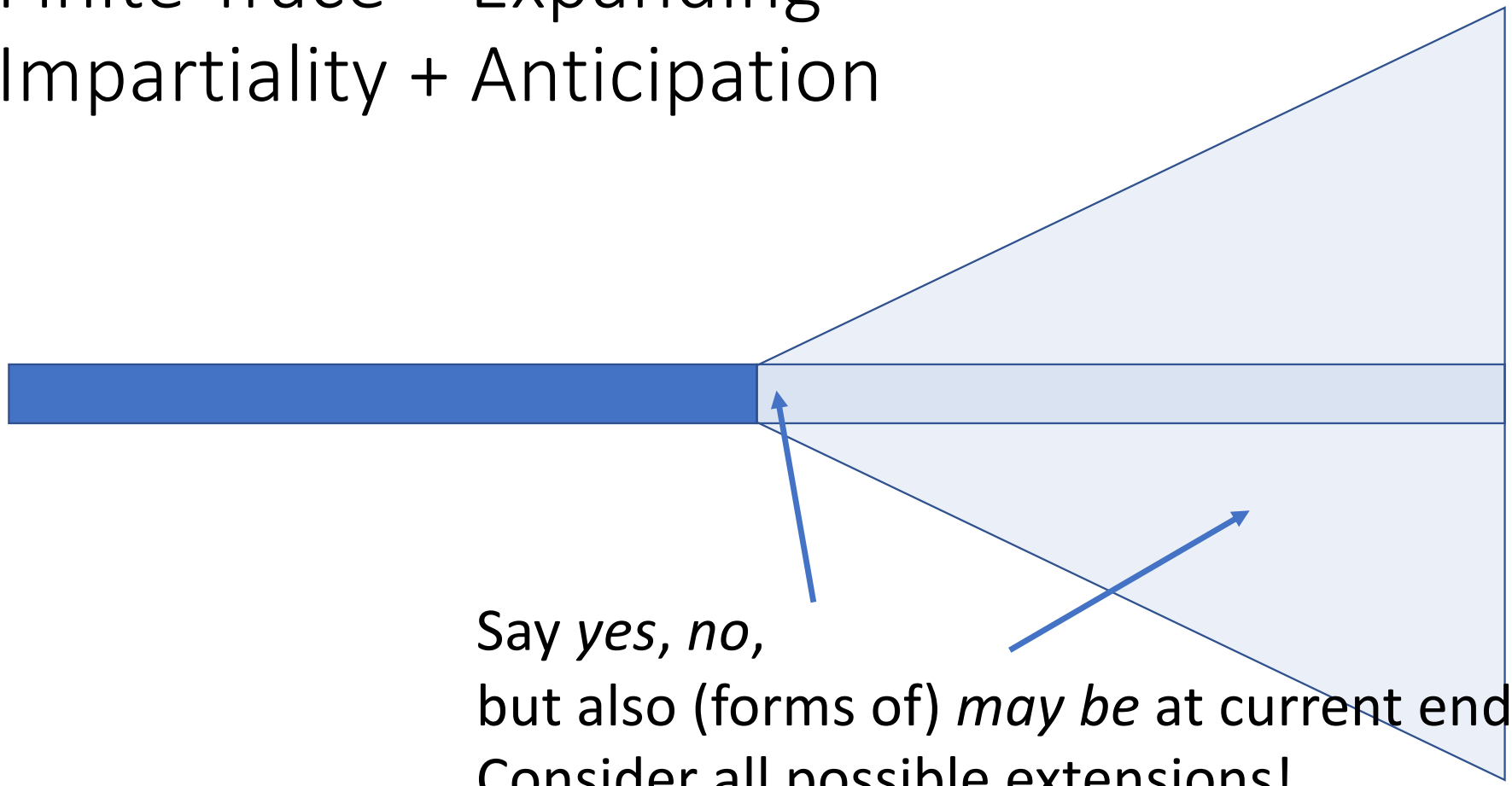
Cindy Eisner, Dana Fisman, John Havlicek, Yoad Lustig, Anthony McIsaac, and David Van Campenhout. Reasoning with temporal logic on truncated paths. In CAV, volume 2725 of LNCS, pages 27–39, 2003.

Finite Trace – Expanding - Impartiality



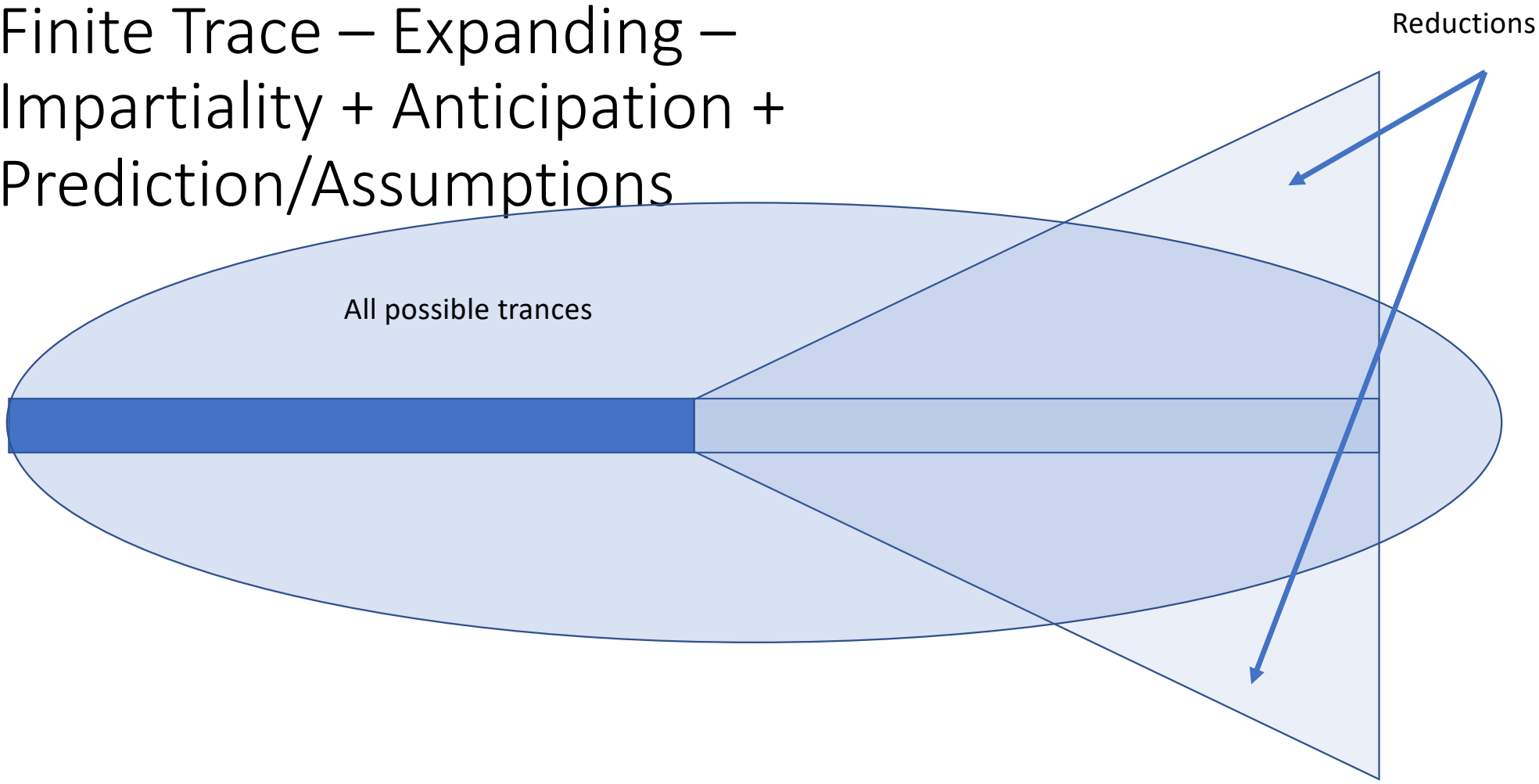
Say *yes*, *no*, but also (forms of) *may be* at current end!

Finite Trace – Expanding – Impartiality + Anticipation



Say *yes*, *no*,
but also (forms of) *may be* at current end,
Consider all possible extensions!

Finite Trace – Expanding – Impartiality + Anticipation + Prediction/Assumptions



The Formalities

LTL syntax

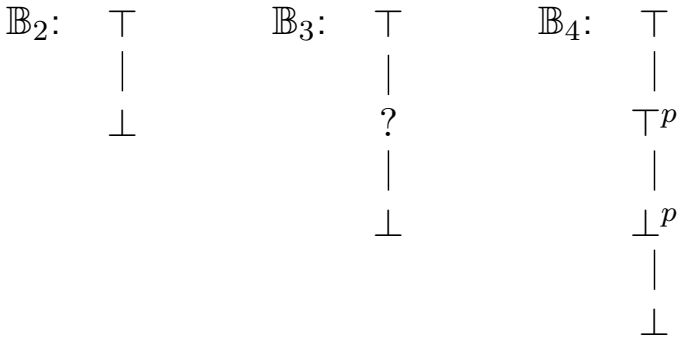
$$
\begin{aligned}
\varphi &::= true \mid p \mid \varphi \vee \varphi \mid \varphi \; U \; \varphi \mid X\varphi \\
\varphi &::= false \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \; R \; \varphi \mid \bar{X}\varphi \\
\varphi &::= \neg\varphi
\end{aligned}
$$

Multi-valued Temporal Logics

- Property *satisfied*, *falsified*, or *indefinite*

Definition (Truth Domain)

A *Truth Domain* is a finite De Morgan Lattice.



4-valued LTL for Finite Executions

Definition (FLTL$_4$ Semantics)

Let φ, ψ be LTL formulae and let $w \in \Sigma^+$ be a finite word. Then the semantics of φ with respect to w is inductively defined as follows:

$$[\![w \models \mathrm{X}\,\varphi]\!]_4 = \begin{cases} [\![w^2 \models \varphi]\!]_4 & \text{if } |w| > 1 \\ \bot^p & \text{else} \end{cases}$$

$$[\![w \models \overline{\mathrm{X}}\,\varphi]\!]_4 = \begin{cases} [\![w^2 \models \varphi]\!]_4 & \text{if } |w| > 1 \\ \top^p & \text{else} \end{cases}$$

Monitor construction

- Progression leads to Mealy machine

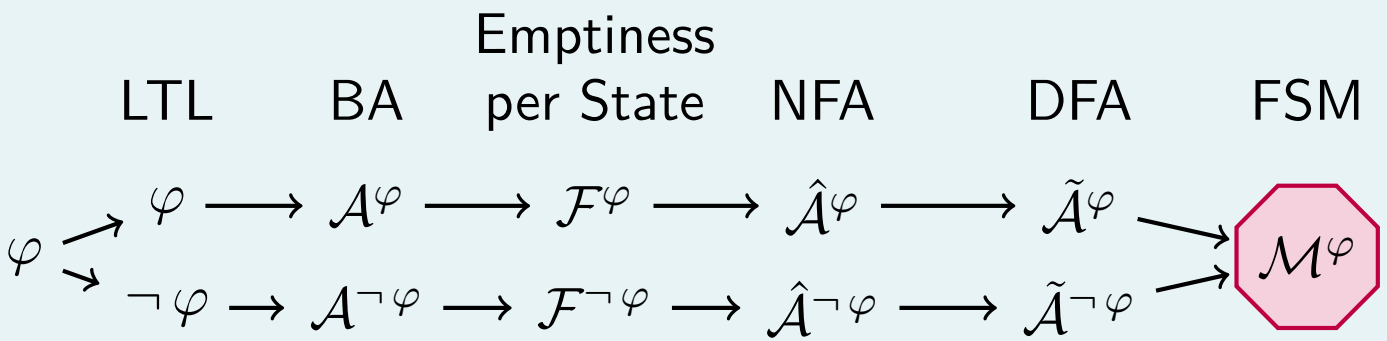
Anticipation – LTL_3

3-valued semantics for LTL over finite words

$$[u \models \varphi] = \begin{cases} \top & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \models \varphi \\ \bot & \text{if } \forall \sigma \in \Sigma^\omega : u\sigma \not\models \varphi \\ ? & \text{else} \end{cases}$$

Monitor Generation for LTL_3

The Construction

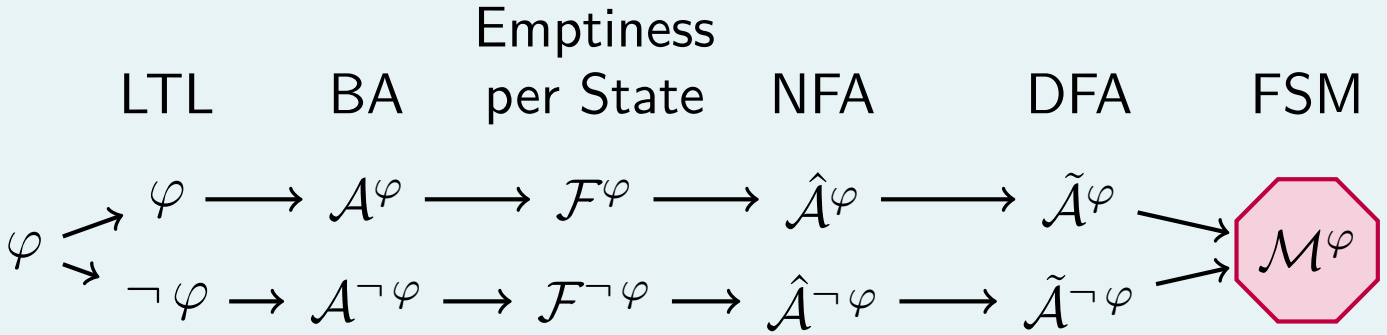


RV-LTL

$$[u \models \varphi]_{RV} = \begin{cases} \top & if\ [u \models \varphi]_3 = \top \\ \bot & if\ [u \models \varphi]_3 = \bot \\ \top^p & if\ [u \models \varphi]_3 =?\ and\ [u \models \varphi]_F = \top \\ \bot^p & if\ [u \models \varphi]_3 =?\ and\ [u \models \varphi]_F = \bot \end{cases}$$

Monitor Generation for RV-LTL

The Construction



In parallel FLTL4/FLTL monitor

Predictive Semantics revisited – LTL_3

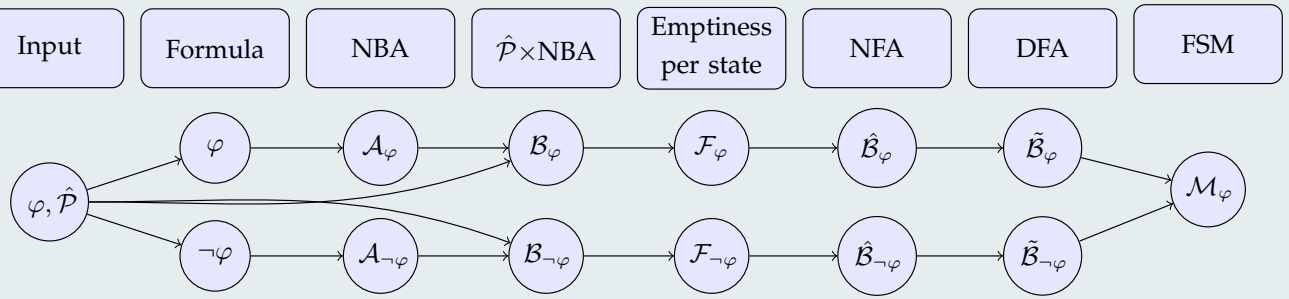
Definition (Predictive Semantics of LTL)

Let $\mathcal{P}$ be a program and let $\hat{\mathcal{P}}$ be an over-approximation of $\mathcal{P}$. Let $u \in \Sigma^*$ denote a finite trace. The *truth value* of u and an LTL formula φ with respect to $\hat{\mathcal{P}}$, denoted by $[\![u \models \varphi]\!]_{\hat{\mathcal{P}}} \in \mathbb{B}_4^i = \{\bot, \top, ?, ¿\}$, and defined as follows:

$$[\![u \models \varphi]\!]_{\hat{\mathcal{P}}} = \begin{cases} \top \text{ if } & u \in_\omega \mathcal{L}(\hat{\mathcal{P}}) \wedge \forall w \in \Sigma^\omega : \\ & uw \in \mathcal{L}(\hat{\mathcal{P}}) \Rightarrow [\![uw \models \varphi]\!]_\omega = \top \\ \bot \text{ if } & u \in_\omega \mathcal{L}(\hat{\mathcal{P}}) \wedge \forall w \in \Sigma^\omega : \\ & uw \in \mathcal{L}(\hat{\mathcal{P}}) \Rightarrow [\![uw \models \varphi]\!]_\omega = \bot \\ ? \text{ if } & \exists w, w' \in \Sigma^\omega : uw, uw' \in \mathcal{L}(\hat{\mathcal{P}}) \wedge \\ & [\![uw \models \varphi]\!]_\omega = \top \wedge [\![uw' \models \varphi]\!]_\omega = \bot \\ ¿ \text{ if } & u \notin_\omega \mathcal{L}(\hat{\mathcal{P}}) \end{cases}$$

Monitor construction

The procedure for getting $[u \models_{\hat{\mathcal{P}}} \varphi]$ for a given φ and over-approximation $\hat{\mathcal{P}}$



Comparison

Boolean laws

Tertium-non-datur laws

$$\varphi \vee \neg\varphi \equiv \mathit{true}$$
$$\varphi \wedge \neg\varphi \equiv \mathit{false}$$

distributive laws

$$\varphi \vee (\psi \wedge \eta) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \eta)$$
$$\varphi \wedge (\psi \vee \eta) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \eta)$$

de Morgan laws

$$\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$$
$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$$
$$\neg\neg\varphi \equiv \varphi$$

de Morgan-X law

$$\neg X\varphi \equiv_\omega \bar{X}\neg\varphi$$

de Morgan-U/R laws

$$\neg(\varphi\ U\ \psi) \equiv \neg\varphi\ R\ \neg\psi$$
$$\neg(\varphi\ R\ \psi) \equiv \neg\varphi\ U\ \neg\psi$$

unwinding laws

$$\varphi\ U\ \psi \equiv \psi \vee (\varphi \wedge X(\varphi\ U\ \psi))$$
$$\varphi\ R\ \psi \equiv \psi \wedge (\varphi \vee \bar{X}(\varphi\ R\ \psi))$$

Maxims

(1) **Existential next**

(2) **Complementation** by negation requires that a negated formula evaluates to the complemented and different truth value.

(3) **Impartiality** requires that a finite trace is not evaluated to $\top$ ($\bot$) if there still exists an infinite continuation leading to another verdict, and

(4) **Anticipation** requires that once every infinite continuation of a finite trace leads to the same verdict, then the finite trace evaluates to this very same verdict.

Results

	LTL	FLTL	$\text{LTL}^{\mp}$	LTL_3	RV-LTL
Domain	Σ^{∞}	$u \neq \emptyset, u \in \Sigma^*$ (4)	Σ^* (9)	Σ^* (15)	$u \neq \emptyset, u \in \Sigma^*$ (21)
Exitential Next (Maxim 1)		yes	yes (+)/no (-)	yes	yes
Complementation by Negation (Maxim 2)		yes	no	no	yes
Impartiality (Maxim 3)		no	no	yes	yes
Anticipation (Maxim 4)		no	no	yes	yes
Boolean laws	yes (1)	yes (6)	no (10)	yes (17)	yes (23)
Equivalences (Fig. 2)	yes (2)	yes (7)	yes (12)	yes (18)	yes (24)
LTL compliant		no (5)	no (11)	yes (16)	no (25)
Negation normalform	yes (3)	yes (8)	yes (13)	yes (19)	yes (24)
Inductive definition	yes	yes	yes	no (14)	no (22)