Pure-Past Linear Temporal and Dynamic Logic on Finite Traces

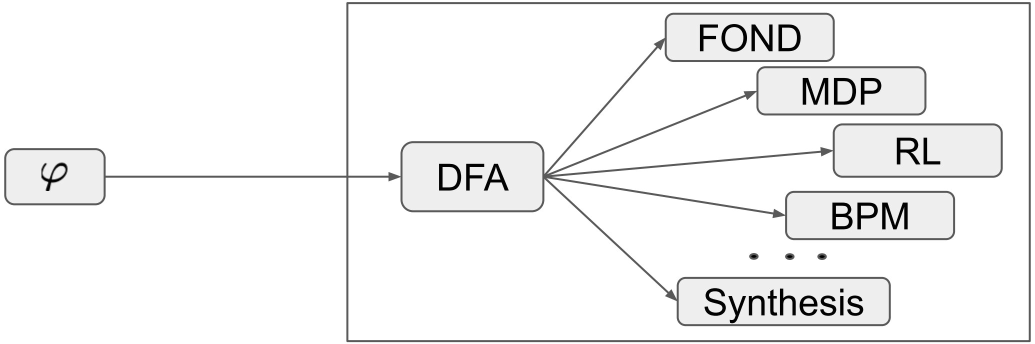
Giuseppe De Giacomo, Antonio Di Stasio, **Francesco Fuggitti**, Sasha Rubin

IJCAI-PRICAI 2020
Survey Track

YORK UNIVERSITÉ UNIVERSITY
SAPIENZA UNIVERSITÀ DI ROMA
THE UNIVERSITY OF SYDNEY

Context and Motivation

- Linear Temporal and Dynamic Logic on *finite* traces (LTL_f /LDL_f) studied as compelling formal languages to express temporal specifications
  - *Ease of use* and *intuitiveness*
  - Possibility of using DFAs in the translation



$PLTL_f$ /$PLDL_f$

- Most of work focused on the pure-future LTL_f /LDL_f

- Sometimes specifications are *easier* and *more natural* to express referring to the past ($PLTL_f$ /$PLDL_f$) [Lichtenstein et al. 1985]
  - non-Markovian models [Gabaldon2011]
  - non-Markovian rewards in MDPs [Bacchus et al. 1996]
  - normative properties in multi-agent systems [FisherWooldridge2005;Knobbout et al. 2016;Alechina et al. 2018]

- Computational advantage wrt LTL_f /LDL_f [Chandra et al., 1981]

Examples

Example 1: "every time you took the bus, you bought a new ticket beforehand"

PLTL$_f$: $$\boxminus(takeB \Rightarrow \ominus(\neg takeB\ \mathcal{S}\ buyT))$$

LTL$_f$: $$(buyT\ \mathcal{R}\ takeB) \ \wedge\ \Box(takeB \Rightarrow (buyT\ \vee\ \circ(buyT\ \mathcal{R}\ \neg takeB)))$$

Example 2: "every time, if the cargo-ship departed (*cs*), then there was an alternation of *grab* and unload (*unl*) of containers before"

PLDL$_f$: $$[\![true^*]\!](\langle\!\langle cs\rangle\!\rangle tt \Rightarrow \langle\!\langle(unl; grab)^*; (unl; grab)\rangle\!\rangle start)$$

LDL$_f$: $$\langle(\neg cs + (grab \wedge \neg cs); (unl; (grab \wedge \neg cs))^*; (cs \wedge unl)); \neg cs^*\rangle end$$

From φ to Automata

- $\text{LTL}_f/\text{LDL}_f$:

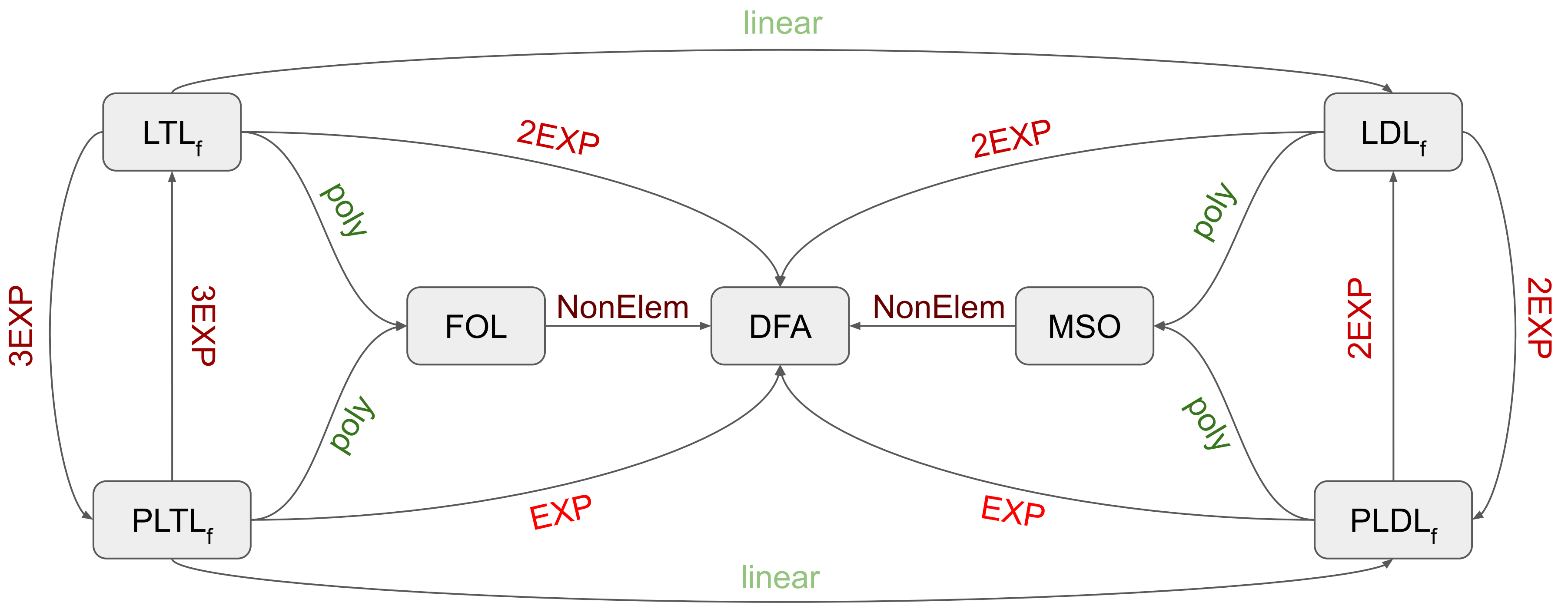
$$\begin{array}{c}\text{LTL}_f/\text{LDL}_f\\(L)\end{array} \xrightarrow{\text{lin}} \text{AFA} \xrightarrow{\text{EXP}} \text{NFA} \xrightarrow{\text{EXP}} \begin{array}{c}\text{DFA}\\(L)\end{array}$$

- $\text{PLTL}_f/\text{PLDL}_f$:

$$\begin{array}{c}\text{PLTL}_f/\text{PLDL}_f\\(L)\end{array} \xrightarrow{\text{lin}} \begin{array}{c}\text{LTL}_f/\text{LDL}_f\\(L^{rev})\end{array} \xrightarrow{\text{lin}} \begin{array}{c}\text{AFA}\\(L^{rev})\end{array} \xrightarrow{\text{EXP}} \begin{array}{c}\text{DFA}\\(L)\end{array}$$

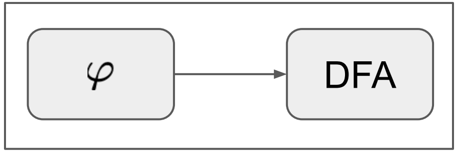
- Given an AFA of k states for language L, there exists a DFA of at most 2^k states for language $L^{reverse}$ [Chandra et al. 1981]

Transformations

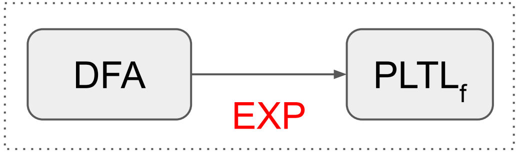


Useful Results

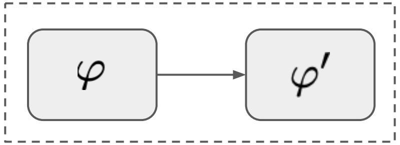
1. Transformations seen before



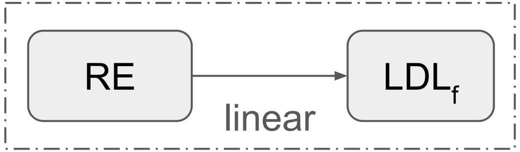
2. From DFA (star-free) to PLTL_f [Maler&Pnueli 1990]



3. Syntactic swap of temporal operators

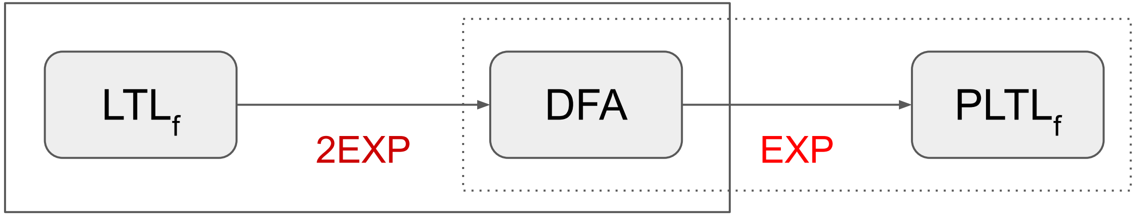


4. From RE to LDL_f [DeGiacomo&Vardi 2013]

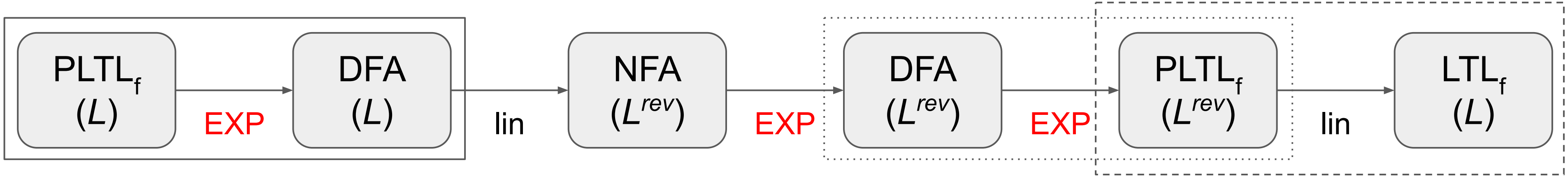


LTL_f to $PLTL_f$ (and vice versa)

- From LTL_f to $PLTL_f$

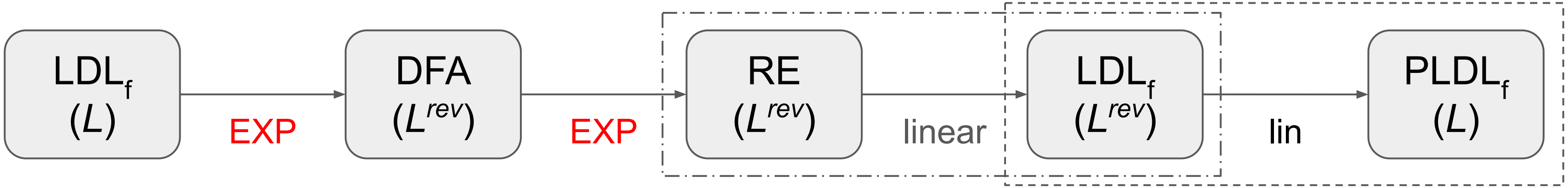


- From $PLTL_f$ to LTL_f

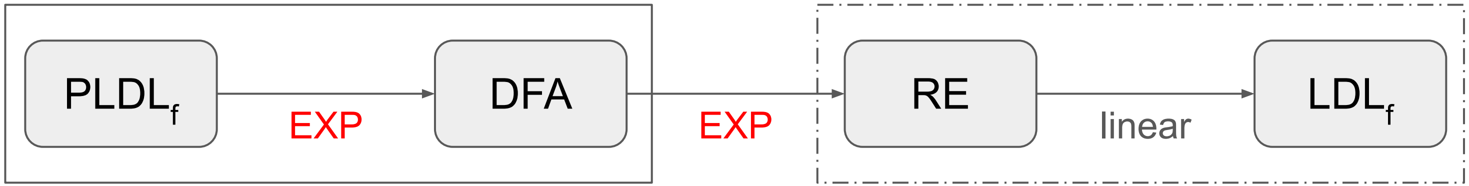


LDL_f to $PLDL_f$ (and vice versa)

- From LDL_f to $PLDL_f$



- From $PLDL_f$ to LDL_f



Impact of adopting $PLTL_f$ /$PLDL_f$

- Exponential gain reflected in an exponential gain in solving different forms of sequential decision making involving temporal specifications:
  - FOND Planning for temporally extended goals
  - MDPs with non-Markovian rewards
  - Reinforcement Learning with temporally extended rewards
  - Planning in non-Markovian domains
  - non-Markovian decision processes
- Problems are EXPTIME-complete in the goal/reward natively expressed in $PLTL_f$ /$PLDL_f$ (vs. 2EXPTIME-complete in the LTL_f /LDL_f goal/reward)

Takeaways

- If you can *naturally* express the specification in $PLTL_f$ /$PLDL_f$, then do it to get the computational advantage.
- Converting LTL_f /LDL_f into $PLTL_f$ /$PLDL_f$ to get the exponential advantage is not computationally sensible
- Complexities are just worst-case, in many applications the size of the resulting DFA is actually manageable