Expressivity of Linear Temporal Logic under Team Semantics

Max Sandström

University of Helsinki

max.sandstrom@helsinki.fi

10.08.2020

Overview

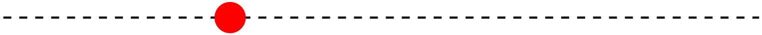
Linear temporal logic (LTL) is a language for describing events on a timeline, known as traces.

Linear temporal logic (LTL) is a language for describing events on a timeline, known as traces.

It has applications in system verification, wherein the timelines are thought of as computations of a program.

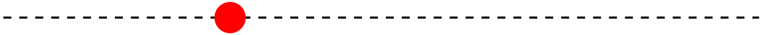
Linear temporal logic (LTL) is a language for describing events on a timeline, known as traces.

It has applications in system verification, wherein the timelines are thought of as computations of a program.
Within the logic one may express (among others) the notions of:
Eventually *Fr*



Linear temporal logic (LTL) is a language for describing events on a timeline, known as traces.

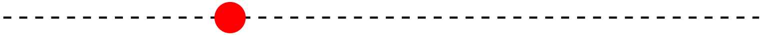
It has applications in system verification, wherein the timelines are thought of as computations of a program.
Within the logic one may express (among others) the notions of:
Eventually Fr



Globally Gr

Linear temporal logic (LTL) is a language for describing events on a timeline, known as traces.

It has applications in system verification, wherein the timelines are thought of as computations of a program.
Within the logic one may express (among others) the notions of:
Eventually Fr



Globally Gr

Until rUy

The properties defined by the formulas of LTL are sets of traces satisfying the formula.

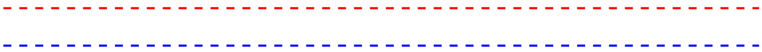


Figure: Trace property for $Gr \vee Gb$

The properties defined by the formulas of LTL are sets of traces satisfying the formula.

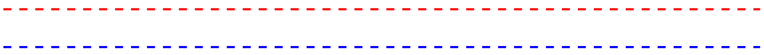
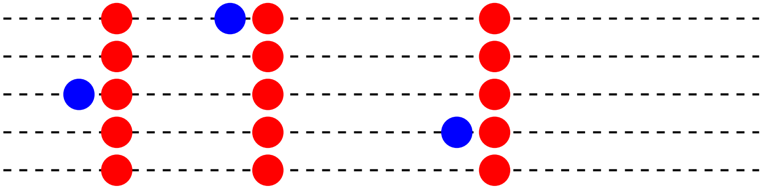


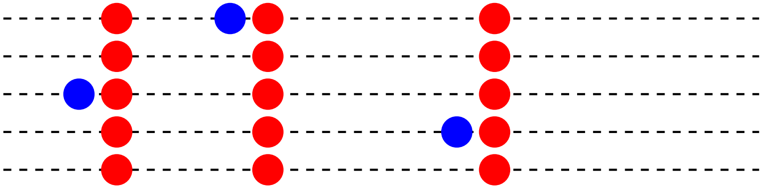
Figure: Trace property for $Gr \vee Gb$

However, there are sets of traces, which are relevant to software verification that are not definable by LTL formulas. Specifically the limitation of the logic is due to the formulas only referring to single assignments; any property involving dependencies between the traces are not definable in LTL.

For instance the following property, where a blue world on any trace is followed by red on all traces is not definable in LTL.



For instance the following property, where a blue world on any trace is followed by red on all traces is not definable in LTL.

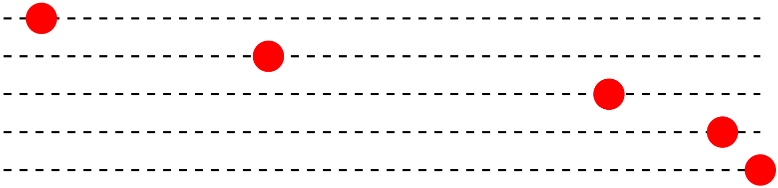


Team semantics provides one framework for referring to multiple traces.

There are two ways of defining team semantics for linear temporal logic: an asynchronic and a synchronic.

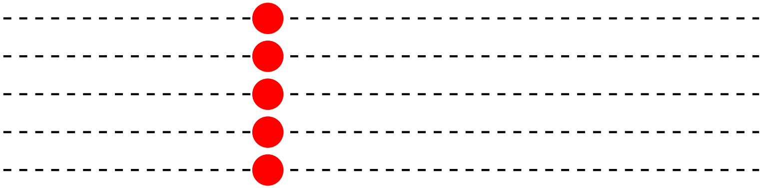
There are two ways of defining team semantics for linear temporal logic: an asynchronic and a synchronic.

In the asynchronic semantics teams satisfying a formula may do so at different points in time, e.g. a team satisfying $F^a r$ may look like



There are two ways of defining team semantics for linear temporal logic: an asynchronic and a synchronic.

In the synchronic semantics teams satisfying a formula do so at the same points in time, e.g. a team satisfying $F^s r$ may look like



The two semantics for TeamLTL have different expressive power and complexity. It has been shown that the synchronous semantics are expressively incomparable to HyperLTL (a different approach of grasping sets of traces). The asynchronous semantics on the other hand collapses to a fragment of HyperLTL.
The purpose of this work is to complete the picture regarding the expressivity of the two logics with regards to the better known team logics, by defining compositional translations between the logics.

The first hurdle in this process is that members of teams in TeamLTL and first order team semantics are essentially different things.

The first hurdle in this process is that members of teams in TeamLTL and first order team semantics are essentially different things.
In TeamLTL teams consist of functions $\pi \colon W \to \mathcal{P}(\Phi)$, which map the possible worlds of W to sets of propositions that hold in the world.

The first hurdle in this process is that members of teams in TeamLTL and first order team semantics are essentially different things.
In TeamLTL teams consist of functions $\pi\colon W \to \mathcal{P}(\Phi)$, which map the possible worlds of W to sets of propositions that hold in the world.
On the other hand the teams of first order team semantics are made up of functions $s\colon Var \to M$, which map variables to elements of the domain.

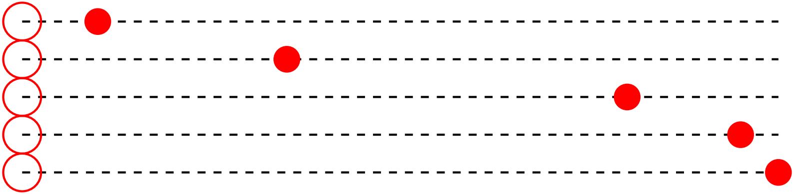
The first hurdle in this process is that members of teams in TeamLTL and first order team semantics are essentially different things.
In TeamLTL teams consist of functions $\pi\colon W \to \mathcal{P}(\Phi)$, which map the possible worlds of W to sets of propositions that hold in the world.
On the other hand the teams of first order team semantics are made up of functions $s\colon \mathit{Var} \to M$, which map variables to elements of the domain.
Hence the construction of the model and team corresponding to a set of traces needs to be done with careful consideration.

The idea is to take a copy of the natural numbers for each trace in the team, assign a predicate for each proposition in Φ, and take an order relation only defined within each copy of the naturals. Then a model can be created where each natural is placed in a predicate if the corresponding proposition holds in the respective world. The team then consists of interpretations mapping the variable x to the first world on the corresponding trace.

Translation of Asynchronous Eventually

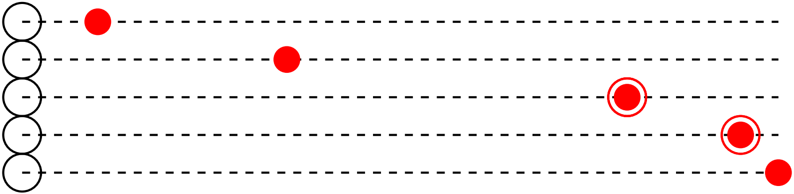
The initial team is insufficient to capture events in the future.

$P_r x$



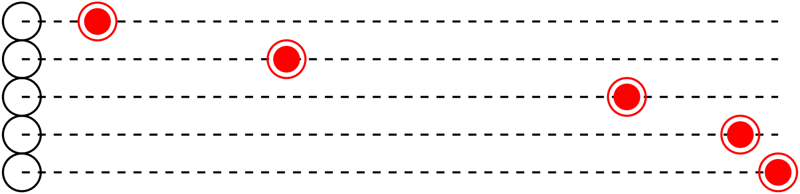
Translation of Asynchronous Eventually

We update the team using the existential quantifier.
$\exists y(P_r y)$



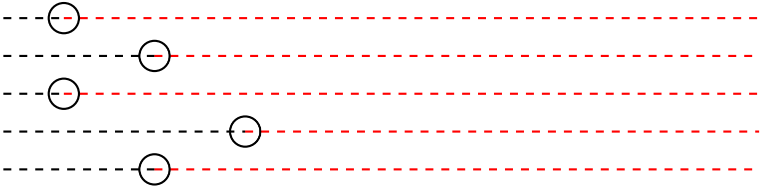
Translation of Asynchronous Eventually

We restrict the supplementation to the corresponding trace by comparing the new variable y to x.
$\exists y(x \leq y \wedge P_r y)$



Translation of Asynchronous Globally

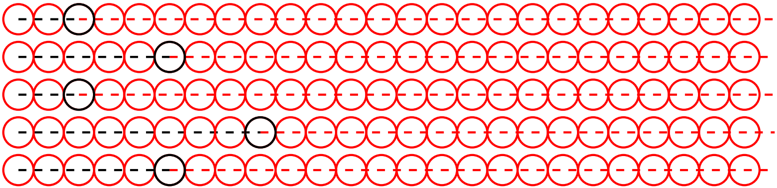
For globally let's look at a situation where the team is already updated.



Translation of Asynchronous Globally

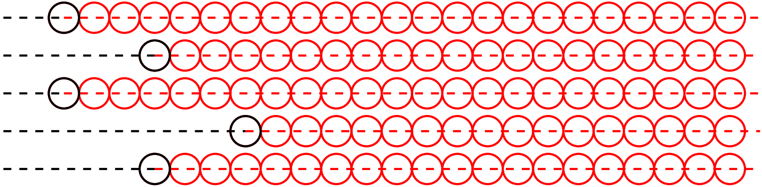
Since we want to refer to all points in time universal quantification is the intuitive choice.

$\forall y(P_r y)$



Translation of Asynchronous Globally

This time we restrict the team by splitting away the undesired part.
$\forall y(x \leq y \hookrightarrow P_r y)$



Here $x \leq y \hookrightarrow P_r y = \neg x \leq y \vee (x \leq y \wedge P_r y)$.

In this way we define a translation from TeamLTL[a] to FO^3 under team semantics as follows:

$$\begin{aligned}
ST_x(p_i) &= P_i(x)\\
ST_x(\neg p_i) &= \neg P_i(x)\\
ST_x(\varphi \wedge \psi) &= ST_x(\varphi) \wedge ST_x(\psi)\\
ST_x(\varphi \vee \psi) &= ST_x(\varphi) \vee ST_x(\psi)\\
ST_x(X\varphi) &= \exists y(x < y \wedge ST_y(\varphi) \wedge\\
&\quad \forall z \neg(x < z \wedge z < y))
\end{aligned}$$

$$
\begin{aligned}
ST_x(G^a\varphi) &= \forall y(x \leq y \hookrightarrow ST_y(\varphi)) \\
ST_x(F^a\varphi) &= \exists y(x \leq y \wedge ST_y(\varphi)) \\
ST_x(\varphi U^a \psi) &= \exists y(x \leq y \wedge ST_y(\psi) \wedge \\
&\quad \forall z((x \leq z \wedge z < y) \hookrightarrow ST_z(\varphi))) \\
ST_x(\varphi R^a \psi) &= \forall y(x \leq y \hookrightarrow (ST_y(\psi) \vee \\
&\quad \exists z(x \leq z \wedge z < y \wedge ST_z(\varphi)))).
\end{aligned}
$$

The previous translation relies heavily on the fact that both the logics are flat. However some extensions of asynchronous TeamLTL are not flat, and hence a translation that does not utilize the flatness property is of interest.

The previous translation relies heavily on the fact that both the logics are flat. However some extensions of asynchronous TeamLTL are not flat, and hence a translation that does not utilize the flatness property is of interest.
The translation is an evolution of the previous translation with two changes. In practice the two differences stem from the need to pick out single worlds into the new team and limiting the scope of the universal quantifier. These goals are achieved with the dependence atom $=(x, y)$ and the dual of the existential quantifier $\sim \exists \sim \varphi$, respectively.

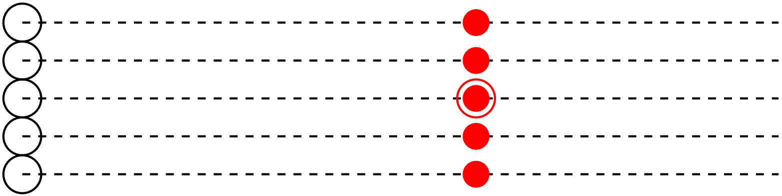
Thus we define a translation of TeamLTL^a formulas to $\text{FO}^3(=(\dots), \sim)$ as follows:
The translation is analogous to the previous translation for the atomic propositions, $\wedge$, $\vee$, and X.

$$
\begin{aligned}
ST_x^*(F^a\varphi) &= \exists y(x \leq y \wedge =(x,y) \wedge ST_y^*(\varphi)) \\
ST_x^*(G^a\varphi) &= \sim \exists y(x \leq y \wedge =(x,y) \wedge \sim ST_y^*(\varphi)) \\
ST_x^*(\varphi U^a \psi) &= \exists y(x \leq y \wedge =(x,y) \wedge ST_y^*(\psi) \wedge \\
&\quad \sim \exists z(x \leq z \wedge z \leq y \wedge =(x,z) \wedge \sim ST_z^*(\varphi))) \\
ST_x^*(\varphi R^a \psi) &= \sim \exists y(x \leq y \wedge =(x,y) \wedge \sim ST_y^*(\psi) \wedge \\
&\quad \exists z(x \leq z \wedge z < y \wedge =(x,z) \wedge \sim ST_z^*(\varphi))).
\end{aligned}
$$

Translation of Synchronous Eventually

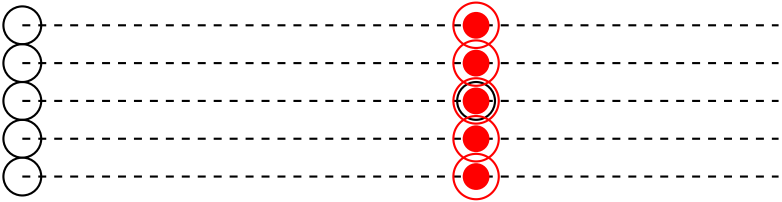
The translation for the synchronous semantics is hinged upon finding representatives for the slice of time.

$\exists y(=(y) \wedge P_r y)$



Translation of Synchronous Eventually

Then we use existential quantification and an equal level predicate to assign a world from the slice to each interpretation.
$\exists y(=(y) \wedge \exists z(E(y,z) \wedge x \leq z \wedge P_r z)$



Capturing the representative for the team requires an additional variable, and thus we define a translation from TeamLTL^s to FO$^4(=(\dots),\sim)$ as follows: The translation is analogous to the previous translations for the atomic propositions, $\wedge$, $\vee$, and X.

$$
\begin{aligned}
ST^*_x(F^s\varphi) &= \exists y(=(y) \wedge \exists z(E(y,z) \wedge x \leq z \wedge ST^*_z(\varphi)))\\
ST^*_x(G^s\varphi) &= \sim \exists y(=(y) \wedge \exists z(E(y,z) \wedge x \leq z \wedge \sim ST^*_z(\varphi)))\\
ST^*_x(\varphi U^s\psi) &= \exists y(=(y) \wedge \exists z(E(y,z) \wedge x \leq z \wedge ST^*_z(\psi) \wedge\\
&\quad \sim \exists y(=(y) \wedge \exists w(E(y,w) \wedge x \leq w \wedge w \leq z \wedge\\
&\quad ST^*_w(\varphi)))))\\
ST^*_x(\varphi R^s\psi) &= \sim \exists y(=(y) \wedge \exists z(E(y,z) \wedge x \leq z \wedge \sim ST^*_z(\psi) \wedge\\
&\quad \exists y(=(y) \wedge \exists w(E(y,w) \wedge x \leq w \wedge w < z \wedge\\
&\quad \sim ST^*_w(\varphi))))).
\end{aligned}
$$

Conclusion

We introduced three translations from semantics of TeamLTL to first order team semantics. These translations lay bare the hierarchy of expressive power for these logics.

Conclusion

We introduced three translations from semantics of TeamLTL to first order team semantics. These translations lay bare the hierarchy of expressive power for these logics.

- Asynchronous TeamLTL can be translated to FO^3 under team semantics.

Conclusion

We introduced three translations from semantics of TeamLTL to first order team semantics. These translations lay bare the hierarchy of expressive power for these logics.

- Asynchronous TeamLTL can be translated to FO^3 under team semantics.
- Without assuming flatness, asynchronous TeamLTL can be translated to $\mathrm{FO}^3(=(\dots), \sim)$. This translation enables us to define translations of extensions without the flatness property.

Conclusion

We introduced three translations from semantics of TeamLTL to first order team semantics. These translations lay bare the hierarchy of expressive power for these logics.

- Asynchronous TeamLTL can be translated to FO^3 under team semantics.
- Without assuming flatness, asynchronous TeamLTL can be translated to $\mathrm{FO}^3(=(\dots), \sim)$. This translation enables us to define translations of extensions without the flatness property.
- Synchronous TeamLTL can be translated to $\mathrm{FO}^4(=(\dots), \sim)$.

Thank you for listening!