Technical Report soft-14-01, Chair for Software Engineering, University of Konstanz, Copyright by the Authors 2014

On the Relationship of Event Order Logic and Linear Temporal Logic

Adrian Beer, Florian Leitner-Fischer and Stefan Leue

University of Konstanz, Germany

Abstract. In recent work [1,2] we have proposed the event order logic (EOL) which is used to reason about the occurrence and order of events in formal system models. In this paper we will discuss the relationship of the event order logic and the linear temporal logic and further more show how EOL formulas can be translated into LTL formulas.

1 Introduction

In recent work [1,2] we have proposed the event order logic (EOL) which is used to reason about the occurrence and order of events in formal system models. In this paper we will discuss the relationship of the event order logic and the linear temporal logic and further more show how EOL formulas can be translated into LTL formulas.

The contributions of this paper can be summarized as follows:

- The semantics of the EOL is further refined.
- We define equivalence transformations for EOL formulas..
- We defined the Event Order Logic Normal Form (EONF) and prove that each EOL formula can be transformed into an EOL formula in EONF.
- We show how EOL formulas in EONF can be translated into LTL formulas that accept the same set of paths and are thus equivalent.

The remainder of this paper is structured as follows: We first briefly introduce the syntax and semantics of linear temporal logic, a running example of a railroad crossing that we use throughout the paper for illustrative purposes, and the underlying system model in Section 2. We further refine the semantics of the event order logic and show how EOL formulas can be transformed into equivalent EOL formulas in Section 3. Section 4 is devoted to the translation of EOL formulas to LTL formulas. We conclude in Section 5.

2 Preliminaries

2.1 Linear Temporal Logic

The analysis aims at identifying the violation of functional safety requirements. Such a violation is also referred to as a hazard. We use linear time temporal logic (LTL) using its standard syntax and semantics as defined in [3] in order to specify hazards.

Hazards imply the reachability of unsafe states and they hence belong to the class of reachability properties. Hence we only need to consider finite execution fragments [4]. Hazards fall within the class of safety properties in the commonly used classification scheme of safety and liveness properties. We use $T \vDash_l \varphi$ to express that the LTL formula φ holds for the transition system T and $\sigma \vDash_l \varphi$ respectively for execution traces.

Definition 1. *Syntax of LTL. Formulas in LTL over the set AP of atomic proposition are formed according to the following grammar given in BNF:*

$$\varphi ::= true \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\ \varphi \mid \varphi_1\ \mathcal{U}\ \varphi_2$$

where $a \in AP$. Additionally the following operators are defined as syntactic sugaring of those above:

$$\Diamond\varphi = true\ \mathcal{U}\ \varphi \text{ and } \Box\varphi = \neg\Diamond\neg\varphi$$

Definition 2. *Semantics of LTL over Executions and States. Let $T = (S, Act, \rightarrow, I, AP, L)$ be a transition system, let φ an LTL formula over AP and σ a finite execution of T and $\sigma[j...]$ the suffix of σ starting at s_j. then the semantic is defined by induction on the structure of φ*

- $\sigma \vDash_l true$
- $\sigma \nvDash_l false$
- $\sigma \vDash_l p$ *iff* $p \in L(s_0)$
- $\sigma \vDash_l \neg\varphi$ *iff* $\sigma \nvDash_l \varphi$
- $\sigma \vDash_l \varphi_1 \wedge \varphi_2$ *iff* $\sigma \vDash_l \varphi_1$ *and* $\sigma \vDash_l \varphi_2$
- $\sigma \vDash_l \varphi_1 \vee \varphi_2$ *iff* $\sigma \vDash_l \varphi_1$ *or* $\sigma \vDash_l \varphi_2$
- $\sigma \vDash_l \bigcirc\ \varphi$ *iff* $\sigma[1...] \vDash_l \varphi$
- $\sigma \vDash_l \varphi_1 \mathcal{U}\ \varphi_2$ *iff* $\exists k \geq 0\ .\ \sigma[k...] \vDash_l \varphi_2$ *and* $\forall 0 \leq j < k\ .\ \sigma[j...] \vDash_l \varphi_1$

and for the derived operators $\Diamond$ and $\Box$.

- $\sigma \vDash_l \Diamond\varphi$ *iff* $\exists j : j \geq 0\ .\ \sigma[j..] \vDash_l \varphi$
- $\sigma \vDash_l \Box\varphi$ *iff* $\forall j : j \geq 0\ .\ \sigma[j..] \vDash_l \varphi$

2.2 Running Example: Railroad Crossing

We will demonstrate the presented definitions on a running example of a railroad crossing system. In the running example a train can approach the crossing (Ta), cross the crossing (Tc) and finally leave the crossing (Tl). Whenever a train is approaching, the gate should close (Gc) and will open when the train has left the crossing (Go). It might also be the case that the gate fails (Gf). The car approaches the crossing (Ca) and crosses the crossing (Cc) if the gate is open and finally leaves the crossing (Cl). We are interested in finding those events that lead to a hazard state in which both the car and the train are in the crossing. This hazard can be characterized by the LTL formula $\varphi = \Box\neg(car\_crossing \wedge train\_crossing)$.

2.3 System Model and Events

The systems that we apply causality checking to are concurrent systems. For the formalization of the system model we follow the formalization of a model for concurrent computing systems proposed in [4]. The system model is given by a Transition System which is defined as follows:

Definition 3. *Transition System. A* transition system *TS is a tuple* $(S, Act, \rightarrow, I, AP, L)$ *where S is a finite set of* states*, Act is a finite set of* actions*, $\rightarrow \subseteq S \times Act \times S$ is a* transition relation*, $I \subseteq S$ is a set of* initial states*, AP is a set of* atomic propositions*, and $L : S \rightarrow 2^{AP}$ is a* labeling function*.*

A Transition System defines a Kripke structure. Each state $s \in S$ is labeled with the set $L(s)$ of all atomic state propositions that are true in this state. The set Act contains all actions that can trigger the system to transit from some state into a successor state. The execution semantics of a transition system is defined as follows:

Definition 4. *Execution Trace of a Transition System. Let $T = (S, Act, \rightarrow, I, AP, L)$ be a transition system. A finite execution σ of T is an alternating sequence of states $s \in S$ and actions $\alpha \in Act$ ending with a state. $\sigma = s_0\ \alpha_1\ s_1\ \alpha_2\ ...\ \alpha_n\ s_n$ s.t. $s_i \xrightarrow{\alpha_{i+1}} s_{i+1}$ for all $0 \leq i < n$.*

In the following we will use short-hand notation σ = "a_{α_1},a_{α_2}, ... , a_{α_n}" for an execution trace $\sigma = s_0\ \alpha_1\ s_1\ \alpha_2\ ...\ \alpha_n\ s_n$. The trace σ = "Ta, Ca, Gf, Cc, Tc", for instance, is a trace of the railroad example from Sec. 2.2 where the train and the car are approaching the crossing (Ta, Ca), the gate fails to close (Gf), the car crosses the crossing (Cc) and finally the train crosses the crossing (Tc).

We can partition the set of all possible execution traces Σ of a transition system T into the set of "good" execution traces, denoted Σ_G, where the LTL formula is not violated and thus the hazard does not occur, and the set of "bad" execution traces, denoted Σ_B, where the LTL formula is violated and thus the hazard occurs. The elements of Σ_B are also referred to as counterexamples in model checking. The trace σ = "Ta, Ca, Gf, Cc, Tc" we already discussed above is a "bad" execution trace, since bot the car and the train are on the crossing at the same time and thus the LTL property is violated. An example for a "good" trace is σ' = "Ta, Ca, Gf, Cc, Cl, Tc" where the car leaves the crossing (Cl) before the train is crossing (Tc) and consequently the train and the car are not on the crossing at the same time and the LTL formula is not violated.

Definition 5. *Good and Bad Execution Traces. Let $T = (S, Act, \rightarrow, I, AP, L)$ be a transition system, let φ an LTL formula over AP and Σ that set of all possible finite executions of T. The set Σ is divided into into the set of "good" execution traces Σ_G and in the set of "bad" execution traces Σ_B as follows: $\Sigma_G = \{\sigma \in \Sigma \mid \sigma \vDash_l \varphi\}$, $\Sigma_B = \{\sigma \in \Sigma \mid \sigma \nvDash_l \varphi\}$ and $\Sigma_G \cup \Sigma_B = \Sigma$ and $\Sigma_G \cap \Sigma_B = \varnothing$.*

We assume that for a given execution trace σ of a transition system T, Act contains the events that we wish to reason about. For an LTL formula φ

specifying a safety requirement and an execution trace σ, the hazard described by the safety requirement occurs on σ if and only if $\sigma \not\models_l \varphi$ holds. Notice that since each transition is only labeled with one action, only one event can occur per transition. In order to be able to reason about the causality of events we have to formally capture the occurrence of events. We assume that there exists a set $\mathcal{A}$ of event variables that contains a boolean variable a for each action $\alpha \in Act$ for some given transition system. The variable a_{Ta} for instance represents the event train approaching the crossing. If multiple instances of one event type occur on one execution trace, for example the two train approaching events on "Ta,Gc,Tc,Tl,Go,Ta", the variables representing them are numbered according to their occurrence, for our example a_{Ta_1} and a_{Ta_2}. In other words, the *i-th* occurrence of some action of type α will be represented by the boolean variable a_{α_i}. In the following we also abbreviate the event variable a_{Ta} by Ta.

Definition 6. *Events, Event Types and Event Variables. Let $T = (S, Act, \rightarrow, I, AP, L)$ a transition system and $\sigma = s_0, \alpha_1, s_1, \alpha_2, \ldots \alpha_n, s_n$ a finite execution trace of T. We define the following: each $\alpha \in Act$ defines an* event type *α. α_i of σ denotes the* i-th *occurrence of an event of the* event type *α. The variable representing the occurrence of the event α_i is denoted by a_{α_i}, and the set $\mathcal{A} = \{a_{\alpha_1}, ..., a_{\alpha_n}\}$ contains a boolean variable for each occurrence of an event.*

3 Event Order Logic

Event variables allow us to reason about the occurrence of single events, but since we want to reason about the combination of events, we need a formalism that allows us to express the occurrence of event combinations. In [2] we presented the event order logic (EOL) which allows one to connect event variables from $\mathcal{A}$ with the boolean connectives $\wedge$, $\vee$ and $\neg$. To express the ordering of events we introduced the ordered conjunction operator $\curlywedge$. The formula $a \curlywedge b$ with events a and b is satisfied if and only if events a and b occur in a trace and a occurs before b. In addition to the $\curlywedge$ operator we introduced the interval operators $\curlywedge_{[}$, $\curlywedge_{]}$, and $\curlywedge_{<}\ \phi\ \curlywedge_{>}$, which define an interval in which an event has to hold in all states. These interval operators are necessary to express the causal non-occurrence of events. We present here an amended version of the event order logic and further refine its semantics.

Definition 7. *Syntax of Event Order Logic (EOL). Simple EOL formulas over a set $\mathcal{A}$ of event variables are formed according to the following grammar:*

$$\phi ::= a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \phi_1 \vee \phi_2$$

where $a \in \mathcal{A}$ and ϕ, ϕ_1 and ϕ_2 are simple EOL formulas. Complex EOL formulas are formed according to the following grammar:

$$\psi ::= \phi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 \mid \psi_1 \curlywedge \psi_2 \mid \psi \curlywedge_{[}\ \phi \mid \phi \curlywedge_{]}\ \psi \mid \psi_1 \curlywedge_{<} \phi \curlywedge_{>} \psi_2$$

where ϕ is a simple EOL formula and ψ_1 and ψ_2 are complex EOL formulas. Note that the $\neg$ operator binds more tightly than the $\curlywedge$, $\curlywedge_{[}$, $\curlywedge_{]}$, and $\curlywedge_{<}\ \phi\ \curlywedge_{>}$, operators and those bind more tightly than the $\vee$ and $\wedge$ operator.

The formal semantics of this logic is defined over execution traces. Notice that the $\dot\wedge$, $\dot\wedge_{[}$, $\dot\wedge_{]}$, and $\dot\wedge_{<}\ \phi\ \dot\wedge_{>}$ operators are linear temporal logic operators and that the execution trace σ is akin to a linearly ordered Kripke structure.

Definition 8. *Semantics of Event Order Logic (EOL). Let $T = (S, Act, \rightarrow, I, AP, L)$ a transition system, let ϕ, ϕ_1, ϕ_2 simple EOL formulas, let ψ, ψ_1, ψ_2 complex EOL formulas, and let $\mathcal{A}$ a set of event variables, with $a_{\alpha_i} \in \mathcal{A}$, over which ϕ, ϕ_1, ϕ_2 are built. Let $\sigma = s_0, \alpha_1, s_1, \alpha_2, \dots \alpha_n, s_n$ a finite execution trace of T and $\sigma[i..r] = s_i, \alpha_{i+1}, s_{i+1}, \alpha_{i+2}, \dots \alpha_r, s_r$ a partial trace. We define that an execution trace σ satisfies a formula ψ, written as $\sigma \models_e \psi$, as follows:*

$$s_j \models_e a_{\alpha_i} \text{ iff } s_{j-1} \xrightarrow{\alpha_i} s_j$$

$$s_j \models_e \neg\phi \text{ iff not } s_j \models_e \phi$$

$$\sigma[i..r] \models_e \phi \text{ iff } \exists j : i \le j \le r \,.\, s_j \models_e \phi$$

$$\sigma[i..r] \models_e \neg\phi \text{ iff } \forall j : i \le j \le r \,.\, s_j \models_e \neg\phi$$

$$\sigma \models_e \psi \text{ iff } \sigma[0..n] \models_e \psi, \textit{ where } n \textit{ is the length of } \sigma.$$

$$\sigma[i..r] \models_e \phi_1 \wedge \phi_2 \text{ iff } \sigma[i..r] \models_e \phi_1 \textit{ and } \sigma[i..r] \models_e \phi_2$$

$$\sigma[i..r] \models_e \phi_1 \vee \phi_2 \text{ iff } \sigma[i..r] \models_e \phi_1 \textit{ or } \sigma[i..r] \models_e \phi_2$$

$$\sigma[i..r] \models_e \neg(\phi_1 \wedge \phi_2) \text{ iff } \sigma[i..r] \models_e \neg\phi_1 \textit{ and } \sigma[i..r] \models_e \neg\phi_2$$

$$\sigma[i..r] \models_e \neg(\phi_1 \vee \phi_2) \text{ iff } \sigma[i..r] \models_e \neg\phi_1 \textit{ and } \sigma[i..r] \models_e \neg\phi_2$$

$$\sigma[i..r] \models_e \psi_1 \wedge \psi_2 \text{ iff } \sigma[i..r] \models_e \psi_1 \textit{ and } \sigma[i..r] \models_e \psi_2$$

$$\sigma[i..r] \models_e \psi_1 \vee \psi_2 \text{ iff } \sigma[i..r] \models_e \psi_1 \textit{ or } \sigma[i..r] \models_e \psi_2$$

$$\sigma[i..r] \models_e \psi_1 \dot\wedge \psi_2 \text{ iff } \exists j, k : i \le j < k \le r \,.\, \sigma[i..j] \models_e \psi_1 \textit{ and } \sigma[k..r] \models_e \psi_2$$

$$\sigma[i..r] \models_e \psi \dot\wedge_{[} \phi \text{ iff } (\exists j : i \le j \le r \,.\, \sigma[j..j] \models_e \psi \textit{ and } (\forall k : j \le k \le r \,.\, \sigma[k..k] \models_e \phi))$$

$$\sigma[i..r] \models_e \psi_1 \dot\wedge \psi_2 \dot\wedge_{[} \phi \text{ iff } \sigma[i..r] \models_e \psi_1 \dot\wedge \psi_2 \textit{ and } \sigma[i..r] \models_e \psi_2 \dot\wedge_{[} \phi$$

$$\sigma[i..r] \models_e (\psi_1 \wedge \psi_2) \dot\wedge_{[} \phi \text{ iff } \sigma[i..r] \models_e \psi_1 \dot\wedge \psi_2 \dot\wedge_{[} \phi \textit{ or } \sigma[i..r] \models_e \psi_2 \dot\wedge \psi_1 \dot\wedge_{[} \phi$$

$$\sigma[i..r] \models_e (\psi_1 \vee \psi_2) \dot\wedge_{[} \phi \text{ iff } \sigma[i..r] \models_e \psi_1 \dot\wedge_{[} \phi \textit{ or } \sigma[i..r] \models_e \psi_2 \dot\wedge_{[} \phi$$

$$\sigma[i..r] \models_e \phi \dot\wedge_{]} \psi \text{ iff } (\exists j : i \le j \le r \,.\, \sigma[j..j] \models_e \psi \textit{ and } (\forall k : 0 \le k \le j \,.\, \sigma[k..k] \models_e \phi))$$

$$\sigma[i..r] \models_e \phi \dot\wedge_{]} \psi_1 \dot\wedge \psi_2 \text{ iff } \sigma[i..r] \models_e \phi \dot\wedge_{]} \psi_1 \textit{ and } \sigma[i..r] \models_e \psi_1 \dot\wedge \psi_2$$

$$\sigma[i..r] \models_e \phi \dot\wedge_{]} (\psi_1 \wedge \psi_2) \text{ iff } \sigma[i..r] \models_e \phi \dot\wedge_{]} \psi_1 \dot\wedge \psi_2 \textit{ or } \sigma[i..r] \models_e \phi \dot\wedge_{]} \psi_2 \dot\wedge \psi_1$$

$$\sigma[i..r] \models_e \phi \dot\wedge_{]} (\psi_1 \vee \psi_2) \text{ iff } \sigma[i..r] \models_e \phi \dot\wedge_{]} \psi_1 \textit{ or } \sigma[i..r] \models_e \phi \dot\wedge_{]} \psi_2$$

$$\sigma[i..r] \models_e \psi_1 \dot\wedge_{<} \phi \dot\wedge_{>} \psi_2 \text{ iff } (\exists j, k : i \le j < k \le r \,.\, \sigma[j..j] \models_e \psi_1 \textit{ and } \sigma[k..r] \models_e \psi_2 \textit{ and } (\forall l : j \le l \le k \,.\, \sigma[l..l] \models_e \phi))$$

$$\sigma[i..r] \models_e \psi_1 \dot\wedge \psi_2 \dot\wedge_{<} \phi \dot\wedge_{>} \psi_3 \text{ iff } \sigma[i..r] \models_e \psi_1 \dot\wedge \psi_2 \textit{ and } \sigma[i..r] \models_e \psi_2 \dot\wedge_{<} \phi \dot\wedge_{>} \psi_3$$

$$\sigma[i..r] \models_e \psi_1 \dot\wedge_{<} \phi \dot\wedge_{>} \psi_2 \dot\wedge \psi_3 \text{ iff } \sigma[i..r] \models_e \psi_1 \dot\wedge_{<} \phi \dot\wedge_{>} \psi_2 \textit{ and } \sigma[i..r] \models_e \psi_2 \dot\wedge \psi_3$$

$$\sigma[i..r] \models_e (\psi_1 \wedge \psi_2) \dot\wedge_{<} \phi \dot\wedge_{>} \psi_3 \text{ iff } \sigma[i..r] \models_e \psi_1 \dot\wedge \psi_2 \dot\wedge_{<} \phi \dot\wedge_{>} \psi_3 \textit{ or } \sigma[i..r] \models_e \psi_2 \dot\wedge \psi_1 \dot\wedge_{<} \phi \dot\wedge_{>} \psi_3$$

$$\sigma[i..r] \models_e (\psi_1 \vee \psi_2) \dot\wedge_{<} \phi \dot\wedge_{>} \psi_3 \text{ iff } \sigma[i..r] \models_e \psi_1 \dot\wedge_{<} \phi \dot\wedge_{>} \psi_3 \textit{ or } \sigma[i..r] \models_e \psi_2 \dot\wedge_{<} \phi \dot\wedge_{>} \psi_3$$

$$\sigma[i..r] \models_e \psi_1 \dot{\wedge}_< \phi \dot{\wedge}_> (\psi_2 \wedge \psi_3) \text{ iff } \sigma[i..r] \models_e \psi_1 \dot{\wedge}_< \phi \dot{\wedge}_> \psi_2 \dot{\wedge} \psi_3$$
$$\quad \textit{or } \sigma[i..r] \models_e \psi_1 \dot{\wedge}_< \phi \dot{\wedge}_> \psi_3 \dot{\wedge} \psi_2$$
$$\sigma[i..r] \models_e \psi_1 \dot{\wedge}_< \phi \dot{\wedge}_> (\psi_2 \vee \psi_3) \text{ iff } \sigma[i..r] \models_e \psi_1 \dot{\wedge}_< \phi \dot{\wedge}_> \psi_2$$
$$\quad \textit{or } \sigma[i..r] \models_e \psi_1 \dot{\wedge}_< \phi \dot{\wedge}_> \psi_3$$

We define that the transition system T satisfies the formula ψ, written as $T \models_e \psi$, iff $\exists \sigma \in T \,.\, \sigma \models_e \psi$.

We will now show that there are EOL formulas that are different with respect to syntax but equivalent with respect to the semantics and consequently evaluate to the same truth-value under all interpretations. These equivalences can be used to rewrite EOL formulas, which we will later need in order to translate EOL formulas into LTL formulas.

Definition 9. *Equivalences of EOL formulas. Let $T = (S, Act, \rightarrow, I, AP, L)$ a transition system, let ϕ_1, ϕ_2 simple EOL formulas, let ψ_1, ψ_2, ψ_3 complex EOL formulas, and let $\mathcal{A}$ a set of event variables, with $a_{\alpha_i} \in \mathcal{A}$, over which ϕ, ϕ_1, ϕ_2 are built. Let $\sigma = s_0, \alpha_1, s_1, \alpha_2, \dots \alpha_n, s_n$ a finite execution trace of T and $\sigma[i..r] = s_i, \alpha_{i+1}, s_{i+1}, \alpha_{i+2}, \dots \alpha_r, s_r$ a partial trace. Two EOL formulas ψ_1 and ψ_2 are equivalent, denoted by $\psi_1 \equiv \psi_2$ iff $\sigma[i..r] \models_e \psi_1 \Leftrightarrow \sigma[i..r] \models_e \psi_2$.*

$$\begin{array}{ll}
\neg(\phi_1 \wedge \phi_2) & \equiv \neg\phi_1 \wedge \neg\phi_2 \\
\neg(\phi_1 \vee \phi_2) & \equiv \neg\phi_1 \wedge \neg\phi_2 \\
\neg(\phi_1 \wedge \phi_2) & \equiv \neg(\phi_1 \vee \phi_2) \\
\psi_1 \wedge \psi_2 & \equiv \psi_1 \dot{\wedge} \psi_2 \vee \psi_2 \dot{\wedge} \psi_1 \\
(\psi_1 \wedge \psi_2) \dot{\wedge} \psi_3 & \equiv \psi_1 \dot{\wedge} \psi_3 \wedge \psi_2 \dot{\wedge} \psi_3 \\
\psi_1 \dot{\wedge} (\psi_2 \wedge \psi_3) & \equiv \psi_1 \dot{\wedge} \psi_2 \wedge \psi_1 \dot{\wedge} \psi_3 \\
(\psi_1 \vee \psi_2) \dot{\wedge} \psi_3 & \equiv \psi_1 \dot{\wedge} \psi_3 \vee \psi_2 \dot{\wedge} \psi_3 \\
\psi_1 \dot{\wedge} (\psi_2 \vee \psi_3) & \equiv \psi_1 \dot{\wedge} \psi_2 \vee \psi_1 \dot{\wedge} \psi_3 \\
\psi_1 \dot{\wedge} \psi_2 \wedge \psi_1 \dot{\wedge} \psi_3 & \equiv \psi_1 \dot{\wedge} \psi_3 \wedge \psi_1 \dot{\wedge} \psi_2 \\
\psi_1 \dot{\wedge} \psi_2 \dot{\wedge} \psi_3 & \equiv \psi_1 \dot{\wedge} \psi_2 \wedge \psi_1 \dot{\wedge} \psi_3 \wedge \psi_2 \dot{\wedge} \psi_3 \\
\psi_1 \dot{\wedge}_[(\phi_1 \wedge \phi_2) & \equiv \psi_1 \dot{\wedge}_[\phi_1 \wedge \psi_1 \dot{\wedge}_[\phi_2 \\
\psi_1 \dot{\wedge}_[(\phi_1 \vee \phi_2) & \equiv \psi_1 \dot{\wedge}_[\phi_1 \vee \psi_1 \dot{\wedge}_[\phi_2 \\
(\phi_1 \wedge \phi_2) \dot{\wedge}_] \psi_1 & \equiv \phi_1 \dot{\wedge}_] \psi_1 \wedge \phi_2 \dot{\wedge}_] \psi_1 \\
(\phi_1 \vee \phi_2) \dot{\wedge}_] \psi_1 & \equiv \phi_1 \dot{\wedge}_] \psi_1 \vee \phi_2 \dot{\wedge}_] \psi_1 \\
\psi_1 \dot{\wedge}_< (\phi_1 \vee \phi_2) \dot{\wedge}_> \psi_2 & \equiv \psi_1 \dot{\wedge}_< \phi_1 \dot{\wedge}_> \psi_2 \vee \psi_1 \dot{\wedge}_< \phi_2 \dot{\wedge}_> \psi_2
\end{array}$$

Note that $\neg(\phi_1 \wedge \phi_2) \not\equiv (\neg\phi_1 \vee \neg\phi_2)$. An example of two equivalent formulas from the railroad crossing example from Section 2.2 are $\psi_1 = \text{Ta} \dot{\wedge} \text{Tc} \wedge \text{Ca} \dot{\wedge} \text{Cc}$ and $\psi_2 = \text{Ca} \dot{\wedge} \text{Cc} \wedge \text{Ta} \dot{\wedge} \text{Tc}$ which both state that the event train approaching (Ta) happens before the train crossing (Tc) event and the car approaching (Ca) event happens before the car crossing (Cc) event without imposing a restriction on the order of, for instance, the events train approaching and car approaching. Another example is $\psi_3 = (\neg\text{Gc} \wedge \neg\text{Gf}) \dot{\wedge}_] \text{Ca}$ and $\psi_4 = \neg\text{Gc} \dot{\wedge}_] \text{Ca} \wedge \neg\text{Gf} \dot{\wedge}_] \text{Ca}$ which both state that before the car approaching event neither the gate closing event (Gc) nor the gate failed (Gf) event occurs.

We will now prove the equivalences of EOL formulas from Def. 9.

Theorem 1. $\neg(\phi_1 \wedge \phi_2) \equiv \neg\phi_1 \wedge \neg\phi_2$

Proof. $\neg(\phi_1 \wedge \phi_2) \equiv \neg\phi_1 \wedge \neg\phi_2$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \neg(\phi_1 \wedge \phi_2) \Leftrightarrow \sigma \vDash_e \neg\phi_1 \wedge \neg\phi_2$.

$$\begin{aligned} \sigma[i..r] \vDash_e \neg(\phi_1 \wedge \phi_2) &\Leftrightarrow \sigma[i..r] \vDash_e \neg\phi_1 \text{ and } \sigma[i..r] \vDash_e \neg\phi_2 \\ &\Leftrightarrow \sigma[i..r] \vDash_e \neg\phi_1 \wedge \neg\phi_2 \end{aligned}$$

□

Theorem 2. $\neg(\phi_1 \vee \phi_2) \equiv \neg\phi_1 \wedge \neg\phi_2$

Proof. $\neg(\phi_1 \vee \phi_2) \equiv \neg\phi_1 \wedge \neg\phi_2$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \neg(\phi_1 \vee \phi_2) \Leftrightarrow \sigma \vDash_e \neg\phi_1 \wedge \neg\phi_2$.

$$\begin{aligned} \sigma[i..r] \vDash_e \neg(\phi_1 \vee \phi_2) &\Leftrightarrow \sigma[i..r] \vDash_e \neg\phi_1 \text{ and } \sigma[i..r] \vDash_e \neg\phi_2 \\ &\Leftrightarrow \sigma[i..r] \vDash_e \neg\phi_1 \wedge \neg\phi_2 \end{aligned}$$

□

Theorem 3. $\neg(\phi_1 \wedge \phi_2) \equiv \neg(\phi_1 \vee \phi_2)$

Proof. $\neg(\phi_1 \wedge \phi_2) \equiv \neg(\phi_1 \vee \phi_2)$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \neg(\phi_1 \wedge \phi_2) \Leftrightarrow \sigma \vDash_e \neg(\phi_1 \vee \phi_2)$.

$$\begin{aligned} \sigma[i..r] \vDash_e \neg(\phi_1 \wedge \phi_2) &\Leftrightarrow \sigma[i..r] \vDash_e \neg\phi_1 \text{ and } \sigma[i..r] \vDash_e \neg\phi_2 \\ &\Leftrightarrow \sigma[i..r] \vDash_e \neg\phi_1 \wedge \neg\phi_2 \\ &\Leftrightarrow \sigma[i..r] \vDash_e \neg(\phi_1 \vee \phi_2) \end{aligned}$$

□

Theorem 4. $\psi_1 \wedge \psi_2 \equiv \psi_1 \dot\wedge \psi_2 \vee \psi_2 \dot\wedge \psi_1$

Proof. $\psi_1 \wedge \psi_2 \equiv \psi_1 \dot\wedge \psi_2 \vee \psi_2 \dot\wedge \psi_1$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_1 \wedge \psi_2 \Leftrightarrow \sigma \vDash_e \psi_1 \dot\wedge \psi_2 \vee \psi_2 \dot\wedge \psi_1$.

$$\begin{aligned} \sigma[i..r] \vDash_e \psi_1 \wedge \psi_2 &\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \text{ and } \sigma[i..r] \vDash_e \psi_2 \\ &\Leftrightarrow \exists j : i \leq j \leq r \;.\; \sigma[i..j] \vDash_e \psi_1 \text{ or } \sigma[j..r] \vDash_e \psi_1 \\ &\qquad \text{and } \sigma[i..j] \vDash_e \psi_2 \text{ or } \sigma[j..r] \vDash_e \psi_2 \\ &\Leftrightarrow \exists j : i \leq j \leq r \;.\; \sigma[i..j] \vDash_e \psi_1 \text{ and } \sigma[j..r] \vDash_e \psi_2 \\ &\qquad \text{or } \sigma[i..j] \vDash_e \psi_2 \text{ and } \sigma[j..r] \vDash_e \psi_1 \\ &\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \dot\wedge \psi_2 \vee \psi_2 \dot\wedge \psi_1 \end{aligned}$$

□

Theorem 5. $(\psi_1 \wedge \psi_2) \wedge \psi_3 \equiv \psi_1 \wedge \psi_3 \wedge \psi_2 \wedge \psi_3$

Proof. $(\psi_1 \wedge \psi_2) \wedge \psi_3 \equiv \psi_1 \wedge \psi_3 \wedge \psi_2 \wedge \psi_3$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e (\psi_1 \wedge \psi_2) \wedge \psi_3 \Leftrightarrow \sigma \vDash_e \psi_1 \wedge \psi_3 \wedge \psi_2 \wedge \psi_3$.

$$\begin{aligned}
\sigma[i..r] \vDash_e (\psi_1 \wedge \psi_2) \wedge \psi_3 &\Leftrightarrow \exists j,k : i \le j < k \le r \,.\, \sigma[i..j] \vDash_e (\psi_1 \wedge \psi_2) \text{ and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \exists j,k : i \le j < k \le r \,.\, \sigma[i..j] \vDash_e \psi_1 \text{ and } \sigma[i..j] \vDash_e \psi_2 \\
&\quad\; \text{and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \exists j,l,k : i \le j < k \le r \text{ and } i \le l < k \le r \,.\, \sigma[i..j] \vDash_e \psi_1 \\
&\quad\; \text{and } \sigma[i..l] \vDash_e \psi_2 \text{ and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \exists j,l,k : i \le j < k \le r \text{ and } i \le l < k \le r \,.\, \sigma[i..j] \vDash_e \psi_1 \\
&\quad\; \text{and } \sigma[k..r] \vDash_e \psi_3 \text{ and } \sigma[i..l] \vDash_e \psi_2 \; \text{ and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \wedge \psi_3 \wedge \psi_2 \wedge \psi_3
\end{aligned}$$

□

Theorem 6. $\psi_1 \wedge (\psi_2 \wedge \psi_3) \equiv \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3$

Proof. $\psi_1 \wedge (\psi_2 \wedge \psi_3) \equiv \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_1 \wedge (\psi_2 \wedge \psi_3) \Leftrightarrow \sigma \vDash_e \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3$.

$$\begin{aligned}
\sigma[i..r] \vDash_e \psi_1 \wedge (\psi_2 \wedge \psi_3) &\Leftrightarrow \exists j,k : i \le j < k \le r \,.\, \sigma[i..j] \vDash_e \psi_1 \text{ and } \sigma[k..r] \vDash_e (\psi_2 \wedge \psi_3) \\
&\Leftrightarrow \exists j,k : i \le j < k \le r \,.\, \sigma[i..j] \vDash_e \psi_1 \text{ and } \sigma[k..r] \vDash_e \psi_2 \\
&\quad\; \text{and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \exists j,k : i \le j < k \le r \;\; \sigma[i..j] \vDash_e \psi_1 \text{ and } \sigma[k..r] \vDash_e \psi_2 \\
&\quad\; \text{and } \sigma[i..j] \vDash_e \psi_1 \; \text{ and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3
\end{aligned}$$

□

Theorem 7. $(\psi_1 \vee \psi_2) \wedge \psi_3 \equiv \psi_1 \wedge \psi_3 \vee \psi_2 \wedge \psi_3$

Proof. $(\psi_1 \vee \psi_2) \wedge \psi_3 \equiv \psi_1 \wedge \psi_3 \vee \psi_2 \wedge \psi_3$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e (\psi_1 \vee \psi_2) \wedge \psi_3 \Leftrightarrow \sigma \vDash_e \psi_1 \wedge \psi_3 \vee \psi_2 \wedge \psi_3$.

$$\begin{aligned}
\sigma[i..r] \vDash_e (\psi_1 \vee \psi_2) \wedge \psi_3 &\Leftrightarrow \exists j,k : i \le j < k \le r \,.\, \sigma[i..j] \vDash_e (\psi_1 \vee \psi_2) \text{ and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \exists j,k : i \le j < k \le r \,.\, (\sigma[i..j] \vDash_e \psi_1 \text{ or } \sigma[i..j] \vDash_e \psi_2) \\
&\quad\; \text{and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \exists j,l,k : i \le j < k \le r \text{ and } i \le l < k \le r \,.\, (\sigma[i..j] \vDash_e \psi_1 \\
&\quad\; \text{or } \sigma[i..l] \vDash_e \psi_2) \; \text{ and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \exists j,l,k : i \le j < k \le r \text{ and } i \le l < k \le r \,.\, (\sigma[i..j] \vDash_e \psi_1 \\
&\quad\; \text{and } \sigma[k..r] \vDash_e \psi_3) \text{ or } (\sigma[i..l] \vDash_e \psi_2 \; \text{ and } \sigma[k..r] \vDash_e \psi_3) \\
&\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \wedge \psi_3 \vee \psi_2 \wedge \psi_3
\end{aligned}$$

□

Theorem 8. $\psi_1 \wedge (\psi_2 \vee \psi_3) \equiv \psi_1 \wedge \psi_2 \vee \psi_1 \wedge \psi_3$

Proof. $\psi_1 \wedge (\psi_2 \vee \psi_3) \equiv \psi_1 \wedge \psi_2 \vee \psi_1 \wedge \psi_3$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_1 \wedge (\psi_2 \vee \psi_3) \Leftrightarrow \sigma \vDash_e \psi_1 \wedge \psi_2 \vee \psi_1 \wedge \psi_3$.

$$\begin{aligned}
\sigma[i..r] \vDash_e \psi_1 \wedge (\psi_2 \vee \psi_3) &\Leftrightarrow \exists j,k : i \leq j < k \leq r \;.\; \sigma[i..j] \vDash_e \psi_1 \text{ and } \sigma[k..r] \vDash_e (\psi_2 \vee \psi_3) \\
&\Leftrightarrow \exists j,k : i \leq j < k \leq r \;.\; \sigma[i..j] \vDash_e \psi_1 \text{ and } (\sigma[k..r] \vDash_e \psi_2 \\
&\quad \text{ or } \sigma[k..r] \vDash_e \psi_3) \\
&\Leftrightarrow \exists j,k : i \leq j < k \leq r \;\; \sigma[i..j] \vDash_e \psi_1 \text{ and } \sigma[k..r] \vDash_e \psi_2 \\
&\quad \text{ or } \sigma[i..j] \vDash_e \psi_1 \text{ and } \sigma[k..r] \vDash_e \psi_3 \\
&\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \wedge \psi_2 \vee \psi_1 \wedge \psi_3
\end{aligned}$$

□

Theorem 9. $\psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3 \equiv \psi_1 \wedge \psi_3 \wedge \psi_1 \wedge \psi_2$

Proof. $\psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3 \equiv \psi_1 \wedge \psi_3 \wedge \psi_1 \wedge \psi_2$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3 \Leftrightarrow \sigma \vDash_e \psi_1 \wedge \psi_3 \wedge \psi_1 \wedge \psi_2$.

$$\begin{aligned}
\sigma[i..r] \vDash_e \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3 &\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \wedge \psi_2 \text{ and } \sigma[i..r] \vDash_e \psi_1 \wedge \psi_3 \\
&\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \wedge \psi_3 \wedge \psi_1 \wedge \psi_2
\end{aligned}$$

□

Theorem 10. $\psi_1 \wedge \psi_2 \wedge \psi_3 \equiv \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3 \wedge \psi_2 \wedge \psi_3$

Proof. $\psi_1 \wedge \psi_2 \wedge \psi_3 \equiv \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3 \wedge \psi_2 \wedge \psi_3$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_1 \wedge \psi_2 \wedge \psi_3 \Leftrightarrow \sigma \vDash_e \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3 \wedge \psi_2 \wedge \psi_3$.

$$\begin{aligned}
\sigma[i..r] \vDash_e \psi_1 \wedge \psi_2 \wedge \psi_3 &\Leftrightarrow \exists j,k,l : i \leq j < k < l \leq r \;.\; \sigma[i..j] \vDash_e \psi_1 \\
&\quad \text{and } \sigma[k..l-1] \vDash_e \psi_2 \text{ and } \sigma[l..r] \vDash_e \psi_3 \\
&\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \wedge \psi_2 \wedge \psi_1 \wedge \psi_3 \wedge \psi_2 \wedge \psi_3
\end{aligned}$$

□

Theorem 11. $\psi_1 \wedge_{[} (\phi_1 \wedge \phi_2) \equiv \psi_1 \wedge_{[} \phi_1 \wedge \psi_1 \wedge_{[} \phi_2$

Proof. $\psi_1 \wedge_{[} (\phi_1 \wedge \phi_2) \equiv \psi_1 \wedge_{[} \phi_1 \wedge \psi_1 \wedge_{[} \phi_2$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_1 \wedge_{[} (\phi_1 \wedge \phi_2) \Leftrightarrow \sigma \vDash_e \psi_1 \wedge_{[} \phi_1 \wedge \psi_1 \wedge_{[} \phi_2$.

$$
\begin{aligned}
\sigma[i..r] \vDash_e \psi_1 \dot{\wedge}_{[} (\phi_1 \wedge \phi_2) &\Leftrightarrow (\exists j : i \leq j \leq r \;.\; \sigma[j..j] \vDash_e \psi_1 \text{ and} \\
&\quad\;\, (\forall k : j \leq k \leq r \;.\; \sigma[k..k] \vDash_e (\phi_1 \wedge \phi_2))) \\
&\Leftrightarrow (\exists j : i \leq j \leq r \;.\; \sigma[j..j] \vDash_e \psi_1 \text{ and} \\
&\quad\;\, (\forall k : j \leq k \leq r \;.\; \sigma[k..k] \vDash_e \phi_1 \text{ and } \sigma[k..k] \vDash_e \phi_2)) \\
&\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \dot{\wedge}_{[} \phi_1 \wedge \psi_1 \dot{\wedge}_{[} \phi_2
\end{aligned}
$$

In theory $\sigma[k..k] \vDash_e (\phi_1 \wedge \phi_2)$ is not possible since only one event is allowed per state s_k but this form is needed for $\sigma[k..k] \vDash_e \neg(\phi_1 \wedge \phi_2)$ □

Theorem 12. $\psi_1 \dot{\wedge}_{[} (\phi_1 \vee \phi_2) \equiv \psi_1 \dot{\wedge}_{[} \phi_1 \vee \psi_1 \dot{\wedge}_{[} \phi_2$

Proof. $\psi_1 \dot{\wedge}_{[} (\phi_1 \vee \phi_2) \equiv \psi_1 \dot{\wedge}_{[} \phi_1 \vee \psi_1 \dot{\wedge}_{[} \phi_2$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_1 \dot{\wedge}_{[} (\phi_1 \vee \phi_2) \Leftrightarrow \sigma \vDash_e \psi_1 \dot{\wedge}_{[} \phi_1 \vee \psi_1 \dot{\wedge}_{[} \phi_2$.

$$
\begin{aligned}
\sigma[i..r] \vDash_e \psi_1 \dot{\wedge}_{[} (\phi_1 \vee \phi_2) &\Leftrightarrow (\exists j : i \leq j \leq r \;.\; \sigma[j..j] \vDash_e \psi_1 \text{ and} \\
&\quad\;\, (\forall k : j \leq k \leq r \;.\; \sigma[k..k] \vDash_e (\phi_1 \vee \phi_2))) \\
&\Leftrightarrow (\exists j : i \leq j \leq r \;.\; \sigma[j..j] \vDash_e \psi_1 \text{ and} \\
&\quad\;\, (\forall k : j \leq k \leq r \;.\; \sigma[k..k] \vDash_e \phi_1 \text{ or } \sigma[k..k] \vDash_e \phi_2)) \\
&\Leftrightarrow \sigma[i..r] \vDash_e \psi_1 \dot{\wedge}_{[} \phi_1 \vee \psi_1 \dot{\wedge}_{[} \phi_2
\end{aligned}
$$

□

Theorem 13. $(\phi_1 \wedge \phi_2) \dot{\wedge}_{]} \psi_1 \equiv \phi_1 \dot{\wedge}_{]} \psi_1 \wedge \phi_2 \dot{\wedge}_{]} \psi_1$

Proof. $(\phi_1 \wedge \phi_2) \dot{\wedge}_{]} \psi_1 \equiv \phi_1 \dot{\wedge}_{]} \psi_1 \wedge \phi_2 \dot{\wedge}_{]} \psi_1$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e (\phi_1 \wedge \phi_2) \dot{\wedge}_{]} \psi_1 \Leftrightarrow \sigma \vDash_e \phi_1 \dot{\wedge}_{]} \psi_1 \wedge \phi_2 \dot{\wedge}_{]} \psi_1$.

$$
\begin{aligned}
\sigma[i..r] \vDash_e (\phi_1 \wedge \phi_2) \dot{\wedge}_{]} \psi_1 &\Leftrightarrow (\exists j : i \leq j \leq r \;.\; \sigma[j..j] \vDash_e \psi_1 \text{ and} \\
&\quad\;\, (\forall k : 0 \leq k \leq j \;.\; \sigma[k..k] \vDash_e (\phi_1 \wedge \phi_2))) \\
&\Leftrightarrow (\exists j : i \leq j \leq r \;.\; \sigma[j..j] \vDash_e \psi_1 \text{ and} \\
&\quad\;\, (\forall k : 0 \leq k \leq j \;.\; \sigma[k..k] \vDash_e \phi_1 \text{ and } \sigma[k..k] \vDash_e \phi_2)) \\
&\Leftrightarrow \sigma[i..r] \vDash_e \phi_1 \dot{\wedge}_{]} \psi_1 \wedge \phi_2 \dot{\wedge}_{]} \psi_1
\end{aligned}
$$

In theory $\sigma[k..k] \vDash_e (\phi_1 \wedge \phi_2)$ is not possible since only one event is allowed per state s_k but this form is needed for $\sigma[k..k] \vDash_e \neg(\phi_1 \wedge \phi_2)$ □

Theorem 14. $(\phi_1 \vee \phi_2) \dot{\wedge}_{]} \psi_1 \equiv \phi_1 \dot{\wedge}_{]} \psi_1 \vee \phi_2 \dot{\wedge}_{]} \psi_1$

Proof. $(\phi_1 \vee \phi_2) \dot{\wedge}_{]} \psi_1 \equiv \phi_1 \dot{\wedge}_{]} \psi_1 \vee \phi_2 \dot{\wedge}_{]} \psi_1$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e (\phi_1 \vee \phi_2) \dot{\wedge}_{]} \psi_1 \Leftrightarrow \sigma \vDash_e \phi_1 \dot{\wedge}_{]} \psi_1 \vee \phi_2 \dot{\wedge}_{]} \psi_1$.

$$\begin{aligned}\sigma[i..r] \vDash_e (\phi_1 \vee \phi_2) \curlywedge_{]} \psi_1 &\Leftrightarrow (\exists j : i \leq j \leq r \,.\, \sigma[j..j] \vDash_e \psi_1 \text{ and} \\ &\quad\; (\forall k : 0 \leq k \leq j \,.\, \sigma[k..k] \vDash_e (\phi_1 \vee \phi_2))) \\ &\Leftrightarrow (\exists j : i \leq j \leq r \,.\, \sigma[j..j] \vDash_e \psi_1 \text{ and} \\ &\quad\; (\forall k : 0 \leq k \leq j \,.\, \sigma[k..k] \vDash_e \phi_1 \text{ or } \sigma[k..k] \vDash_e \phi_2)) \\ &\Leftrightarrow \sigma[i..r] \vDash_e \phi_1 \curlywedge_{]} \psi_1 \vee \phi_2 \curlywedge_{]} \psi_1\end{aligned}$$

□

Theorem 15. $\psi_1 \curlywedge_{<} (\phi_1 \vee \phi_2) \curlywedge_{>} \psi_2 \equiv \psi_1 \curlywedge_{<} \phi_1 \curlywedge_{>} \psi_2 \vee \psi_1 \curlywedge_{<} \phi_2 \curlywedge_{>} \psi_2$

Proof. $\psi_1 \curlywedge_{<} (\phi_1 \vee \phi_2) \curlywedge_{>} \psi_2 \equiv \psi_1 \curlywedge_{<} \phi_1 \curlywedge_{>} \psi_2 \vee \psi_1 \curlywedge_{<} \phi_2 \curlywedge_{>} \psi_2$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_1 \curlywedge_{<} (\phi_1 \vee \phi_2) \curlywedge_{>} \psi_2 \Leftrightarrow \sigma \vDash_e \psi_1 \curlywedge_{<} \phi_1 \curlywedge_{>} \psi_2 \vee \psi_1 \curlywedge_{<} \phi_2 \curlywedge_{>} \psi_2$.

$$\begin{aligned}\sigma[i..r] \vDash_e \psi_1 \curlywedge_{<} (\phi_1 \vee \phi_2) \curlywedge_{>} \psi_2 &\Leftrightarrow (\exists j,k : i \leq j < k \leq r \,.\, \sigma[j..j] \vDash_e \psi_1 \\ &\quad\; \text{and } \sigma[k..k] \vDash_e \psi_2 \text{ and } (\forall l : j \leq l \leq k \,. \\ &\quad\; \sigma[l..l] \vDash_e (\phi_1 \vee \phi_2))) \\ &\Leftrightarrow (\exists j,k : i \leq j < k \leq r \,.\, \sigma[j..j] \vDash_e \psi_1 \\ &\quad\; \text{and } \sigma[k..k] \vDash_e \psi_2 \text{ and } (\forall l : j \leq l \leq k \,. \\ &\quad\; \sigma[l..l] \vDash_e \phi_1 \text{ or } \sigma[l..l] \vDash_e \phi_2)) \\ &\Leftrightarrow \psi_1 \curlywedge_{<} \phi_1 \curlywedge_{>} \psi_2 \vee \psi_1 \curlywedge_{<} \phi_2 \curlywedge_{>} \psi_2\end{aligned}$$

□

In order to be later able to define translation rules from EOL to LTL we will now define the event order normal form (EONF) which permits the unordered *and*- ($\wedge$) and *or*-operator ($\vee$) only if they are not bound by any ordered operator and in addition permits the unordered *and*-operator ($\wedge$) in the scope of the between operators $\curlywedge_{<}$ and $\curlywedge_{>}$.

Definition 10. *Event Order Normal Form (EONF) The set of EOL formulas over a set $\mathcal{A}$ of event variables in event order normal form (EONF) is given by:*

$$\phi ::= a \mid \neg\phi$$

$$\phi_\wedge ::= \phi \mid \phi_\wedge \mid \neg\phi_\wedge \mid \phi_{\wedge_1} \wedge \phi_{\wedge_2}$$

$$\psi ::= \phi \mid \phi_1 \curlywedge \phi_2 \mid \phi \curlywedge_{[} \phi \mid \phi \curlywedge_{]} \phi \mid \phi_1 \curlywedge_{<} \phi_\wedge \curlywedge_{>} \phi_2$$

$$\psi_\wedge ::= \psi \mid \psi_\wedge \mid \psi_{\wedge_1} \wedge \psi_{\wedge_2} \mid \psi_{\wedge_1} \vee \psi_{\wedge_2}$$

where $a \in \mathcal{A}$ and ϕ, ϕ_1, ϕ_2 are simple EOL formulas in EONF and $\phi_\wedge$, $\phi_{\wedge_1}$ and $\phi_{\wedge_2}$ are simple EOL formulas containing the $\wedge$-operator in EONF and $\psi_\wedge$, $\psi_{\wedge_1}$ and $\psi_{\wedge_2}$ are complex EOL formulas containing the $\wedge$-operator and / or $\vee$-operator in EONF.

An EOL formula can be transformed into an equivalent EOL formula in EONF by rewriting using the equivalence rules from Def. 9. For instance, the EOL formula $\psi = \text{Ta} \wedgedot \text{Gc} \wedgedot \text{Tc}$ can be rewritten in EONF as $\psi' = \text{Ta} \wedgedot \text{Gc} \wedge \text{Gc} \wedgedot \text{Tc} \wedge \text{Ta} \wedgedot \text{Tc}$.

Theorem 16. *For each EOL formula there exists a semantically equivalent EOL formula in EONF.*

Proof. Each of the following transformations allows to rewrite a EOL formula that is not in EONF into an semantically equivalent EOL formula which is in EONF.

$$
\begin{aligned}
&\text{EONF}((\phi_1 \wedge \phi_2) \wedgedot \phi_3) &&= \phi_1 \wedgedot \phi_3 \wedge \phi_2 \wedgedot \phi_3\\
&\text{EONF}((\phi_1 \vee \phi_2) \wedgedot \phi_3) &&= \phi_1 \wedgedot \phi_3 \vee \phi_2 \wedgedot \phi_3\\
&\text{EONF}(\phi_1 \wedgedot (\phi_2 \wedge \phi_3)) &&= \phi_1 \wedgedot \phi_2 \wedge \phi_1 \wedgedot \phi_3\\
&\text{EONF}(\phi_1 \wedgedot (\phi_2 \vee \phi_3)) &&= \phi_1 \wedgedot \phi_2 \vee \phi_1 \wedgedot \phi_3\\
&\text{EONF}(\phi_1 \wedgedot \phi_2 \wedgedot \phi_3) &&= \phi_1 \wedgedot \phi_2 \wedge \phi_1 \wedgedot \phi_3 \wedge \phi_2 \wedgedot \phi_3\\
&\text{EONF}((\phi_1 \wedge \phi_2) \wedgedot_{]} \phi_3) &&= \phi_1 \wedgedot_{]} \phi_3 \wedge \phi_2 \wedgedot_{]} \phi_3\\
&\text{EONF}((\phi_1 \vee \phi_2) \wedgedot_{]} \phi_3) &&= \phi_1 \wedgedot_{]} \phi_3 \vee \phi_2 \wedgedot_{]} \phi_3\\
&\text{EONF}(\phi_1 \wedgedot_{[} (\phi_2 \wedge \phi_3)) &&= \phi_1 \wedgedot_{[} \phi_2 \wedge \phi_1 \wedgedot_{[} \phi_3\\
&\text{EONF}(\phi_1 \wedgedot_{[} (\phi_2 \vee \phi_3)) &&= \phi_1 \wedgedot_{[} \phi_2 \vee \phi_1 \wedgedot_{[} \phi_3\\
&\text{EONF}(\phi_1 \wedgedot_{<} (\phi_2 \vee \phi_3) \wedgedot_{>} \phi_4) &&= \phi_1 \wedgedot_{<} \phi_2 \wedgedot_{>} \phi_4 \vee \phi_1 \wedgedot_{<} \phi_3 \wedgedot_{>} \phi_4
\end{aligned}
$$

□

4 Event Order Logic and Linear Temporal Logic

In this section we will show that the EOL is a sub-set of the LTL and hence, it is possible to translate each EOL formula into an equivalent LTL formula. An EOL formula and an LTL formula are equivalent if they are satisfied by the same set of execution traces. More formally we define

Definition 11. *An EOL formula ψ and an LTL formula φ are equivalent denoted by $\psi \equiv \varphi$ if any transition system T and all traces σ in T: $\sigma \vDash_e \psi \Leftrightarrow \sigma \vDash_l \varphi$.*

We will now define translation rules that can be used to translate a EOL formula in EONF into a equivalent LTL formula.

Definition 12. *LTL formula for an EOL formula. Let ψ an EOL formula that is built over the set of event variables $a \in \mathcal{A}$ and is in EONF. The states in T are labeled with an atomic proposition indicating whether the event represented by the event variable a leads to this states. More formally, if $s_{j-1} \xrightarrow{\alpha_i} s_j$ then $a_{\alpha_i} \in L(s_j)$. The equivalent EOL formula for an EOL formula ψ can be constructed as follows:*

If ψ does contain one of the ordered operators $\wedgedot$, $\wedgedot_{[}$, $\wedgedot_{]}$, or $\wedgedot_{<}...\wedgedot_{>}$ the translation function $LTL_{\wedgedot}(\psi)$ is used, $LTL(\psi)$ else. The translation functions $LTL_{\wedgedot}(\psi)$ and $LTL(\psi)$ are applied recursively over the structure of ψ.

$$
\begin{array}{ll}
LTL(a_{\alpha_i}) & = \Diamond a_{\alpha_i} \\
LTL(\neg a_{\alpha_i}) & = \Box \neg a_{\alpha_i} \\
LTL(\phi_1 \wedge \phi_2) & = LTL(\phi_1) \wedge LTL(\phi_2) \\
LTL(\neg(\phi_1 \wedge \phi_2)) & = LTL(\neg \phi_1) \wedge LTL(\neg \phi_2) \\
LTL_{\curlywedge}(a) & = a \\
LTL_{\curlywedge}(\neg a) & = \neg a \\
LTL_{\curlywedge}(\phi_1 \wedge \phi_2) & = LTL_{\curlywedge}(\phi_1) \wedge LTL_{\curlywedge}(\phi_2) \\
LTL_{\curlywedge}(\neg(\phi_1 \wedge \phi_2)) & = LTL_{\curlywedge}(\neg \phi_1) \wedge LTL_{\curlywedge}(\neg \phi_2) \\
LTL_{\curlywedge}(\psi_{\wedge_1} \wedge \psi_{\wedge_2}) & = LTL_{\curlywedge}(\psi_{\wedge_1}) \wedge LTL_{\curlywedge}(\psi_{\wedge_2}) \\
LTL_{\curlywedge}(\psi_{\wedge_1} \vee \psi_{\wedge_2}) & = LTL_{\curlywedge}(\psi_{\wedge_1}) \vee LTL_{\curlywedge}(\psi_{\wedge_2}) \\
LTL_{\curlywedge}(\phi_1 \curlywedge \phi_2) & = \Diamond(LTL_{\curlywedge}(\phi_1) \wedge \Diamond LTL_{\curlywedge}(\phi_2)) \\
LTL_{\curlywedge}(\phi_1 \curlywedge_{[} \phi_2) & = \Diamond(LTL_{\curlywedge}(\phi_1) \wedge \Box LTL_{\curlywedge}(\phi_2)) \\
LTL_{\curlywedge}(\phi_1 \curlywedge_{]} \phi_2) & = LTL_{\curlywedge}(\phi_1)\, \mathcal{U}\, LTL_{\curlywedge}(\phi_2) \\
LTL_{\curlywedge}(\phi_1 \curlywedge_{<} \phi_{\wedge} \curlywedge_{>} \phi_2) & = \Diamond(LTL_{\curlywedge}(\phi_1) \wedge (LTL_{\curlywedge}(\phi_{\wedge})\, \mathcal{U}\, LTL_{\curlywedge}(\phi_2))
\end{array}
$$

where a_{α_i} is an Event variable and the corresponding atomic proposition with which the state is labeled, ϕ, ϕ_1, ϕ_2 are simple EOL formulas in EONF, ϕ_{wedge} is a simple EOL formula in EONF containing the $\wedge$-operator, and $\psi_{\wedge}$, $\psi_{\wedge_1}$ and $\psi_{\wedge_2}$ are complex EOL formulas containing the $\wedge$-operator and / or $\vee$-operator in EONF.

It remains to show that the translation rules defined in Def. 12 are equivalent with respect to Def. 11.

Theorem 17. $s_j \vDash_e a_{\alpha_i} \equiv s_j \vDash_l a_{\alpha_i}$

Proof. $s_j \vDash_e a_{\alpha_i} \equiv s_j \vDash_l a_{\alpha_i}$ holds if for any transition system T and all states s in T: $s \vDash_e a_{\alpha_i} \Leftrightarrow s \vDash_l a_{\alpha_i}$.

$$
\begin{array}{c}
s_j \vDash_e a_{\alpha_i} \quad \Leftrightarrow \quad s_j \vDash_l a_{\alpha_i} \\
s_j \vDash_e a_{\alpha_i} \textit{ iff } s_{j-1} \xrightarrow{\alpha_i} s_j \Leftrightarrow s_j \vDash_l a_{\alpha_i} \text{ iff } a_{\alpha_i} \in L(s_j) \\
\text{Per definition } a_{\alpha_i} \in L(s_j) \text{ holds if } s_{j-1} \xrightarrow{\alpha_i} s_j.
\end{array}
$$

□

Theorem 18. $s_j \vDash_e \neg a_{\alpha_i} \equiv s_j \vDash_l \neg a_{\alpha_i}$

Proof. $s_j \vDash_e \neg a_{\alpha_i} \equiv s_j \vDash_l \neg a_{\alpha_i}$ holds if for any transition system T and all states s in T: $s \vDash_e \neg a_{\alpha_i} \Leftrightarrow s \vDash_l \neg a_{\alpha_i}$.

$$
s_j \vDash_e \neg a_{\alpha_i} \quad \Leftrightarrow \quad s_j \vDash_l \neg a_{\alpha_i}
$$

$$
s_j \vDash_e \neg a_{\alpha_i} \textit{ iff not } s_{j-1} \xrightarrow{\alpha_i} s_j \Leftrightarrow s_j \vDash_l \neg a_{\alpha_i} \text{ iff not } a_{\alpha_i} \in L(s_j)
$$

□

Theorem 19. $\sigma \vDash_e a_{\alpha_i} \equiv \sigma \vDash_l \Diamond a_{\alpha_i}$

Proof. $\sigma \vDash_e a_{\alpha_i} \equiv \sigma \vDash_l \Diamond a_{\alpha_i}$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e a_{\alpha_i} \Leftrightarrow \sigma \vDash_l \Diamond a_{\alpha_i}$.

$$\sigma \vDash_e a_{\alpha_i} \quad \Leftrightarrow \quad \sigma \vDash_l \Diamond a_{\alpha_i}$$

$$\sigma \vDash_e a_{\alpha_i} \textit{ iff } \exists j : 0 \leq j \leq n \;.\; s_j \vDash_e a_{\alpha_i} \Leftrightarrow \begin{array}{c} \sigma \vDash_l \text{ true } \mathcal{U}\, a_{\alpha_i} \text{ iff } \exists k \geq 0 \;.\; \sigma[k...] \vDash_l a_{\alpha_i} \\ \text{and } \forall j : 0 \leq j < k \;.\; \sigma[j...] \vDash_l \text{ true} \end{array}$$

$$\Leftrightarrow \sigma \vDash_l \text{ true } \mathcal{U}\, a_{\alpha_i} \text{ iff } \exists k \geq 0 \;.\; \sigma[k...] \vDash_l a_{\alpha_i}$$

□

Theorem 20. $\sigma \vDash_e \neg a_{\alpha_i} \equiv \sigma \vDash_l \Box \neg a_{\alpha_i}$

Proof. $\sigma \vDash_e \neg a_{\alpha_i} \equiv \sigma \vDash_l \Box \neg a_{\alpha_i}$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \neg a_{\alpha_i} \Leftrightarrow \sigma \vDash_l \Box \neg a_{\alpha_i}$.

$$\sigma \vDash_e \neg a_{\alpha_i} \quad \Leftrightarrow \quad \sigma \vDash_l \Box \neg a_{\alpha_i}$$

$$\begin{array}{c} \sigma \vDash_e \neg a_{\alpha_i} \textit{ iff} \\ \forall j : 0 \leq j \leq n \;.\; s_j \vDash_e \neg a_{\alpha_i} \end{array} \Leftrightarrow \begin{array}{c} \sigma \vDash_l \neg(\text{ true } \mathcal{U} \neg\neg a_{\alpha_i}) \text{ iff not} \\ \exists k \geq 0 \;.\; \sigma[k...] \vDash_l \neg\neg a_{\alpha_i} \end{array}$$

$$\Leftrightarrow \begin{array}{c} \sigma \vDash_l \neg(\text{ true } \mathcal{U}\, a_{\alpha_i}) \text{ iff not} \\ \exists k \geq 0 \;.\; \sigma[k...] \vDash_l a_{\alpha_i} \end{array}$$

$$\Leftrightarrow \begin{array}{c} \sigma \vDash_l \neg(\text{ true } \mathcal{U}\, a_{\alpha_i}) \text{ iff} \\ \forall k \geq 0 \;.\; \sigma[k...] \vDash_l \neg a_{\alpha_i} \end{array}$$

□

Theorem 21. $\sigma \vDash_e \phi_1 \land \phi_2 \equiv \sigma \vDash_l LTL(\phi_1) \land LTL(\phi_2)$

Proof. $\sigma \vDash_e \phi_1 \land \phi_2 \equiv \sigma \vDash_l \text{LTL}(\phi_1) \land \text{LTL}(\phi_2)$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \phi_1 \land \phi_2 \Leftrightarrow \sigma \vDash_l \text{LTL}(\phi_1) \land \text{LTL}(\phi_2)$

$$\sigma \vDash_e \phi_1 \land \phi_2 \quad \Leftrightarrow \quad \sigma \vDash_l \text{LTL}(\phi_1) \land \text{LTL}(\phi_2)$$

$$\begin{array}{c} \sigma \vDash_e \phi_1 \land \phi_2 \textit{ iff} \\ \sigma \vDash_e \phi_1 \text{ and } \sigma \vDash_e \phi_2 \end{array} \Leftrightarrow \begin{array}{c} \sigma \vDash_l \text{LTL}(\phi_1) \land \text{LTL}(\phi_2) \text{ iff} \\ \sigma \vDash_l \text{LTL}(\phi_1) \text{ and } \sigma \vDash_l \text{LTL}(\phi_2) \end{array}$$

□

Theorem 22. $\sigma \vDash_e \neg(\phi_1 \land \phi_2) \equiv \sigma \vDash_l LTL(\neg\phi_1) \land LTL(\neg\phi_2)$

Proof. $\sigma \vDash_e \neg(\phi_1 \land \phi_2) \equiv \sigma \vDash_l \text{LTL}(\neg\phi_1) \land \text{LTL}(\neg\phi_2)$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \neg(\phi_1 \land \phi_2) \Leftrightarrow \sigma \vDash_l \text{LTL}(\neg\phi_1) \land \text{LTL}(\neg\phi_2)$

$$\sigma \vDash_e \neg(\phi_1 \land \phi_2) \quad \Leftrightarrow \quad \sigma \vDash_l \text{LTL}(\neg\phi_1) \land \text{LTL}(\neg\phi_2)$$

$$\begin{array}{c} \sigma \vDash_e \neg(\phi_1 \land \phi_2) \textit{ iff} \\ \sigma \vDash_e \neg\phi_1 \text{ and } \sigma \vDash_e \neg\phi_2 \end{array} \Leftrightarrow \begin{array}{c} \sigma \vDash_l \text{LTL}(\neg\phi_1) \land \text{LTL}(\neg\phi_2) \text{ iff} \\ \sigma \vDash_l \text{LTL}(\neg\phi_1) \text{ and } \text{LTL}(\neg\phi_2) \end{array}$$

□

Theorem 23. $\sigma \vDash_e \phi_1 \land \phi_2 \equiv \sigma \vDash_l LTL_{\land}(\phi_1) \land LTL_{\land}(\phi_2)$

Proof. $\sigma \vDash_e \phi_1 \wedge \phi_2 \equiv \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \mathrm{LTL}_{\curlywedge}(\phi_2)$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \phi_1 \wedge \phi_2 \Leftrightarrow \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \mathrm{LTL}_{\curlywedge}(\phi_2)$. If $\mathrm{LTL}_{\curlywedge}$ is applied $\phi_1 \wedge \phi_2$ can only occur in the formula $\phi_{1'} \curlywedge_< (\phi_1 \wedge \phi_2) \curlywedge_> \phi_{2'}$ and consequently according to the semantics of EOL has to hold on some $\sigma[l..l] \equiv s_l$.

$$s_l \vDash_e \phi_1 \wedge \phi_2 \quad \Leftrightarrow \quad s_l \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \mathrm{LTL}_{\curlywedge}(\phi_2)$$

$$\begin{array}{c} s_l \vDash_e \phi_1 \wedge \phi_2 \textit{ iff} \\ s_l \vDash_e \phi_1 \text{ and } s_l \vDash_e \phi_2 \end{array} \Leftrightarrow \begin{array}{c} s_l \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \mathrm{LTL}_{\curlywedge}(\phi_2) \text{ iff} \\ s_l \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_1) \text{ and } s_l \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_2) \end{array}$$

□

Theorem 24. $\sigma \vDash_e \neg(\phi_1 \wedge \phi_2) \equiv \sigma \vDash_l LTL_{\curlywedge}(\neg\phi_1) \wedge LTL_{\curlywedge}(\neg\phi_2)$

Proof. $\sigma \vDash_e \neg(\phi_1 \wedge \phi_2) \equiv \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\neg\phi_1) \wedge \mathrm{LTL}_{\curlywedge}(\neg\phi_2)$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \neg(\phi_1 \wedge \phi_2) \Leftrightarrow \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\neg\phi_1) \wedge \mathrm{LTL}_{\curlywedge}(\neg\phi_2)$. If $\mathrm{LTL}_{\curlywedge}$ is applied $\neg(\phi_1 \wedge \phi_2)$ can only occur in the formula $\phi_{1'} \curlywedge_< \neg(\phi_1 \wedge \phi_2) \curlywedge_> \phi_{2'}$ and consequently according to the semantics of EOL has to hold on some $\sigma[l..l] \equiv s_l$.

$$s_l \vDash_e \neg(\phi_1 \wedge \phi_2) \quad \Leftrightarrow \quad s_l \vDash_l \mathrm{LTL}_{\curlywedge}(\neg\phi_1) \wedge \mathrm{LTL}_{\curlywedge}(\neg\phi_2)$$

$$\begin{array}{c} s_l \vDash_e \neg(\phi_1 \wedge \phi_2) \textit{ iff} \\ s_l \vDash_e \neg\phi_1 \text{ and } s_l \vDash_e \neg\phi_2 \end{array} \Leftrightarrow \begin{array}{c} s_l \vDash_l \mathrm{LTL}_{\curlywedge}(\neg\phi_1) \wedge \mathrm{LTL}_{\curlywedge}(\neg\phi_2) \text{ iff} \\ s_l \vDash_l \mathrm{LTL}_{\curlywedge}(\neg\phi_1) \text{ and } \mathrm{LTL}_{\curlywedge}(\neg\phi_2) \end{array}$$

□

Theorem 25. $\sigma \vDash_e \psi_{\wedge_1} \wedge \psi_{\wedge 2} \equiv \sigma \vDash_l LTL_{\curlywedge}(\psi_{\wedge_1}) \wedge LTL_{\curlywedge}(\psi_{\wedge 2})$

Proof. $\sigma \vDash_e \psi_{\wedge_1} \wedge \psi_{\wedge 2} \equiv \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \wedge \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2})$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_{\wedge_1} \wedge \psi_{\wedge 2} \Leftrightarrow \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \wedge \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2})$

$$\sigma \vDash_e \psi_{\wedge_1} \wedge \psi_{\wedge 2} \quad \Leftrightarrow \quad \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \wedge \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2})$$

$$\begin{array}{c} \sigma \vDash_e \psi_{\wedge_1} \wedge \psi_{\wedge 2} \textit{ iff} \\ \sigma \vDash_e \psi_{\wedge_1} \text{ and } \sigma \vDash_e \psi_{\wedge 2} \end{array} \Leftrightarrow \begin{array}{c} \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \wedge \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2}) \text{ iff} \\ \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \text{ and } \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2}) \end{array}$$

□

Theorem 26. $\sigma \vDash_e \psi_{\wedge_1} \vee \psi_{\wedge 2} \equiv \sigma \vDash_l LTL_{\curlywedge}(\psi_{\wedge_1}) \vee LTL_{\curlywedge}(\psi_{\wedge 2})$

Proof. $\sigma \vDash_e \psi_{\wedge_1} \vee \psi_{\wedge 2} \equiv \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \vee \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2})$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \psi_{\wedge_1} \vee \psi_{\wedge 2} \Leftrightarrow \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \vee \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2})$

$$\sigma \vDash_e \psi_{\wedge_1} \vee \psi_{\wedge 2} \quad \Leftrightarrow \quad \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \vee \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2})$$

$$\begin{array}{c} \sigma \vDash_e \psi_{\wedge_1} \vee \psi_{\wedge 2} \textit{ iff} \\ \sigma \vDash_e \psi_{\wedge_1} \text{ or } \sigma \vDash_e \psi_{\wedge 2} \end{array} \Leftrightarrow \begin{array}{c} \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \vee \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2}) \text{ iff} \\ \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge_1}) \text{ or } \sigma \vDash_l \mathrm{LTL}_{\curlywedge}(\psi_{\wedge 2}) \end{array}$$

□

Theorem 27. $\sigma \vDash_e \phi_1 \curlywedge \phi_2 \equiv \Diamond(LTL_{\curlywedge}(\phi_1) \wedge \Diamond LTL_{\curlywedge}(\phi_2))$

Proof. $\sigma \vDash_e \phi_1 \curlywedge \phi_2 \equiv \Diamond(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \Diamond\mathrm{LTL}_{\curlywedge}(\phi_2))$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \phi_1 \curlywedge \phi_2 \Leftrightarrow \Diamond(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \Diamond\mathrm{LTL}_{\curlywedge}(\phi_2))$

$$\sigma \vDash_e \phi_1 \curlywedge \phi_2 \quad \Leftrightarrow \quad \Diamond(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \Diamond \mathrm{LTL}_{\curlywedge}(\phi_2))$$

$$\begin{array}{c} \sigma \vDash_e \phi_1 \curlywedge \phi_2 \textit{ iff} \\ \exists j,k : 0 \le j < k \le n \,.\, \sigma[0..j] \vDash_e \phi_1 \\ \text{and } \sigma[k..n] \vDash_e \phi_2 \end{array} \Leftrightarrow \begin{array}{c} \sigma \vDash_l \text{ true } \mathcal{U} \,(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \\ \text{true } \mathcal{U} \,(\mathrm{LTL}_{\curlywedge}(\phi_2))) \text{ iff} \\ \exists j \ge 0 \,. \\ \sigma[j...] \vDash_l (\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \\ \text{true } \mathcal{U} \,(\mathrm{LTL}_{\curlywedge}(\phi_2))) \end{array}$$

$$\Leftrightarrow \begin{array}{c} \sigma \vDash_l \text{ true } \mathcal{U} \,(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \\ \text{true } \mathcal{U} \,(\mathrm{LTL}_{\curlywedge}(\phi_2))) \text{ iff} \\ \exists j \ge 0 \,.\, \sigma[j...] \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_1) \\ \text{and} \\ \exists k \ge j \,.\, \sigma[k...] \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_2) \end{array}$$

□

Theorem 28. $\sigma \vDash_e \phi_1 \curlywedge_{[} \phi_2 \equiv \sigma \vDash_l \Diamond(LTL_{\curlywedge}(\phi_1) \wedge \Box LTL_{\curlywedge}(\phi_2))$

Proof. $\sigma \vDash_e \phi_1 \curlywedge_{[} \phi_2 \equiv \sigma \vDash_l \Diamond(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \Box \mathrm{LTL}_{\curlywedge}(\phi_2))$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \phi_1 \curlywedge_{[} \phi_2 \Leftrightarrow \sigma \vDash_l \Diamond(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \Box \mathrm{LTL}_{\curlywedge}(\phi_2))$

$$\sigma \vDash_e \phi_1 \curlywedge_{[} \phi_2 \quad \Leftrightarrow \quad \sigma \vDash_l \Diamond(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \Box \mathrm{LTL}_{\curlywedge}(\phi_2))$$

$$\begin{array}{c} \sigma \vDash_e \phi_1 \curlywedge_{[} \phi_2 \text{ iff} \\ \exists j : 0 \le j \le n \,.\, \sigma[j..j] \vDash_e \phi_1 \text{ and} \\ \forall k : j \le k \le n \,.\, \sigma[k..k] \vDash_e \phi_2 \end{array} \Leftrightarrow \begin{array}{c} \sigma \vDash_l \Diamond(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \Box \mathrm{LTL}_{\curlywedge}(\phi_2)) \text{ iff} \\ \text{true } \mathcal{U}(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge \neg(\text{ true } \mathcal{U} \neg \mathrm{LTL}_{\curlywedge}(\phi_2))) \end{array}$$

$$\Leftrightarrow \quad \begin{array}{c} \exists j : j \ge 0 \,.\, \sigma[j..] \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_1) \text{ and} \\ \forall k : k \ge j \,.\, \sigma[k..] \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_2) \end{array}$$

□

Theorem 29. $\sigma \vDash_e \phi_1 \curlywedge_{]} \phi_2 \equiv LTL_{\curlywedge}(\phi_1) \,\mathcal{U}\, LTL_{\curlywedge}(\phi_2)$

Proof. $\sigma \vDash_e \phi_1 \curlywedge_{]} \phi_2 \equiv \mathrm{LTL}_{\curlywedge}(\phi_1) \,\mathcal{U}\, \mathrm{LTL}_{\curlywedge}(\phi_2)$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \phi_1 \curlywedge_{]} \phi_2 \Leftrightarrow \mathrm{LTL}_{\curlywedge}(\phi_1) \,\mathcal{U}\, \mathrm{LTL}_{\curlywedge}(\phi_2)$

$$\sigma \vDash_e \phi_1 \curlywedge_{]} \phi_2 \quad \Leftrightarrow \quad \mathrm{LTL}_{\curlywedge}(\phi_1) \,\mathcal{U}\, \mathrm{LTL}_{\curlywedge}(\phi_2)$$

$$\begin{array}{c} \sigma \vDash_e \phi_1 \curlywedge_{]} \phi_2 \text{ iff} \\ \exists j : i \le j \le r \,.\, \sigma[j..j] \vDash_e \psi \text{ and} \\ \forall k : 0 \le k \le j \,.\, \sigma[k..k] \vDash_e \phi_1 \end{array} \Leftrightarrow \begin{array}{c} \mathrm{LTL}_{\curlywedge}(\phi_1) \,\mathcal{U}\, \mathrm{LTL}_{\curlywedge}(\phi_2) \text{ iff} \\ \exists j : j \ge 0 \,.\, \sigma[j..] \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_2) \text{ and} \\ \forall k : 0 \le k \le j \,.\, \sigma[k..] \vDash_l \mathrm{LTL}_{\curlywedge}(\phi_1) \end{array}$$

□

Theorem 30. $\sigma \vDash_e \phi_1 \curlywedge_{<} \phi_{\wedge} \curlywedge_{>} \phi_2 \equiv \sigma \vDash_l \Diamond(LTL_{\curlywedge}(\phi_1) \wedge (LTL_{\curlywedge}(\phi_{\wedge}) \,\mathcal{U}\, LTL_{\curlywedge}(\phi_2)))$

Proof. $\sigma \vDash_e \phi_1 \curlywedge_{<} \phi_{\wedge} \curlywedge_{>} \phi_2 \equiv \sigma \vDash_l \Diamond(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge (\mathrm{LTL}_{\curlywedge}(\phi_{\wedge}) \,\mathcal{U}\, \mathrm{LTL}_{\curlywedge}(\phi_2)))$ holds if for any transition system T and all traces σ in T: $\sigma \vDash_e \phi_1 \curlywedge_{<} \phi_{\wedge} \curlywedge_{>} \phi_2 \Leftrightarrow \sigma \vDash_l \Diamond(\mathrm{LTL}_{\curlywedge}(\phi_1) \wedge (\mathrm{LTL}_{\curlywedge}(\phi_{\wedge}) \,\mathcal{U}\, \mathrm{LTL}_{\curlywedge}(\phi_2)))$

$$\sigma \vDash_e \phi_1 \dot{\wedge}_< \phi_\wedge \dot{\wedge}_> \phi_2 \quad \Leftrightarrow \quad \begin{array}{c} \sigma \vDash_l \Diamond(\mathrm{LTL}_{\dot{\wedge}}(\phi_1) \wedge \\ (\mathrm{LTL}_{\dot{\wedge}}(\phi_\wedge)\,\mathcal{U}\,\mathrm{LTL}_{\dot{\wedge}}(\phi_2))) \end{array}$$

$$\begin{array}{c} \sigma \vDash_e \phi_1 \dot{\wedge}_< \phi_\wedge \dot{\wedge}_> \phi_2 \text{ iff} \\ \exists j, k : i \leq j < k \leq r \,.\, \sigma[j..j] \vDash_e \phi_1 \\ \text{and } \sigma[k..r] \vDash_e \phi_2 \\ \text{and } \forall l : j \leq l \leq k \,.\, \sigma[l..l] \vDash_e \phi \end{array} \Leftrightarrow \begin{array}{c} \sigma \vDash_l \Diamond(\mathrm{LTL}_{\dot{\wedge}}(\phi_1) \wedge \\ (\mathrm{LTL}_{\dot{\wedge}}(\phi_\wedge)\,\mathcal{U}\,\mathrm{LTL}_{\dot{\wedge}}(\phi_2))) \text{ iff} \\ \exists j : j \leq 0 \,.\, \sigma[j..] \vDash_l \mathrm{LTL}_{\dot{\wedge}}(\phi_1) \text{ and} \\ \exists k : k > j \,.\, \sigma[k..] \vDash_l \mathrm{LTL}_{\dot{\wedge}}(\phi_2) \text{ and} \\ \forall l : j \leq l \leq k \,.\, \sigma[l..] \vDash_l \mathrm{LTL}_{\dot{\wedge}}(\phi) \end{array}$$

□

We have shown for all translation rules that the LTL formula generated by the translation rule is equivalent, with respect to Def. 9, to the EOL formula from which it was generated.

Theorem 31. *It follows from Definition 12 that EOL is a sub-set of LTL and each EOL formula can be translated into an equivalent LTL formula.*

5 Conclusion

In this paper we have refined the semantics of the event order logic and have shown how EOL formulas can be transformed into semantically equivalent EOL formulas. Furthermore, we have defined the Event Order Logic Normal Form (EONF) and proven that each EOL formula can be rewritten in EONF. In addition we have shown that each EOL formula can be translated to an LTL formula that is satisfied for the same set of execution traces.

References

1. F. Leitner-Fischer and S. Leue, “On the synergy of probabilistic causality computation and causality checking,” in *Proceedings of International SPIN Symposium on Model Checking of Software. Stony Brook, NY, USA*, 2013.
2. ——, “Probabilistic fault tree synthesis using causality computation,” *International Journal of Critical Computer-Based Systems*, 2013.
3. Z. Manna and A. Pnueli, *The temporal logic of reactive and concurrent systems.* Springer-Verlag New York, Inc., 1992.
4. C. Baier and J.-P. Katoen, *Principles of Model Checking.* The MIT Press, 2008.