Decomposition of TrPTL formulas

Raphaël Meyer and Antoine Petit

LSV, URA 2236 CNRS, ENS de Cachan,
61, av. du Prés. Wilson, F-94235 Cachan Cedex

Abstract. Partial orders based verifications methods are now well developed. In this framework, several suitable logics have already been defined. We focus on this paper on the logic $TrPTL$, as defined by Thiagarajan, for which models are the well known (infinite) Mazurkiewicz traces. We study the case where the alphabet is not connected. Our main theoretical result is that any $TrPTL$ formula can be decomposed in an effective way as the disjunction of formulas on the connected components. Note that this result can be viewed as a direct logical counterpart of the famous Mezei's theorem on recognizable sets in a direct product of free monoids.
Finally, we show that our result can also be of practical interest. Precisely, we exhibit families of formulas for which the use of our decomposition procedure decreases the complexity of the decision procedure of satisfiability.

1 Introduction

The industrial and economic need of correct software has increased the interest on researches on specification and verification of sequential and distributed programs. In order to express properties of these programs, several logics have been defined, among which the famous Propositional linear time Temporal Logic (PTL) of Pnueli has to be mentioned [Pnu77]. These logics have been interpreted for a long time on infinite sequences describing the program behaviors. In the case of distributed programs, techniques allowing to verify a property for just one representative sequential behavior of each partially ordered computation is a subject of active research (see e.g. [KP92,GW94]).
An alternative way to treat the distributed programs is to represent theirs behaviors directly by partial order based models. Among the possible models, Mazurkiewicz traces [Maz77] play a central role. Indeed, the theory of traces is very well developed (see e.g [Die90,DR95]) and strongly related to other partial order based formalisms such as Petri nets or event structures. Moreover recognizable languages of infinite traces have been characterized from algebraic, automata and logical points of view [GP92,EM93,GPZ94], extending the classical theory of infinite words [Tho90].
In a natural way, several logics directly interpreted over traces or partial order based models have been proposed [Pen88,LRT92,MT92,PK95]. Unfortunately, these logics do not have natural automata counterparts. This motivated the

work of Thiagarajan to define the logic $TrPTL$ as a natural extension of PTL to be interpreted on infinite traces. Using automata techniques, the satisfiability problem for $TrPTL$ turns out to be decidable [Thi94,MT96]. A major open question on this logic is to know whether it is as expressive as the extension of the first-order logic $FO(<)$ to traces proposed by Thomas [Tho89] and studied also in [EM93].

We focus in this paper of the decomposition of $TrPTL$ formulas in the case where the underlying dependent alphabet is not connected. Our main theoretical result claims that any $TrPTL$ formula can be decomposed in an effective way as the disjunction of formulas on the connected components. Note that this result can be seen as a direct logical counterpart of the famous Mezei's theorem [Ber79] on recognizable sets in a direct product of free monoids. However, it is not a consequence since, as recalled above, the equivalence between $TrPTL$ and first-order logic $FO(<)$ is still an open problem.

Finally, we show that our result can also be of practical interest. Precisely, we exhibit families of formulas for which the use of our decomposition procedure decreases the complexity of the decision procedure of satisfiability. Note that, obviously, we can not expect to decrease time in the worst case with any such decomposition.

The paper is organized as follows. In Section 2, we briefly recall the basis on dependence graphs and $TrPTL$ logic. We give our main decomposition theorem in Section 3. In order to prove this theorem, we introduce a new operator which can be viewed as a "reset" of the configuration. When dealing with initial equivalence of formulas, we prove that this operator does not increase the power of $TrPTL$. Section 4 is dedicated to the proof of the main theorem. Finally, we show in Section 5 how our theoretical result can be of practical interest for decreasing the complexity procedure of satisfiability in some suitable cases, and give our conclusion in Section 6.

2 TrPTL

We recall in this section the needed bases on traces and on the $TrPTL$ logic (see [DR95] and [Thi94,MT96] for more complete presentations on these subjects).

Let $P = \{1, \ldots, k\}$ be a set of processes, each process $i \in P$ having a local alphabet Σ_i of actions. We may have $\Sigma_i \cap \Sigma_j \neq \emptyset$, some actions involving different processes. Let $\Sigma = \bigcup_{i=1}^{k} \Sigma_i$ be the global alphabet of actions. Each action $a \in \Sigma$ can be mapped to the set $pr(a) = \{i \in P \mid a \in \Sigma_i\}$ of the process it involves. Upon the alphabet Σ, a reflexive and symmetrical relation $D \subseteq \Sigma \times \Sigma$, called the dependence relation, is defined in the natural following way: $D = \{(a,b) \in \Sigma \times \Sigma \mid pr(a) \cap pr(b) \neq \emptyset\}$. This relation expresses the fact that two actions which involve a common process can not be executed simultaneously, and are therefore dependent. The complementary relation $I = (\Sigma \times \Sigma) \setminus D$ is called the independence relation induced by D upon Σ.

2.1 Infinite traces

An infinite trace is a (equivalence class up to isomorphism of) dependence graph(s), that is to say a Σ-labelled partially-ordered graph $F = (E, \leq, \lambda)$ satisfying the following conditions:

- E is a countable set of vertices;
- $\leq$ is a partial order relation on E; we shall write $e \lessdot e'$ (covering relation) iff $e < e'$ and $\forall e''$, $e \leq e'' \leq e'$ implies $e = e''$ or $e'' = e'$;
- λ is an (labelling) application from E to Σ;
- $\forall e \in E$, $\downarrow e = \{e' \in E \mid e' \leq e\}$ is a finite set;
- $\forall e, e' \in E$, $(\lambda(e), \lambda(e')) \in D$ implies $e \leq e'$ or $e' \leq e$;
- $\forall e, e' \in E$, $e \lessdot e'$ implies $(\lambda(e), \lambda(e')) \in D$.

Let $F = (E, \leq, \lambda)$ be a dependence graph. The set E is called the set of events of F, and the partial order relation $\leq$ is called the causality relation ($e \leq e'$ meaning that event e occurs before event e' in F). Let $E_i = \{e \in E \mid \lambda(e) \in \Sigma_i\}$ be the set of events relative to process i. A configuration of a dependence graph F is a finite set of events which respects the causality relation. More formally, c is a configuration of F iff c is a finite subset of E such that $\downarrow c = c$, where $\downarrow c = \bigcup_{e \in c} \downarrow e$. Let C_F be the set of all configurations of F. The notion of configuration is a set-theoretical translation of the one of prefix. For instance, the null prefix of a trace corresponds to the $\emptyset$ configuration.
A transition relation between configurations of a dependence graph is defined in the following way:

$$\forall c, c' \in C_F,\ \forall e \in E,\ \ (c \overset{e}{\Longrightarrow}_F c') \Leftrightarrow (c' = c \cup \{e\}\ and\ e \notin c)$$

For every configuration c of F, one can define the i-view of c, denoted by $\downarrow^i(c)$: $\downarrow^i(c) = \downarrow(c \cap E_i)$. Obviously, $\downarrow^i(c)$ is also a configuration.

2.2 Syntax and semantics of TrPTL

In order to express and verify properties on traces, we shall use the $TrPTL$ logic defined by Thiagarajan [Thi94]. This logic $TrPTL$ is built up from a countable set $AP = \{p, q, \ldots\}$ of atomic propositions indexed by the processes, the boolean connectives $\vee$ and $\neg$, and two temporal operators O_i (which is local "next-time" operator), and U_i (which is a local "until" operator).
The syntax of $TrPTL$ is defined in the following way:

$$\Phi_{TrPTL} ::= p(i) \mid \neg\alpha \mid \alpha \vee \beta \mid O_i\alpha \mid \alpha U_i \beta \ where\ p \in AP\ and\ i \in P.$$

A model for $TrPTL$ is a pair $M = (F, V)$, where F is a dependence graph and V an evaluation function from C_F into $(\wp(AP))^k$, k being the number of processes. Intuitively, for any configuration c and any process i, $V(c)[i]$ is the set of atomic propositions verified by process i at configuration c. In order to keep the distributive aspect of the model, we assume that the following locality condition is verified:

$$c \Longrightarrow_F^e c' \wedge \lambda(e) = a \Rightarrow \forall i \notin pr(a)\ ,\ V(c)[i] = V(c')[i]$$

That is to say, when an action a is being executed, the values of atomic propositions concerning processes not involved in this action can not be modified.
Let $M = (F, V)$ be a model and c a configuration of this model. The satisfiability of a formula α of $TrPTL$ at c in model M, denoted by $c \models_M \alpha$, is defined inductively as follows:

- $c \models_M p(i)$ iff $p \in V(c)[i]$;
- $c \models_M \neg\alpha$ iff $c \not\models_M \alpha$;
- $c \models_M \alpha \vee \beta$ iff $c \models_M \alpha$ or $c \models_M \beta$;
- $c \models_M O_i\alpha$ iff $\exists e \in E_i \mid \downarrow e \models_M \alpha$ and $(c \cap E_i) \subset \downarrow e \cap E_i = (c \cap E_i) \cup \{e\}$;
- $c \models_M \alpha U_i \beta$ iff $\exists c' \in C_F \mid c \subseteq c'$ and $\downarrow^i(c') \models_M \beta$, and $\forall c'' \in C_F$, if $\downarrow^i(c) \subseteq \downarrow^i(c'') \subset \downarrow^i(c)$ then $\downarrow^i(c'') \models_M \alpha$.

The formula $p(i)$, read "p at i", is satisfied if p is part of the atomic propositions provided by V for process i in configuration c. We denote by $\alpha \wedge \beta$ the formula $\neg(\neg\alpha \vee \neg\beta)$. We shall also use $\top = p \vee \neg p$, and $\bot = \neg\top$. The O_i operator is a local "next time" operator. The semantics of $O_i\ \alpha$ is that there exists a next i-view and that, at this i-view, α is satisfied. The U_i operator is an "until" operator restricted to events of E_i.
A formula α is said *satisfiable* if there exists a model $M = (F, V)$, and a configuration $c \in C_F$ such that $c \models_M \alpha$. A formula α is said *root-satisfiable* if there exists a model M such that $\emptyset \models_M \alpha$.
The interest of the $TrPTL$ logic lies mainly on the following result due to Thiagarajan [Thi94].

Theorem 1. *The satisfiability problem for $TrPTL$ formulas is decidable. It can be decided in time $2^{O(max(m^2 log(m), n)m)}$, where n is the size of the formula, and m the number of different processes mentioned in it.*

3 Decomposition theorem

On this paper, we focus on the case where the dependence alphabet is not connected. In this case, the dependence alphabet (Σ, D) is the disjoint union of two dependence alphabets (Σ_A, D_A) and (Σ_B, D_B) that is to say $\Sigma = \Sigma_A \cup \Sigma_B$, $\Sigma_A \cap \Sigma_B = \emptyset$ and $D = D_A \cup D_B$. We will say in the sequel that Σ_A and Σ_B are *components* of Σ. A component Σ_A is said *connected* if the graph of (Σ_A, D_A) is connected. It is easy to show that for every process i, all the actions of Σ_i belong to the same component of Σ. We can thus define $P_A = \{i \in P \mid \Sigma_i \subseteq \Sigma_A\}$ and $P_B = \{i \in P \mid \Sigma_i \subseteq \Sigma_B\}$.
For any subset Q of P (and in particular for P_A and P_B), we can now define in a natural way a corresponding subset of $TrPTL$ formulas, using syntactical restrictions, in the following way:

Definition 2. Let $Q \subseteq P$. We denote by $TrPTL(Q)$ the least subset of $TrPTL$ formulas built from elementary formulas $p(i)$, the connectives $\neg$ and $\vee$, and the operators O_i and U_i with the constraint $i \in Q$.

Two formulas ϕ and ψ are said *equivalent*, denoted by $\phi \equiv \psi$, if for every model M and configuration c of that model, it holds: $c \models_M \phi$ iff $c \models_M \psi$. In a similar way, ϕ and ψ are said *initially equivalent*, denoted by $\phi \equiv_i \psi$, if for every model M, it holds: $\emptyset \models_M \phi$ iff $\emptyset \models_M \psi$.
Our main result is that every $TrPTL$ formula can be effectively decomposed into an initially equivalent boolean combination of formulas of $TrPTL(P_A)$ and formulas of $TrPTL(P_B)$. As a first step we study the case of the equivalence. To this purpose, we need to define a new temporal operator, denoted by R for *Reset*, which semantics is: $c \models_M R\phi$ iff $\emptyset \models_M \phi$. Intuitively, this Reset operator allows us to go back in time by getting rid of the current configuration and continuing the semantic interpretation from the initial configuration.
With the help on this new operator, we can define the following class of $TrPTL$ formulas:

Definition 3. A $TrPTL$ formula γ is called a separated formula if there exist formulas γ_A and γ'_A of $TrPTL(P_A)$, and formulas γ_B and γ'_B of $TrPTL(P_B)$ such that

$$\gamma = \gamma_A \wedge R\gamma'_A \wedge \gamma_B \wedge R\gamma'_B$$

We can now state the decomposition theorem related to equivalence of $TrPTL$ formulas.

Theorem 4. *Let (Σ, D) be a non connected dependence alphabet such that $\Sigma = \Sigma_A \cup \Sigma_B$, Σ_A and Σ_B being distinct components of Σ. Every $TrPTL$ formula γ on (Σ, D) is equivalent to a disjunction of separated formulas. Moreover, these formulas can effectively be constructed from γ.*

As a direct corollary of this result, using the fact that Reset operators disappear when dealing with initial equivalence, we obtain a decomposition theorem related to initial equivalence:

Theorem 5. *Let (Σ, D) be a non connected dependence alphabet such that $\Sigma = \Sigma_A \cup \Sigma_B$, Σ_A and Σ_B being distinct components of Σ. Every $TrPTL$ formula γ on (Σ, D) is initially equivalent to a disjunction of formulas $\gamma_A \wedge \gamma_B$, where γ_A belongs to $TrPTL(P_A)$, and γ_B belongs to $TrPTL(P_B)$.*

By an immediate induction, we get the following corollary when dealing with the connected components of the alphabet (Σ, D):

Corollary 6. *Let (Σ, D) be a non connected dependence alphabet such that $\Sigma = \bigcup_{j=1}^{l} \Sigma^j$, the Σ^j being the distinct connected components of Σ. Denote by P_j the set of processes i such that $\Sigma_i \subseteq \Sigma^j$. Every $TrPTL$ formula γ on (Σ, D) is initially equivalent to a disjunction of formulas $\gamma_1 \wedge \gamma_2 \wedge \ldots \wedge \gamma_l$, where for all j,γ_j belongs to $TrPTL(P_j)$.*

Note that this result can be seen as a direct logical counterpart of the famous Mezei's theorem [Ber79] on recognizable sets of product of free monoids since, as we recalled in the introduction, the equivalence between $TrPTL$ and first-order logic $FO(<)$ is still an open problem.

The following section is dedicated to the proof of Theorem 4.

4 Proofs

In order to prove Theorem 4, we will first focus on the distributivity of the temporal operators upon the boolean connectives. The following equivalences follow easily from the definitions of the temporal operators.

Proposition 7. *Let γ, ψ and χ be $TrPTL$ formulas, and let $i \in P$, the following equivalences hold:*

$$O_i(\gamma \vee \psi) \equiv O_i(\gamma) \vee O_i(\psi)$$

$$O_i(\gamma \wedge \psi) \equiv O_i(\gamma) \wedge O_i(\psi)$$

$$(\gamma \wedge \psi)\, U_i\, \chi \equiv (\gamma\, U_i\, \chi) \wedge (\psi\, U_i\, \chi)$$

$$\gamma\, U_i\, (\psi \vee \chi) \equiv (\gamma\, U_i\, \psi) \vee (\gamma\, U_i\, \chi)$$

Now we shall focus on the non-connected aspect of the alphabet Σ. Let $E_A = \{e \in E \mid \lambda(e) \in \Sigma_A\}$ and $E_B = \{e \in E \mid \lambda(e) \in \Sigma_B\}$. Every dependence graph $F = (E, \leq, \lambda)$ on (Σ, D) can thus be decomposed in two dependence graphs $F_A = (E_A, \leq, \lambda)$ and $F_B = (E_B, \leq, \lambda)$ on (Σ_A, D_A) and (Σ_B, D_B) respectively. For every configuration c of C_F, we define the configurations c_A and c_B induced on the two components, and then it holds: $c = c_A \cup c_B$, where the union is to be taken over graphs. Remark that c_A and c_B are configurations of F.

Proposition 8. *Let $i \in P_A$ and let c be a configuration of a model M, then:*

- $c \models_M p(i) \Leftrightarrow c_A \models_M p(i)$
- $c \models_M O_i\alpha \Leftrightarrow c_A \models_M O_i\alpha$
- $c \models_M \alpha U_i\beta \Leftrightarrow c_A \models_M \alpha U_i\beta$

Proof. It suffices to notice that since $i \in P_A$, we have $E_i \subseteq E_A$, and hence $c \cap E_i = c_A \cap E_i$. □

With the help of the previous proposition, we can extend the distributivity of the temporal operators to some particular cases:

Proposition 9. *Let $i \in P_A$, let α and β be $TrPTL$ formulas, let α_A and α_B be $TrPTL(P_A)$ formulas, and let α_B and β_B be $TrPTL(P_B)$ formulas. Then the following equivalences hold:*

1. $O_i\, \alpha_B \equiv O_i\, \top \wedge R\, \alpha_B$
2. $O_i\, (R\, \alpha) \equiv O_i\, \top \wedge R\, \alpha$

3. $\alpha\, U_i\, (\beta_A \wedge \beta_B \wedge R\,\beta) \equiv \alpha\, U_i\, \beta_A \;\wedge R\,(\beta_B \wedge \beta)$
4. $(\alpha_A \vee \alpha_B)\, U_i\, \beta_A \equiv (\alpha_A\, U_i\, \beta_A) \vee (R\,\alpha_B \;\wedge\; \top\, U_i\, \beta_A)$

Proof. We shall only focus on the proof of equivalence 4, which gives a good idea of what the other proofs could look like. Let c be a configuration of a model M. We use the following notation: $S_i(max(c \cap E_i))$ is the set of all E_i-events that are greater than or equal to the greatest E_i-event appearing in configuration c. Now, $c \models_M (\alpha_A \vee \alpha_B)\, U_i\, \beta_A$ iff there exists $e \in S_i(max(c \cap E_i))$, such that $\downarrow e \models_M \beta_A$, and for all $e' \in S_i(max(c \cap E_i))$, if $e' < e$ then $\downarrow e' \models_M \alpha_A \vee \alpha_B$. Since $\downarrow e' \subset E_A$, Proposition 8 shows that $\downarrow e' \models_M \alpha_A \vee \alpha_B$ iff $\downarrow e' \models_M \alpha_A$ or $\emptyset \models_M \alpha_B$, which is equivalent to $c \models_M R\,\alpha_B$. Thus, if $c \models_M R\,\alpha_B$ then we just need to check that $c \models_M \top\, U_i\, \beta_A$; otherwise, we have to check that $c \models_M \alpha_A\, U_i\, \beta_A$. This leads to the equivalence between $c \models_M (\alpha_A \vee \alpha_B)\, U_i\, \beta_A$ and $c \models_M (\alpha_A\, U_i\, \beta_A) \vee (R\,\alpha_B \;\wedge\; \top\, U_i\, \beta_A)$. □

We can now prove our main theorem, that is the decomposition of $TrPTL$ formulas:

Proof (of Theorem 4). We prove the theorem by induction on the length of the formula γ.

- If $|\,\gamma\,| = 1$, then $\gamma = p(i)$. We can assume, without loss of generality, that $i \in P_A$. Then it suffices to write γ as $p(i) \wedge R\,\top \wedge \top \wedge R\,\top$, since $R\,\top \equiv \top$.
- If $|\,\gamma\,| > 1$, several cases occur, depending on the nature of γ:
  1. If $\gamma = \alpha \vee \beta$, the conclusion is trivial: it suffices to use the induction hypothesis on α and β.
  2. If $\gamma = \neg\alpha$, we use the induction hypothesis, then we make intensive use of some easy combinatorial properties and of the fact that the operators $\neg$ and R commute to put things in the right form.
  3. If $\gamma = O_i\, \alpha$, we use the induction hypothesis, then we use the distributivity of O_i upon $\vee$ and $\wedge$, and the equivalences 1 and 2 of Proposition 9.
  4. If $\gamma = \alpha\, U_i\, \beta$, we can also assume that $i \in P_A$. Using the induction hypothesis, β is shown to be equivalent to a disjunction of separated formulas. Using the distributivity of U_i upon $\vee$, γ can be written as an equivalent disjunction of formulas like $\alpha\, U_i\, (\beta_A \wedge R\,\beta'_A \wedge \beta_B \wedge R\,\beta'_B)$. Denote by $\alpha\, U_i\, \beta_{sep}$ this last formula. Using equivalence 3 of Proposition 9, we show the equivalence between $\alpha\, U_i\, \beta_{sep}$ and $\alpha\, U_i\, \beta_A \;\wedge\; R\,(\beta'_A) \wedge R\,(\beta_B \wedge \beta'_B)$. Now, using the induction hypothesis on α, we have: $\alpha \equiv \bigvee_{k=1}^{d_\alpha} \alpha_{A,k} \wedge R\,\alpha'_{A,k} \wedge \alpha_{B,k} \wedge R\,\alpha'_{B,k}$. This can be written as $\alpha \equiv \bigwedge_{K \subseteq \{1,\dots,d_\alpha\}} (\alpha_{A,K} \vee \alpha_{B,K})$, where $\alpha_{A,K} = \bigvee_{k \in K} (\alpha_{A,k} \wedge R\,\alpha'_{A,k})$. Hence $\alpha\, U_i\, \beta_A$ is equivalent to $\bigwedge_{K \subseteq \{1,\dots,d_\alpha\}} ((\alpha_{A,K} \vee \alpha_{B,K})\, U_i\, \beta_A)$. Using equivalence 4 of Proposition 9, we show that $c \models_M (\alpha_{A,K} \vee \alpha_{B,K})\, U_i\, \beta_A$ iff $c \models_M ((\top\, U_i\, \beta_A) \wedge R\,\alpha_{B,k}) \vee \alpha_{A,K}\, U_i\, \beta_A$. By definition $\alpha_{A,K}$ is equal to $\bigvee_{k \in K} (\alpha_{A,k} \wedge R\,\alpha'_{A,k})$, which is equivalent to $\bigwedge_{L \subseteq K} (\alpha_{A,L} \vee R\,\alpha'_{A,L})$, where $\alpha_{A,L} = \bigvee_{l \in L} \alpha_{A,l}$ and $\alpha'_{A,L} = \bigvee_{l \notin L} \alpha'_{A,l}$, using the distributivity of I upon $\vee$. Thus, $\alpha_{A,K}\, U_i\, \beta_A$ is equivalent to $\bigwedge_{L \subseteq K} (\alpha_{A,L} \vee R\alpha'_{A,L}) U_i \beta_A$,

also equivalent to $\bigwedge_{L\subseteq K}((\alpha_{A,L}\, U_i\, \beta_A) \vee (\top\, U_i\, \beta_A \wedge R\alpha'_{A,L}))$. Replacing the previous results in $\alpha\, \overline{U}_i\, \beta_A$, one can rewrite this formula as a big boolean combination of separated formulas. But disjunctions of separated formulas are stable under negation, and hence under boolean combination. Thus $\alpha\, U_i\, \beta_A$, and then $\alpha\, U_i\, \beta$, can be written as equivalent disjunctions of separated formulas, which concludes the proof of the theorem. □

5 Application to verification of TrPTL formulas

The proof of Theorem 4 has shown that decomposing a $TrPTL$ formula can lead to combinatorial explosions, especially when dealing with negations and U_i operators. In order to avoid this explosion, we can define a subclass of $TrPTL$ formulas for which the separation procedure is actually efficient.
Recall that if a formula is of size n and involves m different process, then the satisfiability problem for this formula is in time $2^{O(max(m^2 log(m), n)m)}$.
Let Φ_s be the least set of $TrPTL$ formulas satisfying the following conditions:

1. $p(i) \in \Phi_s$ for all $p \in PA$ and all $i \in P$.
2. If $\alpha, \beta \in \Phi_s$, then $\alpha \vee \beta \in \Phi_s$.
3. If $\alpha \in \Phi_s$, then for all $i \in P$, $O_i\, \alpha \in \Phi_s$.
4. If $\alpha \in TrPTL(P_A)$, and $\beta \in \Phi_s$, then $\alpha\, U_i\, \beta \in \Phi_s$, for all $i \in P_A$.
5. If $\alpha \in TrPTL(P_B)$, and $\beta \in \Phi_s$, then $\alpha\, U_i\, \beta \in \Phi_s$, for all $i \in P_B$.

Note that the formulas $\neg p(i)$ have been excluded only for sake of simplicity, since they can be simulated by defining new atomic propositions.
The Φ_s formulas, called separable formulas, can be decomposed as a disjunction of separated formulas, without any risk of combinatorial explosion.

Proposition 10. *Let γ be a Φ_s formula of size n containing d disjunction operators, then γ can effectively be written as an equivalent conjuntion of d formulas of size at most $(n-d)$ not containing any $\vee$ operator.*

Proof. Since it holds that $O_i(\alpha \vee \beta) \equiv O_i(\alpha) \vee O_i(\beta)$, and $\alpha U_i(\beta \vee \theta) \equiv \alpha U_i \beta \vee \alpha U_i \theta$, we can move all the $\vee$ operators out of the temporal ones, which leads immediatly to the conclusion. □

Now that we are dealing with formulas without $\vee$ operators, we consider the following function defined inductively on these formulas (we write $(i,j) \in C$ whenever i and j belong to the same connected component):

$$\forall i \in P,\ \delta(p(i)) = p(i)$$

If $(i,j) \in C$ then:	If $(i,j) \notin C$ then:
$\delta(O_i p(j)) = O_i p(j)$	$\delta(O_i p(j)) = O_i \top \wedge Rp(j)$
$\delta(\alpha U_i p(j)) = \alpha U_i p(j)$	$\delta(\alpha U_i p(j)) = \alpha \wedge Rp(j)$
$\delta(O_i O_j \alpha) = O_i \delta(O_j \alpha)$	$\delta(O_i O_j \alpha) = O_i \top \wedge \delta(O_j \alpha)$
$\delta(O_i(\alpha U_i \beta) = O_i \delta(\alpha U_i \beta)$	$\delta(O_i(\alpha U_i \beta)) = O_i \top \wedge \delta(\alpha U_i \beta)$
$\delta(\alpha U_i(O_j \beta)) = \alpha U_i(\delta(O_j \beta))$	$\delta(\alpha U_i(O_j \beta)) = \alpha \wedge R(\delta(O_j \beta))$
$\delta(\alpha U_i(\beta U_j \theta) = \alpha U_i(\delta(\beta U_j \theta))$	$\delta(\alpha U_i(\beta U_j \theta)) = \alpha \wedge R(\delta(\beta U_j \theta))$

Then it is easy to prove (by induction) the following:

Proposition 11. *Let γ be a Φ_s formula without any $\vee$ operator, then $\gamma \equiv \delta(\gamma)$.*

Example 12. Let $\gamma = p(1)U_1(O_1(p(2)U_2q(1)))$ with $1 \in P_A$ and $2 \in P_B$. Then $\delta(\gamma) = p(1)U_1(O_1\top \wedge R(p(2) \wedge Rq(1)))$.

Now, using $\alpha U_i(\beta \wedge R\theta) \equiv \alpha U_i\beta \wedge R\theta$, $O_i(\alpha \wedge R\beta) \equiv O_i\alpha \wedge R\beta$, and $R(\alpha \wedge R\beta) \equiv R\alpha \wedge R\beta$, we can easily transform $\delta(\gamma)$ into a separated formula.

Example 13. With the same formula γ as above, it holds that $\delta(\gamma) \equiv p(1)U_1(O_1\top) \wedge Rp(2) \wedge Rq(1)$.

When dealing with initial equivalence, we can then remove all the R operators from $\delta(\gamma)$. This gives us a formula that is initially equivalent to γ. Intuitively, this formula is built from γ by cutting γ in different pieces whenever two temporal operators relating to different components occur successively.
Obviously no progress will be made in the worst case, for instance if we start with a formula already in $TrPTL(P_A)$. But, in the best case, one can hope to bound by the size of the subformulas by a constant, while the number of processes involved can be divided by 2. Testing the root-satifiability problem for the subformulas can thus sometimes be done dramatically faster than the initial problem for γ, depending on the structure of γ.

Example 14. Let $\gamma = O_1\, O_2\, O_1\, O_2 \ldots O_1\, O_2\, p(2)$, with $1 \in P_A$ and $2 \in P_B$. Then γ can be decomposed as the initially equivalent formula $O_1 \top \wedge O_2\, p(2)$, regardless of the size of γ.

6 Conclusion

By the time we had submitted this work, Thiagarajan and Walukiewicz [TW97] have exhibed a new logic on traces, called $LTrL$, that is actually expressively complete (i.e. equivalent to the first-order logic $FO(<)$ on traces), but does not have yet an elementary decision procedure, as opposed to $TrPTL$. Therefore Mezei's theorem holds for $LTrL$-definable languages. Nevertheless, [TW97] does not give any way to obtain an effective decomposition of $LTrL$ formulas. Thus, we shall extend the results of the present paper to this new logic $LTrL$ in a future version of this work.

References

[Ber79] J. Berstel. *Transductions and context-free languages.* Teubner Studienbücher, 1979.

[Die90] V. Diekert. *Combinatorics on Traces.* Number 454 in LNCS. Springer, 1990.

[DR95] V. Diekert and G. Rozenberg, editors. *The Book of Traces.* World Scientific, Singapore, 1995.

[EM93] W. Ebinger and A. Muscholl. Logical definability on infinite traces. In A. Lingas, R. Karlsson, and S. Carlsson, editors, *Proc. of the 20th ICALP, Lund (Sweden) 1993*, number 700 in LNCS, pages 335–346. Springer, 1993.

[GP92] P. Gastin and A. Petit. Asynchronous automata for infinite traces. In W. Kuich, editor, *Proc. of the 19th ICALP, Vienna (Austria) 1992*, number 623 in LNCS, pages 583–594. Springer, 1992.

[GPZ94] P. Gastin, A. Petit, and W. Zielonka. An extension of Kleene's and Ochmański's theorems to infinite traces. *Theoret. Comp. Sci.*, 125:167–204, 1994. A preliminary version was presented at ICALP'91, LNCS 510 (1991).

[GW94] P. Godefroid and P. Wolper. A partial approach to model checking. *Inform. and Comp.*, 110:305–326, 1994.

[KP92] S. Katz and D. Peled. Interleaving set temporal logic. *Theoret. Comp. Sci.*, 75:21–43, 1992.

[LRT92] K. Lodaya, R. Ramajunam, and P.S. Thiagarajan. Temporal logics for communicating sequential agents:I. *Int. J. of Found. of Comp. Sci.*, 3(2):117–159, 1992.

[Maz77] A. Mazurkiewicz. Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University, Aarhus, 1977.

[MT92] M. Mukund and P.S. Thiagarajan. A logical characterization of well branching event structures. *Theoret. Comp. Sci.*, 96:35–72, 1992.

[MT96] M. Mukund and P.S. Thiagarajan. Linear time temporal logics over Mazurkiewicz traces. In *Proc. of the 21th MFCS, 1996*, number 1113 in LNCS, pages 62–92. Springer, 1996.

[Pen88] W. Penczek. A temporal logic for event structures. *Fundamenta Informaticae*, XI:297–326, 1988.

[PK95] W. Penczek and R. Kuiper. Traces and logic. In V. Diekert and G. Rozenberg, editors, *The book of Traces*, pages 307–381, 1995.

[Pnu77] A. Pnueli. The temporal logics of programs. In *Proc. of the 18th IEEE FOCS, 1977*, pages 46–57, 1977.

[Thi94] P.S. Thiagarajan. A trace based extension of linear time temporal logic. In *Proc. of the 9th LICS, 1994*, pages 438–447, 1994.

[Tho89] W. Thomas. On logical definability of trace languages. In V. Diekert, editor, *Proc. an ASMICS workshop, Kochel am See 1989*, Report TUM-I9002, Technical University of Munich, pages 172–182, 1989.

[Tho90] W. Thomas. Automata on infinite objects. In J. v. Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 133–191. Elsevier Science Publishers, 1990.

[TW97] P.S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *Proc. of LICS'97 (to appear)*, 1997.