Transaction Logic with External Actions

Ana Sofia Gomes and José Júlio Alferes

Departamento de Informática
Faculdade Ciências e Tecnologias
Universidade Nova de Lisboa
2829-516 Caparica, Portugal

Abstract. We propose External Transaction Logic (or $\mathcal{ETR}$), an extension of Transaction Logic able to represent updates in internal and external domains whilst ensuring a relaxed transaction model. With this aim, $\mathcal{ETR}$ deals with two main components: an internal knowledge base where updates follow the strict ACID model, given by the semantics of Transaction Logic; and an external knowledge base of which one has limited or no control and can only execute *external* actions. When executing actions in an external domain, if a failure occurs, it is no longer possible to simply rollback to the initial state before executing the transaction. For dealing with this, similarly to what is done in databases, we define compensating operations for each external action to be performed to ensure a relaxed model of atomicity and consistency. By executing these compensations in backward order, we obtain a state considered to be equivalent to the initial one. The definition of $\mathcal{ETR}$ is parametric on a so called external oracle, which describes the behavior of the external knowledge base. This allows for accommodating several different languages and semantics for describing effects of actions in an external knowledge base. For example, languages such as Action Languages can be used as external oracles. This way, $\mathcal{ETR}$ can also be seen as a logic that combines Transaction Logic with Action Languages. In this paper we present the language and semantics of $\mathcal{ETR}$, give some examples of oracles, and construct a sound and complete SLD-style proof theory for a Horn-like subset of the logic.

1 Introduction

Transaction Logic ($\mathcal{TR}$) is an extension of predicate logic proposed in [2] which exhibits a clean and declarative semantics, along with a sound and complete proof theory, to reason about state changes in arbitrary logical theories (as databases, logic programs or other knowledge bases). Unlike many other logic systems, $\mathcal{TR}$ imposes that the knowledge base evolves only into consistent states respecting ACID properties[1]. $\mathcal{TR}$ is parameterized by a pair of oracles that encapsulate elementary knowledge base operations of querying and updating, respectively, allowing $\mathcal{TR}$ to reason about elementary updates but without committing to a particular theory. Thus $\mathcal{TR}$ provides a logical foundation for knowledge base transactions accommodating a wide variety of semantics.

Example 1 (Financial Transactions). As illustration of $\mathcal{TR}$, consider a knowledge base of a bank (taken from [2]) where the balance of a bank account is given by the relation

[1] As usual in databases, ACID stands for Atomicity, Consistency, Isolation and Durability.

$balance(Acnt, Amt)$. To modify this relation, we are provided with a pair of elementary update operations: $balance(Acnt, Amt).del$ to delete a tuple from the relation, and $balance(Acnt, Amt).ins$ to insert a tuple into the relation. Using these two updates, we define four transactions: $changeBalance(Acnt, Bal, Bal')$ to change the balance of an account; $withdraw(Amt, Acnt)$ to withdraw an amount from an account; $deposit(Amt, Acnt)$ to deposit an amount into an account, and to transfer an amount from one account to another $transfer(Amt, Acnt, Acnt')$. These transactions can be defined in $\mathcal{TR}$ by the following four rules:

$$transfer(Amt, Acnt, Acnt') \leftarrow withdraw(Amt, Acnt) \otimes deposit(Amt, Acnt')$$
$$withdraw(Amt, Acnt) \leftarrow balance(Acnt, B) \otimes changeBalance(Acnt, B, B - Amt)$$
$$deposit(Amt, Acnt) \leftarrow balance(Acnt, B) \otimes changeBalance(Acnt, B, B + Amt)$$
$$changeBalance(Acnt, B, B') \leftarrow balance(Acnt, B).del \otimes balance(Acnt, B').ins$$

Intuitively, the first rule states that a transfer of amount Amt from account $Acnt$ to account $Acnt'$ is performed if first a withdrawal of Amt from $Acnt$ is performed, and then a deposit of the same amount to $Acnt'$ is performed. The last rule states that changing the balance of account $Acnt$ from B to B' is true (in a sequence of knowledge base states) in case first the truth of $balance(Acnt, B)$ is deleted from the knowledge base according to the update-oracle, and then $balance(Acnt, B')$ is inserted.

State change and evolution in $\mathcal{TR}$ is caused by executing ACID transactions. These transactions are formulas that can be posed into the system in a Prolog-like style as $? - \varphi$. One characteristic of the theory is that every formula is assumed as a transaction. Particularly, by posing φ we know that either φ can be executed respecting all ACID properties and the knowledge base evolves from an initial state D_0 into a final state D_n passing through an arbitrary number of states n; or φ cannot be executed under these conditions and so the knowledge base does not evolve and remains in the state D_0.

Unfortunately, $\mathcal{TR}$ is not suitable to model situations where besides executing ACID actions, some steps of the transaction require interaction with an external domain. That is, systems where the internal state of the knowledge base evolves ensuring the ACID properties, but this evolution depends on the execution of some actions in an external knowledge base, of which one has a limited control and interaction. This is, e.g., the case of web-based systems with an internal knowledge base that follows the strict ACID model, and that interact with other systems that they do not control, for instance, via web-services. As illustration, consider a system for a web shop that accepts orders from clients. Whenever a client submits an order, the system must take care of payments and updates the inventory of the product to be sell. Obviously, it is crucial that each order is internally treated as a transaction. However, payments are validated and executed externally by the system of a bank, with which it communicates, but is external to the web shop, and over which it has a limited control.

Ensuring the standard ACID model in an external world is no longer possible. Particularly, the external actions executed in these domains cannot be rollbacked, as one has no control of the *external* system where they were performed. Moreover, since these actions require interaction with an external entity, this kind of transaction can last for relatively long periods of time, delaying the termination of shorter and more common transactions. To address this problem, [6] proposes the notion of long-lived transaction or sagas. In such situations, where simply restoring the initial state is no longer possible, the approach of [6] is to define compensating operations for each operation to be performed. The idea is that if these compensations are performed in backward order, then they lead the database into a state that is considered equivalent to the initial one, thus ensuring some weaker form of atomicity.

In this paper we propose External Transaction Logic ($\mathcal{ETR}$) that augments $\mathcal{TR}$ theory with the ability to reason about an external domain, and with a notion of compensations. The external reasoning is performed by an external oracle, parametric to the language, which describes the behavior of the external knowledge base. Moreover, $\mathcal{ETR}$ allows for two different kinds of formulas: standard transaction formulas that follow the strict ACID model leading the *internal* knowledge base always into consistent states; and external action formulas, of the form $\mathbf{ext}(a, a^{-1})$, that follow a relaxed ACID model, and in case of failure executes the compensation a^{-1} leading the *external* knowledge base into an equivalent consistent state, but possibly different from the original one. As illustration, consider the following example:

Example 2. Consider the system of the web shop mentioned above, where clients submits orders. In the end of each order, a final confirmation is asked to the client that may or not confirm the transaction. If the client accepts it, the order ends successfully. Otherwise, the transaction fails and consistency must be preserved. In this case, it means that we need to rollback the update of the stock, and to compensate for the executed payment. The obvious compensation here is to simply ask the bank to refund the charged money. However, note that the transaction may fail sooner. E.g. the transaction may fail if the bank cannot charge the given amount in the credit card, or if the product is out of stock. This situation can be modeled in $\mathcal{ETR}$ by the rules:

$$\begin{aligned}
buy(Prdt, Card, Amt) \leftarrow\ & \mathbf{ext}(chargeCard(Card, Amt), refundCard(Card, Amt)) \\
& \otimes updateStock(Prdt) \\
& \otimes \mathbf{ext}(confirmTransaction(Product, Card, Amt), ())
\end{aligned}$$

$$updateStock(Prdt) \quad \leftarrow N > 0 \otimes product(Prdt, N).del \otimes product(Prdt, N-1).ins$$

External actions can succeed or fail depending on the state of the external world. For instance, charging a given amount in a credit card depends on many things, e.g. if the credit limit is exceeded, or if the card has expired. As explained, to reason about the outcomes of these external actions $\mathcal{ETR}$ assumes the existence of an external oracle that comes as a parameter of the theory. The flexibility provided by having such an external oracle as a parameter, allows for the combination of $\mathcal{ETR}$ with several different languages and semantics for describing the effects of actions in an external knowledge base. These include powerful formalisms that reason about state change and the related phenomena of time and action as action languages [7], the situation calculus [11], event calculus [10], process logic [8] and many others. All these formalisms are orthogonal to $\mathcal{ETR}$, in the sense that they can just be "plugged" in the theory. Moreover, the main novelty is that $\mathcal{ETR}$ provides such combination whilst also ensuring a relaxed model of transaction, thus being suited for hybrid systems that require the notion of transaction.

We continue, in Section 2, by defining the language of $\mathcal{ETR}$, and by giving examples of external oracles (Section 2.1). We then propose a revised model theory (Section 3), and make the correspondence with $\mathcal{TR}$ Theory (Section 4.1). Afterwards we present a sound and complete top-down procedure for the serial-Horn instantiation of $\mathcal{ETR}$ (Section 5). Finally, we end with some comparisons to related systems (Section 6). *All proofs are included in this version of the paper.*

2 Syntax

Without loss of generality (cf. [4]), we work with a Herbrand instantiation of the $\mathcal{TR}$ framework as defined in [4]. As usual, the Herbrand universe $\mathcal{U}$ is the set of all ground first-order terms that can be constructed from the function symbols in the language; the Herbrand base $\mathcal{B}$ is a set of all ground atomic formulas in the language; and a classical Herbrand structure is any subset of $\mathcal{B}$.

To build complex logical formulas, $\mathcal{TR}$ uses the usual classical logic connectives $\wedge, \vee, \neg, \rightarrow$. In addition, $\mathcal{TR}$ also adds a new connective $\otimes$, denoted *serial conjunction* operator. Informally, the formula $\phi \otimes \psi$ represents an action composed of an execution of ϕ followed by an execution of ψ. Logical formulas in $\mathcal{ETR}$ are called *transaction formulas*, and a set of transaction formulas is called a *transaction base*. Furthermore, $\mathcal{ETR}$ extends $\mathcal{TR}$ with a special kind of formula $\mathbf{ext}(a, a^{-1})$ known as *external*. In this formula a and a^{-1} are atoms, where a denotes the action to be performed and a^{-1} its corresponding compensation. Note that there is no restriction on the signature of these atoms. Particularly, they can be any atom from the Herbrand base $\mathcal{B}$ and they may also appear defined in transaction formulas. However, as we shall see, formulas defined using the special predicate $\mathbf{ext}$ are evaluated w.r.t. the external oracle $\mathcal{O}^e$, while all the others are evaluated w.r.t. the data and transition oracle, $\mathcal{O}^d$ and $\mathcal{O}^t$ respectively.

2.1 States, Operations and Oracles

One characteristic of $\mathcal{TR}$ is that its theory is parameterized by a pair of oracles $\mathcal{O}^d$ and $\mathcal{O}^t$, respectively denoted the data and the transition oracle. These oracles encapsulate the elementary knowledge base operations, allowing the separation of elementary operations from the logic of combining them. As a result of this separation, $\mathcal{TR}$ does not commit to any particular theory of elementary updates. Consequently, the language itself is not fixed and $\mathcal{TR}$ is able to accommodate a wide variety of knowledge base semantics, from classical to non-monotonic to various other non-standard logics [2].

$\mathcal{ETR}$ follows the same principles. In addition to the data oracle and the transition oracle, $\mathcal{ETR}$ requires an additional oracle to evaluate elementary external operations, the *external oracle*. Assuming this external oracle allows $\mathcal{ETR}$ to abstract the theory and semantics of the external domain, encapsulating the elementary operations that can be performed externally. These oracles are not fixed and almost any triple of oracles can be plugged into $\mathcal{ETR}$ theory.

All oracles assume a set of *state identifiers* that uniquely identify a state. The *state data oracle* $\mathcal{O}^d$ is a mapping from state identifiers to sets of first-orders formulas. Intuitively, given a state identifier i, $\mathcal{O}^d(i)$ retrieves the set of formulas consider to be true in d. The *state transition oracle* $\mathcal{O}^t(i_1, i_2)$ is a function that maps pairs of knowledge base states into sets of ground atomic formulas denoted as elementary transitions. Finally, the *external oracle* $\mathcal{O}^e(i_1, i_2)$ is a mapping from pairs of external knowledge base state identifiers into a set of ground formulas denoted as external actions. Intuitively, the actions that are mapped from $\mathcal{O}^e(i_1, i_2)$ are all those that make the external knowledge base evolve from state i_1 to state i_2. A transaction base equipped with these 3 oracles constitutes an $\mathcal{ETR}$ theory.

Moreover, to be able to execute transactions given an initial internal state, and an initial external state, we further consider so called $\mathcal{ETR}$ programs. This is made precise as follows:

Definition 1 **($\mathcal{ETR}$ theories and programs).** *Given a language $\mathcal{L}$, a set of (internal) state identifiers $\mathcal{D}$ and a set of external state identifiers $\mathcal{E}$, an $\mathcal{ETR}$* theory *is composed of a transaction base in the language $\mathcal{L}$, of a state data oracle $\mathcal{O}^d$ mapping elements in $\mathcal{D}$ into transaction formulas in the language of $\mathcal{L}$, a state transition oracle $\mathcal{O}^t$ mapping pairs of elements in $\mathcal{D}$ into transaction formulas of $\mathcal{L}$, and an external oracle $\mathcal{O}^e$ mapping pairs of elements in $\mathcal{E}$ into transaction formulas of $\mathcal{L}$.*

A transaction program P in $\mathcal{ETR}$ consists of three distinct parts: an $\mathcal{ETR}$ theory, an initial internal state identifier $\mathcal{D}_i$, and an initial external state identifier $\mathcal{E}_i$.

2.2 Oracles Examples

Oracles encapsulate the elementary updates of the knowledge base. In practice this means that elementary operations, as insert or delete, are defined only at the oracle level and are independent of the semantics of $\mathcal{TR}$ and $\mathcal{ETR}$. For instance, if one wants the internal knowledge base to behave as a relational database, then intuitively, one needs to choose a data and transition oracle with the corresponding behavior. In the case of a relational oracle it means that each state D should be treated as the set of ground atomic formulas. Then the data oracle simply returns all these formulas, i.e., $\mathcal{O}^d(D) = D$. Moreover, the elementary operations insert and delete need also to be defined in the transition oracle. For that, one can simply consider two special kind of predicates $p.ins$ and $p.del$ for every predicate symbol p in D, representing the insertion and deletion of single atoms, respectively. Formally, $p.ins \in \mathcal{O}^t(D_1, D_2)$ iff $D_2 = D_1 + \{p\}$ and $p.del \in \mathcal{O}^t(D_1, D_2)$ iff $D_2 = D_1 - \{p\}$.

Further semantics are also possible, as for instance the well-founded semantics [13]. This semantics can be implemented by considering a state D as a set of generalized Horn rules[2]. The data oracle $\mathcal{O}^d(D)$ then returns the set of literals in the well-founded model of D. Such oracles can represent any rule base with the well-founded semantics, which includes Horn rule-bases, stratified rule-bases and locally-stratified rule-bases. In the above examples, states are equated with sets of literals. But this is not necessarily so, as $\mathcal{TR}$ can have oracles and states that include rules, this way allowing for incorporating rule updates, such as those in logic programming update languages [1] For a detailed description on these and further examples of data and transition oracle see [2].

Besides these two kinds of oracles, $\mathcal{ETR}$ further demands an external oracle to model the behavior of the external environment. For defining such an oracle, one has to specify which actions make the external knowledge base evolve from one state to another, i.e. to specify how actions change the external knowledge base. Languages

[2] Generalized Horn rules are rules with possibly negated premisses

introduced exactly for encoding these kind of transaction systems, and for representing effects of actions, can of course be used to defined external oracle. As examples of these oracles consider:

Action Language Oracle Recall that an action language system [7] is defined over a labeled direct graph, where the triple $\langle s, A, s' \rangle$ represents an edge leading from s to s' and labeled A. Intuitively, s and s' are states of the world and A is an action that lead s into s'. An external oracle $\mathcal{O}^e$ is a mapping from a pair of states s, s' into a set of actions A such that $\mathcal{O}^e(s, s') = \{A : \langle s, A, s' \rangle\}$.
Note that this definition is general and can be instantiated for any particular action language. For instance, for the action language $\mathcal{B}$[7], $\mathcal{O}^e(s, s') = \{A : Cn_Z(E(A, s) \cup (s \cap s')) = s'\}$, where the $E(A, s)$ lists the effects of action A executed in the state s; the set $s \cap s'$ represent the set of facts that are preserved by inertia; finally the application of Cn_Z adds the indirect effects of A to this union.

Situation Calculus Oracle Here a state id E_i is defined over domain S, the set of all situations s in the world. Recall the operator $do(action, s)$ which is itself a situation [11]. The external oracle returns the action that led to the current situation, i.e. $\mathcal{O}^e(E_1, E_2) = A$, iff $E_2 = do(A, E_1)$.

3 Model Theory

The model theory for $\mathcal{ETR}$ is a generalization of $\mathcal{TR}$'s semantics, where the main novelty in $\mathcal{ETR}$ is the notion of *compensation*. A compensation occurs when the executed transaction ϕ contains external actions and fails. Since, in such a case, it is not possible to simply rollback to the initial state before executing ϕ, a series of compensating actions are executed to restore the consistency of the external knowledge base.

Central to $\mathcal{TR}$'s model theory is the notion of paths, i.e. sequence of states. Logical formulas in $\mathcal{ETR}$ are evaluated on two sets of paths: the path of the internal knowledge base, and the path of the external knowledge base. Intuitively, external formulas of the form $\mathbf{ext}(a, a^{-1})$ are evaluated w.r.t. external paths by the external oracle $\mathcal{O}^e$, whereas the remaining logical formulas are evaluated in internal paths by the data and the transition oracle. Though it is not strictly needed, as we show below, for easing the definitions, we further include for evaluation of the formulas the sequence of external actions which gave rise to the external path.

A semantic structure in $\mathcal{ETR}$ is a mapping from a pair of paths and a sequence of actions to Herbrand Structures, specifying what atoms are true on what paths. These mappings are constrained by the oracles.

Definition 2 (Extended Herbrand Path Structure). *An extended Herbrand path structure is a mapping M that given a path and a sequence of external actions, assigns a classical Herbrand structure (or $\top$) to every path*[3]*. This mapping is subject to the following restriction:*

[3] Similarly, to $\mathcal{TR}$ we augment the classical Herbrand structure with an abstract structure denoted $\top$. This abstract structure represents a technical detail that allows the semantic structures in $\mathcal{ETR}$ (and $\mathcal{TR}$) to be defined as *total* mappings. The interested reader is referred to [4] for more details.

1. $M(\langle D \rangle)(\langle E \rangle)(\emptyset) \models \varphi$ *if* $\mathcal{O}^d(D) \models \varphi$
2. $M(\langle D_1, D_2 \rangle)(\langle E \rangle)(\emptyset) \models \varphi$ *if* $\mathcal{O}^t(D_1, D_2) \models \varphi$
3. $M(\langle D \rangle)(\langle E_1, E_2 \rangle)(\langle A \rangle) \models A$ *if* $\mathcal{O}^e(E_1, E_2) \models A$

The definition of satisfaction of more complex formulas, over general paths, requires the prior definition of operations on paths. These take into account how serial conjunction is satisfied, and how to construct the correct compensation.

Definition 3 (Paths and Splits). *A* path *of length k, or a* k-path*, is any finite sequence of states, $\pi = \langle D_1, \ldots, D_k \rangle$, where $k \geq 1$. A* split *of π is any pair of subpaths, π_1 and π_2, such that $\pi_1 = \langle D_1, \ldots, D_i \rangle$ and $\pi_2 = \langle D_i, \ldots, D_k \rangle$ for some i $(1 \leq i \leq k)$. In this case, we write $\pi = \pi_1 \circ \pi_2$.*

Definition 4 (External action split). *A sequence α of length j is any finite sequence of external actions (possibly empty), $\alpha = \langle A_1, \ldots, A_j \rangle$, where $j \geq 0$. A split of α is any pair of subsequences, α_1 and α_2, such that $\alpha_1 = \langle A_1, \ldots, A_i \rangle$ and $\alpha_2 = \langle A_{i+1}, \ldots, A_j \rangle$ for some i ($0 \leq i \leq k$). In this case, we write $\alpha = \alpha_1 \circ \alpha_2$.*

Note that there is a significant difference between Definition 3 and Definition 4. In fact, splits for sequences of external actions can be empty, and particularly, it is also possible to define splits of empty sequences as $\emptyset = \emptyset \circ \emptyset$; whereas, a split of a path requires a sequence with at least length 1.

Definition 5 (Rollback split). *A rollback split of π is any pair of finite subpaths, π_1 and π_2, such that $\pi_1 = \langle D_1, \ldots, D_i, D_1 \rangle$ and $\pi_2 = \langle D_1, D_{i+1}, \ldots, D_k \rangle$.*

Definition 6 (Inversion). *An external action inversion of a sequence α where $\alpha = (\mathbf{ext}(a_1, a_1^{-1}), \ldots, \mathbf{ext}(a_n, a_n^{-1}))$, denoted α^{-1}, is the corresponding sequence of compensating external actions performed in the inverse way as $(a_n^{-1}, \ldots, a_1^{-1})$.*

Definition 7 (Satisfaction). *Let M be a Herbrand path structure, π be an internal path, ϵ be an external path and α be a sequence of external actions. If $M(\pi, \epsilon, \alpha) = \top$ then $M, \pi, \epsilon, \alpha \models \phi$ for every transaction formula ϕ; otherwise:*

1. ***Base Case:*** *$M, \pi, \epsilon, \alpha \models p$ if $p \in M(\pi, \epsilon, \alpha)$ for any atomic formula p*
2. ***Negation:*** *$M, \pi, \epsilon, \alpha \models \neg\phi$ if it is not the case that $M, \phi, \epsilon, \alpha \models \phi$*
3. ***"Classical" Conjunction:*** *$M, \pi, \epsilon, \alpha \models \phi \wedge \psi$ if $M, \pi, \epsilon, \alpha \models \phi$ and $M, \pi, \epsilon, \alpha \models \psi$.*
4. ***Serial Conjunction:*** *$M, \pi, \epsilon, \alpha \models \phi \otimes \psi$ if $M, \pi_1, \epsilon_1, \alpha_1 \models \phi$ and $M, \pi_2, \epsilon_2, \alpha_2 \models \psi$ for some split $\pi_1 \circ \pi_2$ of path π, some split $\epsilon_1 \circ \epsilon_2$ of path ϵ, and some external action split $\alpha_1 \circ \alpha_2$ of external actions α.*
5. ***Compensating Case:*** *$M, \pi, \epsilon, \alpha \models \phi$ if $M, \pi_1, \epsilon_1, \alpha_1\alpha_1^{-1} \rightsquigarrow \phi$ and $M, \pi_2, \epsilon_2, \alpha_2 \models \phi$ for some rollback split π_1, π_2 of π, some split $\epsilon_1 \circ \epsilon_2$ of path ϵ, and some external action split α_1, α_2 of α.*
6. *For no other $M, \pi, \epsilon, \alpha, \phi$, $M, \pi, \epsilon, \alpha \models \phi$.*

In the sequel we also mention the satisfaction of disjunctions and implications, where as usual $\phi \vee \psi$ means $\neg(\neg\phi \wedge \neg\psi)$, and $\phi \leftarrow \psi$ means $\phi \vee \neg\psi$.

Comparing with the $\mathcal{TR}$ model theory definition as presented in [2], it is easy to check that we extend the model theory with the notion of compensation strategy (Point 5 of Definition 7). Intuitively, $M, \pi, \epsilon, \alpha, \alpha^{-1} \rightsquigarrow \phi$ means that, in the *failed* attempt to execute ϕ, a sequence of α external actions were performed. Since it is impossible to rollback to the point before the execution of these actions, consistency is ensured by performing a sequence of compensating external actions in backward order. Note that from the external oracle point of view there is no difference between a non-compensating external action and a compensating external action since both can fail or succeed, and in the latter case, evolving the external knowledge base. The path π represents the sequence of states consistent with the execution of α, but where $\pi = \langle D_1, \ldots, D_k, D_1 \rangle$, i.e. we explicitly rollback to the initial state, but keeping the trace of the failed evolution. We say that $\pi, \epsilon, \alpha, \alpha^{-1}$ is a consistency preserving path for formula ϕ, defined as follows.

Definition 8 (Consistency Preserving Path). *Let M be a Herbrand path structure, π be an internal path, ϵ an external path, and α be a* non-empty *sequence of external actions. The path π' is obtained from $\pi = \langle D_1, \ldots, D_n \rangle$ by removing the state D_n from the sequence; α^{-1} is a non-empty sequence of external actions obtained from α by inversion; ϵ_1 and ϵ_2 are some split of ϵ. We say that $M, \pi, \epsilon, \alpha, \alpha^{-1} \rightsquigarrow \phi$ iff $\exists b_1 \otimes \ldots \otimes b_i \otimes \ldots \otimes b_n$ such that:*
$M, \pi', \epsilon_1, \alpha \models \phi \leftarrow (b_1 \otimes \ldots \otimes b_i \otimes \ldots \otimes b_n)$
$M, \pi', \epsilon_1, \alpha \models b_1 \otimes \ldots \otimes b_i$
$M, \pi', \epsilon_1, \alpha \models \neg\, b_{i+1}$
$M, \pi', \epsilon_2, \alpha^{-1} \models \bigotimes \alpha^{-1}$
where $\bigotimes$ represents the operation of combining a sequence of actions using $\otimes$.

Note that this notion only holds for cases where α is not empty. Particularly, if no external actions are performed, then $\mathcal{ETR}$ theory corresponds to the standard $\mathcal{TR}$ theory. This relation between $\mathcal{ETR}$ and $\mathcal{TR}$ is studied with more detailed in Section 4.1.

Definition 9 (Models). *A path structure M is a model of a transaction formula ϕ if $M, \pi, \epsilon, \alpha \models \phi$ for every internal path π, every external path ϵ, and every action sequence α. In this case, we write $M \models \phi$. A path structure is a model of a set of formulas if it is a model of every formula in the set.*

4 Executional Entailment

Similarly to $\mathcal{TR}$, in addition to logical entailment $\mathcal{ETR}$ supports another form of entailment called *executional entailment*. Logical entailment allows one to *reason* about $\mathcal{ETR}$ theories, whilst executional entailment provides a logical account to *execute* $\mathcal{ETR}$ programs.

Definition 10 (Executional Entailment). *Let P be an $\mathcal{ETR}$ program, let ϕ be a transaction formula, let $D_1, \ldots, D_n$ be a sequence of knowledge base states, let $E_1, \ldots, E_m$ be a sequence of external knowledge base states, and $A_1, \ldots A_{m-1}$ be a sequence of external actions. Then, the following statement:*

$$P, (D_1, \ldots, D_n), (E_1, \ldots, E_m), (A_1, \ldots A_{m-1}) \models \phi \qquad (1)$$

is true if $M, \langle D_1, \dots, D_n \rangle, \langle E_1, \dots, E_m \rangle, \langle A_1, \dots A_{m-1} \rangle \models \phi$ *for every model* M *of* P. *We also define*

$$P, D_1 - - -, E_1 - - - \models \phi \tag{2}$$

to be true, if there is an internal knowledge base state sequence $D_1, \dots, D_n$ *and an external state sequence* $E_1, \dots, E_m$ *with actions* $A_1, \dots, A_{m-1}$ *that makes (1) true.*

Intuitively, statement 1 says that a successful execution of transaction ϕ can change the internal knowledge base from state D_1 into D_n, and the external knowledge base from state E_1 into E_m by executing the external actions $A_1, \dots, A_{m-1}$. Formally, this means that in every model of P, the path $\langle D_1, \dots, D_n \rangle, \langle E_1, \dots, E_m \rangle, \langle A_1, \dots A_{m-1} \rangle$ satisfies the formula ϕ.

Notice that, cf. Definitions 2 and Definition 7, every transition between external states E_i into E_{i+1} is caused by performing an external action, and these actions can be easily inferred by the theory. As such, to simplify the notation, we also consider executional entailment without explicitly referring the sequence of actions, and use the notations interchangeably throughout the paper:

$$P, (D_1, \dots, D_n), (E_1, \dots, E_m) \models \phi \tag{3}$$

Besides executional entailment, which accounts for successful execution of transactions, one can also define executional compensation to account for failed attempts to execute a transaction.

Definition 11 (Executional Compensation). *Let* P *be an* $\mathcal{ETR}$ *program, let* ϕ *be a transaction formula, let* $D_1, \dots, D_n$ *be a sequence of internal knowledge base states,* $E_1, \dots, E_m$ *be a sequence of external knowledge base states and let* $A_1, \dots, A_{m-1}$ *be a sequence of external actions. Then, the following statement:*

$$P, (D_1, \dots, D_n), (E_1, \dots, E_m), (A_1, \dots A_{m-1}) \rightsquigarrow \phi$$

is true if $M, \langle D_1, \dots, D_n \rangle, \langle E_1, \dots, E_m \rangle, \langle A_1, \dots A_{m-1} \rangle \rightsquigarrow \phi$ *for every model* M *of* P.

Proposition 1. *An executional compensation*
$P, (D_1, \dots, D_k, D_1), (E_1, \dots, E_j, \dots, E_{2j}), (A_1, \dots, A_{j-1}, A_{j-1}^{-1}, \dots, A_1^{-1}) \rightsquigarrow \phi$ *is true if* $\exists b_1 \otimes \dots \otimes b_i \otimes \dots \otimes b_n$ *such that:*

1. $P, (D_1, \dots, D_k), (E_1, \dots, E_j), (A_1, \dots, A_{j-1}) \models \phi \leftarrow b_1 \otimes \dots \otimes b_i \otimes \dots \otimes b_n$
2. $P, (D_1, \dots, D_k), (E_1, \dots, E_j), (A_1, \dots, A_{j-1}) \models b_1 \otimes \dots \otimes b_i$
3. $P, (D_1, \dots, D_k), (E_1, \dots, E_j), (A_1, \dots, A_{j-1}) \models \neg b_{i+1}$
4. $P, (D_1, \dots, D_k), (E_j, \dots, E_{2j}), (A_{j-1}^{-1}, \dots, A_1^{-1}) \models A_{j-1}^{-1} \otimes \dots \otimes A_1^{-1}$

Proof. Let us assume that conditions (1-4) are true. Then by Definition 1:

$$M, \langle D_1, \dots, D_k \rangle, \langle E_1, \dots, E_j \rangle, \langle A_1, \dots A_{j-1} \rangle \models b_1 \otimes \dots \otimes b_i \otimes \dots \otimes b_n \rightarrow \phi$$
$$M, \langle D_1, \dots, D_k \rangle, \langle E_1, \dots, E_j \rangle, \langle A_1, \dots A_{j-1} \rangle \models b_1 \otimes \dots \otimes b_i$$
$$M, \langle D_1, \dots, D_k \rangle, \langle E_1, \dots, E_j \rangle, \langle A_1, \dots A_{j-1} \rangle \models \neg b_{i+1}$$
$$M, \langle D_1, \dots, D_k \rangle, \langle E_j, \dots, E_{2j} \rangle, \langle A_{j-1}^{-1}, \dots A_1^{-1} \rangle \models A_{j-1}^{-1} \otimes \dots \otimes A_1^{-1}$$

for every model M of P. Now, by Definition 8:

$$M, \langle D_1, \dots, D_k, D_1\rangle, \langle E_1, \dots, E_{2j}\rangle, \langle A_1, \dots A_{j-1}, A_{j-1}^{-1}, \dots, A_1^{-1}\rangle \rightsquigarrow \phi$$

which by Definition 11 let us conclude:

$$P, (D_1, \dots, D_k, D_1), (E_1, \dots, E_j, \dots, E_{2j})(A_1, \dots, A_{j-1}, A_{j-1}^{-1}, \dots, A_1^{-1}) \rightsquigarrow \phi$$

as intended. □

4.1 Correspondence with $\mathcal{TR}$

We now show that $\mathcal{ETR}$ indeed is an extension of $\mathcal{TR}$. Particularly, we show that if the program P does not contain external actions, or if all external actions are evaluated by the external oracle as `true`, then $\mathcal{ETR}$ corresponds with $\mathcal{TR}$. This relation is made precise as follows:

Definition 12. *Let P be a program containing $\mathcal{ETR}$ rules. We denote $P^{\perp}$ (resp $P^{\top}$) as a program transformation from P that replaces every external action $\mathbf{ext}(a, a^{-1})$ by `false` (resp. `true`), where `false` (resp.`true`) is just a constant standing for any formula that is false (resp. true) in every structure.*

Theorem 1. *An execution path $(D_1, \dots, D_n)$ of a program $P^{\perp}$ is a valid execution for ϕ in $\mathcal{TR}$ if and only if it is also a valid execution in $\mathcal{ETR}$ without the execution of external actions. More precisely:*

$$P^{\perp}, (D_1, \dots, D_n) \models_{\mathcal{TR}} \phi \textit{ iff } P, (D_1, \dots, D_n), (E_1), () \models_{\mathcal{ETR}} \phi$$

Corollary 1. *If P does not contain external actions, then $\mathcal{ETR}$ corresponds to $\mathcal{TR}$*

Proof. If P does not contain external actions, then $P^{\perp} = P$ and this result comes immediately from Theorem 1.

Definition 13 (Compensation-Free). *A sequence of external actions is said to be compensation-free if and only if it does not contain an inversion A_i^{-1}. We denote this sequence as $(A_1, \dots, A_m)^+$.*

Lemma 1. *If there exists a compensation-free sequence of actions $(A_1, \dots, A_m)^+$ for formula ϕ such that $P, (D_1, \dots, D_n), (E_1, \dots, E_m), (A_1, \dots, A_{m-1})^+ \models_{\mathcal{ETR}} \phi$ then $P^{\top}, (D_1, \dots, D_n) \models_{\mathcal{TR}} \phi$*

Proof. As we already pointed out, the essential difference between $\mathcal{TR}$ model theory with $\mathcal{ETR}$ model theory is an additional compensating case and an additional condition in the serial conjunction case. If $P, (D_1, \dots, D_n), (E_1, \dots, E_m), (A_1, \dots, A_{m-1})^+ \models_{\mathcal{ETR}} \phi$ then for all models M, $M, \langle D_1, \dots, D_n\rangle, \langle E_1, \dots, E_m\rangle, \langle A_1, \dots A_{m-1}\rangle^+ \models_{\mathcal{ETR}} \phi$. As a result, if ϕ is proven in a compensation-free sequence of external actions, it means that ϕ is proven without the need to employ the compensating case of $\mathcal{ETR}$ model theory, for all models M that satisfy ϕ. Moreover, without the compensating case, $\mathcal{ETR}$ model theory is a restricted version of $\mathcal{TR}$ model theory in the sense that it has an additional condition. As a result if $M, \langle D_1, \dots, D_n\rangle, \langle E_1, \dots, E_m\rangle, \langle A_1, \dots A_{m-1}\rangle^+ \models_{\mathcal{ETR}} \phi$ holds then for $M, \langle D_1, \dots, D_n\rangle \models_{\mathcal{TR}} \phi$ also holds for all models M for the program $P^{\top}$.

Lemma 2. *If there exists a sequence of external actions* $(A_1, \ldots, A_{m-1})$ *together with a sequence of external states* $(E_1, \ldots, E_m)$ *such that*

$$P, (D_1, \ldots, D_n), (E_1, \ldots, E_m), (A_1, \ldots, A_{m-1}) \models_{\mathcal{ETR}} \phi$$

then

$$P^{\top}(D'_1, \ldots, D'_j) \models_{\mathcal{TR}} \phi$$

is true and $(D'_1, \ldots, D'_j)$ *is a subsequence of* $(D_1, \ldots, D_n)$.

Proof. If the sequence $(A_1, \ldots, A_{m-1})$ is empty or compensating-free then this result comes immediately from Theorem 1 and Lemma 1. Thus, it remains to show that this result still holds for non-compensating-free sequences. So assume that $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m), (A_1, \ldots, A_{m-1}) \models \phi$. Now, if this is the case, then for all models M of P: $M, \langle D_1, \ldots, D_n \rangle, \langle E_1, \ldots, E_m \rangle, \langle A_1, \ldots, A_{m-1} \rangle \models \phi$. Since $(A_1, \ldots, A_{m-1})$ is non-compensating-free, then ϕ must be proven by using the Compensating Case of Definition 7 since it is the only case that introduces compensations. So this means that it is possible to construct a subsequence of $D_1, \ldots, D_n, E_1, \ldots, E_m$ and $A_1, \ldots, A_{m-1}$ such that:
$(D_1, \ldots, D_n) = (D_1, \ldots, D_k, D'_1, \ldots, D'_q)$,
$(E_1, \ldots, E_m) = (E_1, \ldots, E_k, \ldots, E_{2k}, \ldots, E_p)$,
$(A_1, \ldots, A_{m-1}) = (A_1, \ldots, A_{k-1}, A^{-1}_{k-1}, A^{-1}_1, A_{2k-1}, \ldots A_{p-1})$
$M, \langle D_1, \ldots, D_k, D_1 \rangle, \langle E_1 \ldots, E_k, \ldots, E_{2k} \rangle, \langle A_1, \ldots, A_{k-1}, A^{-1}_{k-1} \rangle \rightsquigarrow \phi$,
and $M, \langle D'_1, \ldots, D'_q \rangle, \langle E_{2k} \ldots, E_p \rangle, \langle A_{2k-1}, \ldots, A_{p-1} \rangle \models \phi$

So now if $(A_{2k-1}, \ldots, A_{p-1})$ is empty or compensating-free then by Theorem 1 and Lemma 1 we have that $P, (D'_1, \ldots, D'_q) \models_{\mathcal{TR}} \phi$ where $D'_1, \ldots, D'_q$ is a subsequence of $(D_1, \ldots, D_n)$. However, if this is not the case, then we still need to show this property but now for $(D'_1, \ldots, D'_q), (E_{2k} \ldots, E_p), (A_{2k-1}, \ldots, A_{p-1})$, where $(A_{2k-1}, \ldots, A_{p-1})$ is non-compensating-free. We can now prove this by induction on the lenght of the path. Since $(D'_1, \ldots, D'_q), (E_{2k} \ldots, E_p), (A_{2k-1}, \ldots, A_{p-1})$ is strictly smaller than the the original $(D_1, \ldots, D_n), (E_1, \ldots, E_m), (A_1, \ldots, A_{m-1})$, then we know that since ϕ is proven, then it needs eventually to fall into the case where the external actions are compensating free, and thus ϕ is proven also in $\mathcal{TR}$ in a subsequence of $D_1, \ldots, D_n$.

Theorem 2. *If a transaction formula* ϕ *is* $\mathcal{ETR}$ *valid in a program* P *then* ϕ *is* $\mathcal{TR}$ *valid in a program* $P^{\top}$.

Proof. Immediately from Lemma 2

5 A Top-Down Procedure for serial-$\mathcal{ETR}$

Next we extend the proof theory for ground[4] serial-Horn $\mathcal{TR}$ fragment as described in [2]. The advantage of this particular fragment is that it can be formulated as a least-fixpoint theory in a logic programming style.

[4] The restriction to ground formulas is not essential and can be easily lifted. We only require it in order to simplify the presentation.

A serial-Horn program P is a finite set of *serial goals*. A serial-goal is a transaction formula of the form $a_1 \otimes a_2 \otimes \ldots \otimes a_n$, where each a_i is an atomic formula and $n \geq 0$. When $n = 0$, we write $()$, which denotes the empty goal. A *serial-Horn rule* has the form $b \leftarrow a_1 \otimes \ldots \otimes a_n$, where the body $a_1 \otimes \ldots \otimes a_n$ is a serial goal and the head b is an atom. As in [2] we restrict the oracles to be independent of P, i.e., for every knowledge base state D, no predicate symbol occurs both in a rule-head in P, and in a rule-body in the oracle.

We present a top-down procedure for verifying that $P, D_0 - - -, E_0 - - - \models \varphi$, i.e, that a transaction φ can succeed starting from state D_0, E_0. This procedure succeeds if and only if it finds an execution path for the transaction φ, that is a sequence of internal states $D_0, \ldots, D_n$, a sequence of external states $E_0, \ldots, E_m$ and a sequence of external actions $A_1, \ldots A_{m-1}$, such that $P, (D_0, \ldots, D_n), (E_0, \ldots, E_m), (A_1, \ldots A_{m-1}) \models \varphi$.

Definition 14. *Let P be a transaction program and T a transaction goal. An $SLD_{\mathcal{ETR}}$-derivation of $P \cup \{T\}$ consists of a (finite or infinite) sequence $S_0, S_1, \ldots$ of sequents for T in P, where each element of the sequence S_i is obtained by S_{i-1} using the following rules: $S_0 = (D_0), (E_0), () \vdash T$.*
Let $S_i = D_1, E_1, A_1 \vdash L_1 \otimes \ldots \otimes L_k$:

1. *if exists a rule $L_1 \leftarrow B_1 \otimes \ldots \otimes B_j$ then:*
   $S_{i+1} = D_1, E_1, A_1 \vdash B_1 \otimes \ldots \otimes B_j \otimes \ldots \otimes L_k$
2. *if $\mathcal{O}^d(D_1) \models L_1$ then:*
   $S_{i+1} = D_1, E_1, A_1 \vdash L_2 \otimes \ldots \otimes L_k$
3. *if $\mathcal{O}^t(D_1, D_2) \models L_1$ then:*
   $S_{i+1} = D_2, E_1, A_1 \vdash L_2 \otimes \ldots \otimes L_k$
4. *if $\mathcal{O}^e(E_1, E_2) \models L_1$ then:*
   $S_{i+1} = D_1, E_2, L_1 \vdash L_2 \otimes \ldots \otimes L_k$
5. *if a successful consistency preserving derivation can be constructed with root L_1 such that:*
   $P, (D_1, \ldots, D_j, D_1), (E_1, \ldots, E_{2p}), (A'_1, \ldots, A'_{p-1}, A'^{-1}_{p-1}, \ldots, A'^{-1}_1) \mid\!\leadsto L_1$ *then:*
   $S_{i+1} = D_1, E_{2p}, A_1^{-1} \vdash L_1 \otimes \ldots \otimes L_k$

An $SLD_{\mathcal{ETR}}$-derivation for T is successful *if the sequence $S_1, S_2, \ldots, S_f$ is finite and ends in the empty goal, i.e., $S_f = (D_1, \ldots, D_n)(E_1, \ldots, E_m)(A_1, \ldots, A_{m-1}) \vdash ()$.*

The procedure manipulates expressions of the form $P, D - - -, E - - - \vdash T$ called *sequents*. If T is the empty transaction $()$ then the sequent is said to be successful. Given this procedure, a proof of a sequent seq_n is a series of sequents, $seq_1, \ldots, seq_{n-1}, seq_n$ where each seq_i is either an axiom-sequent or is derived from earlier sequents by one of the rules. If $(D_0, E_0), \ldots, (D_{n-1}, E_{m-1}), (D_n, E_m)$ are the internal and external knowledge base states of these sequents, respectively, then $(D_n, E_m), \ldots (D_0, E_0)$ is called the *execution path* of the deduction.

Note that the procedure presented on Definition 14 resembles a SLD-style procedure and is an extension of the inference system for serial-Horn $\mathcal{TR}$ as presented in [2]. Particularly, besides evaluating the external actions w.r.t the external oracle $\mathcal{O}^e$ (Point 4), we also introduce the possibility of executing compensations at any step of the derivation (Point 5) non-deterministically.

Definition 15 (Consistency Preserving Derivation). *Let P be a transaction program, T a transaction goal, D_i a given internal knowledge base state, and E_j a given external knowledge base state. A consistency preserving derivation of $P \cup \{T\}$ consists of a twofold finite sequence $S'_0, S'_1, \ldots, S'_c, S'_{c+1}, \ldots, S'_f$ of sequents for T in P starting in (D_i, E_j). In the first part of the sequence from S'_0 until S'_c, each element of the sequence S'_i is obtained by S'_{i-1} with $S'_0 = (D_i)(E_j)() \vdash' T$ using the rules of Definition 14,* excluding *Point (5). This first part of the sequence is said to be successful if the last sequent of the derivation is $S'_c = (D_i, \ldots, D_n)(E_i, \ldots, E_m)(A_1, \ldots, A_{m-1}) \vdash' b_1 \otimes \ldots \otimes b_k$ and no rule can be applied to b_1.*

The second part of the sequence is from S'_{c+1} until S'_f, where each element S'_i is obtained by S'_{i-1} with $S_{c+1} = (D_n), (E_m)() \vdash A^{-1}_{m-1} \otimes \ldots \otimes A^{-1}_1$ using only the Point (4) of Definition 14. This second part is said to be successful if the sequence ends with the axiom-sequent, i.e., the last sequent of the derivation is $S'_f = (D_n)(E_m, \ldots, E_{2m})(A^{-1}_{m-1}, \ldots A^{-1}_1) \vdash ()$. The sequence is said to be successful if their both subsequences are successful. We denote this derivation as:

$$P, (D_i, \ldots, D_n, D_i), (E_k, \ldots, E_{2m})(A_1, \ldots, A_{m-1}, A^{-1}_{m-1}, \ldots, A^{-1}_1) \mid\!\leadsto T$$

Theorem 3 (Soundness $\vdash$). *Let P be a ground transaction program with external actions (possibly infinite) and ϕ is a serial-Horn goal:*
If $P, D{-}{-}{-}, E{-}{-}{-} \vdash \phi$ then $P, D{-}{-}{-}, E{-}{-}{-} \models \phi$

Proof. To prove Theorem 3 it is enough to show that the axioms and derivations of the system are sound. This is proven by Lemma 3 and Lemma 4 respectively.

Lemma 3 (Soundness of Axiom). $P, D, E \models ()$ *for any transaction base P, any internal knowledge base D and any external knowledge base E.*

Proof. The transaction () describes the empty transaction, which by definition does nothing and always succeeds disregarding the current states of the internal and external knowledge base.

Lemma 4 (Soundness of Rules).

1. *Suppose a rule $\phi_1 \leftarrow \psi_1 \otimes \ldots \otimes \psi_j$ exists in P.*
   If $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \models \psi_1 \otimes \ldots \otimes \psi_j \otimes \phi_2 \otimes \ldots \otimes \phi_k$ then
   $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \models \phi_1 \otimes \ldots \otimes \phi_k$
2. *Suppose that $\mathcal{O}^d(D_1) \models \phi_1$.*
   If $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \models \phi_2 \otimes \ldots \otimes \phi_k$ then
   $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \models \phi_1 \otimes \ldots \otimes \phi_k$
3. *Suppose $\mathcal{O}^t(D_1, D_2) \models \phi_1$.*
   If $P, (D_2, \ldots, D_n), (E_1, \ldots, E_m) \models \phi_2 \otimes \ldots \otimes \phi_k$ then
   $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \models \phi_1 \otimes \ldots \otimes \phi_k$
4. *Suppose that $\mathcal{O}^e(E_1, \ldots, E_p) \models \phi_1$ then*
   If $P, (D_1, \ldots, D_n), (E_p, \ldots, E_m), (A_2, \ldots, A_q) \models \phi_2 \otimes \ldots \otimes \phi_k$ then
   $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m)(\phi_1, A_2, \ldots, A_q) \models \phi_1 \otimes \ldots \otimes \phi_k$

5. *Suppose a consistency preserving derivation can be constructed for* ϕ_1*:*
$P,(D_1,\ldots,D_j),(E_1,\ldots,E_{2l}),(A_1,\ldots,A_{l-1},A_{l-1}^{-1},\ldots,A_1^{-1}) \mid\!\leadsto \phi_1$
If $P,(D_1,\ldots,D_j,D_1,\ldots,D_n)(E_1,\ldots,E_l,E_{l+1},\ldots,E_{2l},E_{2l+1},\ldots,E_m) \models \phi_1\otimes \ldots\otimes\phi_k$ *then*
$P,(D_1,\ldots,D_n),(E_{2l},\ldots,E_m) \models \phi_1\otimes\ldots\otimes\phi_k$

Proof. We prove each item in turn:

1. Note that since $\phi_1 \leftarrow \psi_1\otimes\ldots\otimes\psi_j$ is in P, then every model of P is also a model of $\phi_1 \leftarrow \psi_1\otimes\ldots\otimes\psi_j$. Then for every model M of P,

$$M,\langle D_1,\ldots,D_n\rangle\langle E_1,\ldots,E_m\rangle \models \psi_1\otimes\ldots\otimes\psi_j\otimes\phi_2\otimes\ldots\otimes\phi_k$$

By Definition 7 there are two possible cases in this situation:
a) Serial Conjunction Case:
$M,\langle D_1,\ldots,D_p\rangle\langle E_1,\ldots,E_q\rangle \models \psi_1\otimes\ldots\otimes\psi_j$
and $M,\langle D_p,\ldots,D_n\rangle\langle E_q,\ldots,E_m\rangle \models \phi_2\otimes\ldots\otimes\phi_k$

$M,\langle D_1,\ldots,D_p\rangle\langle E_1,\ldots,E_q\rangle \models \phi_1$ Since M is a model of $\phi_1 \leftarrow \psi_1\otimes\ldots\otimes\psi_j$
and $M,\langle D_p,\ldots,D_n\rangle\langle E_q,\ldots,E_m\rangle \models \phi_2\otimes\ldots\otimes\phi_k$

$M,\langle D_1,\ldots,D_n\rangle\langle E_1,\ldots,E_m\rangle \models \phi_1\otimes\phi_2\otimes\ldots\otimes\phi_k$ By Definition 7 (Serial Conjunction Case)

b) Compensating Case:
$\langle D_1,\ldots,D_n\rangle = \langle D_1,\ldots,D_p,D_1,\ldots D_r\rangle$
and $\langle E_1,\ldots,E_m\rangle = \langle E_1,\ldots,E_q,E_{q+1},\ldots,E_{2q},E_{2q+1},\ldots E_s\rangle$,
$M,\langle D_1,\ldots,D_p,D_1\rangle\langle E_1,\ldots,E_q,E_{q+1},\ldots,E_{2q}\rangle \leadsto \psi_1\otimes\ldots\otimes\psi_j\otimes\phi_2\otimes\ldots\otimes\phi_k$
and $M,\langle D_1,\ldots,D_r\rangle\langle E_{2q},\ldots,E_s\rangle \models \psi_1\otimes\ldots\otimes\psi_j\otimes\phi_2\otimes\ldots\otimes\phi_k$

So, we need to show exactly the same property but now for $M,\langle D_1,\ldots,D_r\rangle\langle E_{2q},\ldots,E_s\rangle \models \psi_1\otimes\ldots\otimes\psi_j\otimes\phi_2\otimes\ldots\otimes\phi_k$. Again in this case, there are two possibilities that can be applied. However, the path $\langle D_1,\ldots,D_r\rangle\langle E_{2q},\ldots,E_s\rangle$ is strictly smaller than the original $\langle D_1,\ldots,D_n\rangle\langle E_1,\ldots,E_m\rangle$. As a result, by induction on the length of the path, we know that the path will eventually fall into the Serial Conjunction Case in order to be satisfied, and thus that $M,\langle D_1,\ldots,D_r\rangle\langle E_{2q},\ldots,E_s\rangle \models \phi_1\otimes\phi_2\otimes\ldots\otimes\phi_k$.
Furthermore, since $(\psi_1\otimes\ldots\otimes\psi_j\otimes\phi_2\otimes\ldots\otimes\phi_k)\rightarrow(\phi_1\otimes\ldots\otimes\phi_k)$, then by transitivity of $\rightarrow$, $M,\langle D_1,\ldots,D_p,D_1\rangle\langle E_1,\ldots,E_q,E_{q+1},\ldots,E_{2q}\rangle \leadsto \phi_1\otimes\phi_2\otimes\ldots\otimes\phi_k$ and consequently, $M,\langle D_1,\ldots,D_n\rangle\langle E_1,\ldots,E_m\rangle \models \phi_1\otimes\phi_2\otimes\ldots\otimes\phi_k$

Finally since this is true for every model M of P it follows that

$$P,(D_1,\ldots,D_n),(E_1,\ldots,E_m) \models \phi_1\otimes\ldots\otimes\phi_k$$

2. Suppose that $\mathcal{O}^d(D_1) \models \phi_1$. Then for every M, $M,\langle D_1\rangle \models \phi_1$. Moreover, since $P,(D_1,\ldots,D_n),(E_1,\ldots,E_m) \models \phi_2\otimes\ldots\otimes\phi_k$, for every model M of P:

By Definition 7 there are two possible cases in this situation:
a) Serial Conjunction Case:
$M,\langle D_1,\ldots,D_n\rangle\langle E_1,\ldots,E_m\rangle \models \phi_2\otimes\ldots\otimes\phi_k$
$M,\langle D_1\rangle\langle E\rangle \models \phi_1$ by Definition 2
$M,\langle D_1,\ldots,D_n\rangle\langle E_1,\ldots,E_m\rangle \models \phi_1\otimes\ldots\otimes\phi_k$ by Definition 7 (Serial Conjunction Case)

b) Compensating Case:
$\langle D_1, \ldots, D_n\rangle = \langle D_1, \ldots, D_p, D_1, \ldots, D_r\rangle$
and $\langle E_1, \ldots, E_m\rangle = \langle E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q}, E_{2q+1}, \ldots, E_s\rangle$,
$M, \langle D_1, \ldots, D_p, D_1\rangle\langle E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q}\rangle \rightsquigarrow \phi_2 \otimes \ldots \otimes \phi_k$
and $M, \langle D_1, \ldots, D_r\rangle\langle E_{2q}, \ldots, E_s\rangle \models \phi_2 \otimes \ldots \otimes \phi_k$

So, we need to show exactly the same property but now for $M, \langle D_1, \ldots, D_r\rangle\langle E_{2q}, \ldots, E_s\rangle \models \phi_2 \otimes \ldots \otimes \phi_k$. Again in this case, there are two possibilities that can be applied. However, the path $\langle D_1, \ldots, D_r\rangle\langle E_{2q}, \ldots, E_s\rangle$ is strictly smaller than the original $\langle D_1, \ldots, D_n\rangle\langle E_1, \ldots, E_m\rangle$. As a result, by induction on the length of the path, we know that the path will eventually fall into the Serial Conjunction Case in order to be satisfied, and thus that $M, \langle D_1, \ldots, D_r\rangle\langle E_{2q}, \ldots, E_s\rangle \models \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$. Furthermore, since $(\phi_1 \otimes \ldots \otimes \phi_k) \rightarrow (\phi_1 \otimes \ldots \otimes \phi_k)$ and $M, \langle D_1\rangle, \langle E\rangle \models \phi_1$, then $M, \langle D_1, \ldots, D_p, D_1\rangle\langle E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q}\rangle \rightsquigarrow \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$ and consequently, $M, \langle D_1, \ldots, D_n\rangle\langle E_1, \ldots, E_m\rangle \models \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$

Finally since this is true for every model M of P it follows that

$$P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \models \phi_1 \otimes \ldots \otimes \phi_k$$

3. Suppose $\mathcal{O}^t(D_1, D_2) \models \phi_1$ and $P, (D_2, \ldots, D_n), (E_1, \ldots, E_m) \models \phi_2 \otimes \ldots \otimes \phi_k$. Then for every model M of P:

By Definition 7 there are two possible cases in this situation:
a) Serial Conjunction Case:
$M, \langle D_2, \ldots, D_n\rangle\langle E_1, \ldots, E_m\rangle \models \phi_2 \otimes \ldots \otimes \phi_k$
$M, \langle D_1, D_2\rangle\langle E_1\rangle \models \phi_1$ — By Definition 2
$M, \langle D_1, \ldots, D_n\rangle\langle E_1, \ldots, E_m\rangle \models \phi_1 \otimes \ldots \otimes \phi_k$ — By Definition 7 (Serial Conjunction Case)

b) Compensating Case:
$\langle D_2, \ldots, D_n\rangle = \langle D_2, \ldots, D_p, D_2, \ldots D_r\rangle$
and $\langle E_1, \ldots, E_m\rangle = \langle E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q}, E_{2q+1} \ldots, E_s\rangle$,
$M, \langle D_2, \ldots, D_p, D_2\rangle\langle E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q}\rangle \rightsquigarrow \phi_2 \otimes \ldots \otimes \phi_k$
and $M, \langle D_2, \ldots, D_r\rangle\langle E_{2q}, \ldots, E_s\rangle \models \phi_2 \otimes \ldots \otimes \phi_k$

So, we need to show exactly the same property but now for $M, \langle D_2, \ldots, D_r\rangle\langle E_{2q}, \ldots, E_s\rangle \models \psi_1 \otimes \ldots \otimes \psi_j \otimes \phi_2 \otimes \ldots \otimes \phi_k$. Again in this case, there are two possibilities that can be applied. However, the path $\langle D_2, \ldots, D_r\rangle\langle E_{2q}, \ldots, E_s\rangle$ is strictly smaller than the original $\langle D_2, \ldots, D_n\rangle\langle E_1, \ldots, E_m\rangle$. As a result, by induction on the length of the path, we know that the path will eventually fall into the Serial Conjunction Case in order to be satisfied, and thus that $M, \langle D_1, \ldots, D_r\rangle\langle E_{2q}, \ldots, E_s\rangle \models \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$.
Furthermore, since $(\phi_1 \otimes \ldots \otimes \phi_k) \rightarrow (\phi_1 \otimes \ldots \otimes \phi_k)$, and since $M, \langle D_1, D_2\rangle\langle E\rangle \models \phi_1$. By the serial conjunction case we know that $M, \langle D_1, \ldots, D_p, D_1\rangle\langle E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q}\rangle \rightsquigarrow \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$ and consequently, $M, \langle D_1, \ldots, D_n\rangle\langle E_1, \ldots, E_m\rangle \models \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$

Finally since this is true for every model M of P it follows that

$$P,(D_1,\ldots,D_n),(E_1,\ldots,E_m) \models \phi_1 \otimes \ldots \otimes \phi_k$$

4. Suppose $\mathcal{O}^e(E_1,E_2) \models \phi_1$ and $P,(D_1,\ldots,D_n),(E_2,\ldots,E_m) \models \phi_2 \otimes \ldots \otimes \phi_k$. Then for every model M of P:

By Definition 7 there are two possible cases in this situation:
a) Serial Conjunction Case:
$M,\langle D_1,\ldots,D_n\rangle\langle E_2,\ldots,E_m\rangle \models \phi_2 \otimes \ldots \otimes \phi_k$
$M,\langle D_1\rangle\langle E_1,E_2\rangle \models \phi_1$ By Definition 2
$M,\langle D_1,\ldots,D_n\rangle\langle E_1,\ldots,E_m\rangle \models \phi_1 \otimes \ldots \otimes \phi_k$ By Definition 7

b) Compensating Case:
$\langle D_1,\ldots,D_n\rangle = \langle D_1,\ldots,D_p,D_1,\ldots,D_r\rangle$ and
$\langle E_2,\ldots,E_m\rangle = \langle E_2,\ldots,E_q,E_{q+1},\ldots,E_{2q},E_{2q+1}\ldots,E_s\rangle$,
$M,\langle D_1,\ldots,D_p,D_1\rangle\langle E_1,\ldots,E_q,E_{q+1},\ldots,E_{2q}\rangle \rightsquigarrow \phi_2 \otimes \ldots \otimes \phi_k$
and $M,\langle D_1,\ldots,D_r\rangle\langle E_{2q},\ldots,E_s\rangle \models \phi_2 \otimes \ldots \otimes \phi_k$

So, we need to show exactly the same property but now for $M,\langle D_1,\ldots,D_r\rangle\langle E_{2q},\ldots,E_s\rangle \models \psi_1 \otimes \ldots \otimes \psi_j \otimes \phi_2 \otimes \ldots \otimes \phi_k$. Again in this case, there are two possibilities that can be applied. However, the path $\langle D_1,\ldots,D_r\rangle\langle E_{2q},\ldots,E_s\rangle$ is strictly smaller than the original $\langle D_1,\ldots,D_n\rangle\langle E_2,\ldots,E_m\rangle$. As a result, by induction on the length of the path, we know that the path will eventually fall into the Serial Conjunction Case in order to be satisfied, and thus that $M,\langle D_1,\ldots,D_r\rangle\langle E_{2q},\ldots,E_s\rangle \models \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$.
Furthermore, since $(\phi_1 \otimes \ldots \otimes \phi_k) \rightarrow (\phi_1 \otimes \ldots \otimes \phi_k)$, and since $M,\langle D\rangle\langle E_1,E_2\rangle \models \phi_1$. By the serial conjunction case we know that $M,\langle D_1,\ldots,D_p,D_1\rangle\langle E_1,\ldots,E_q,E_{q+1},\ldots,E_{2q}\rangle \rightsquigarrow \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$ and consequently, $M,\langle D_1,\ldots,D_n\rangle\langle E_1,\ldots,E_m\rangle \models \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$
Finally since this is true for every model M of P it follows that

$$P,(D_1,\ldots,D_n),(E_1,\ldots,E_m) \models \phi_1 \otimes \ldots \otimes \phi_k$$

5. Since a consistency preserving derivation can be constructed from φ, then by Theorem 4 we know that $P,(D_1,\ldots,D_k,D_1),(E_1,\ldots,E_l,E_{l+1},\ldots,E_{2l}) \rightsquigarrow \varphi$ and that $P,(D_1,\ldots,D_k,D_1,\ldots,D_n)(E_1,\ldots,E_l,E_{l+1},\ldots,E_{2l},E_{2l+1},\ldots,E_m) \models \varphi$ then for all models M of P. If this is the case, and since φ does not hold in the path $(D_1,\ldots,D_k,D_1),(E_1,\ldots,E_l,E_{l+1},\ldots,E_{2l})$ then by Definition 7 the only possibility is to apply the Compensating Case and thus $P,(D_1,\ldots,D_n)(E_{2l},\ldots,E_m) \models \varphi$ must hold.

□

Theorem 4 (Soundness $\mid\!\rightsquigarrow$).
If $P,(D_1,\ldots,D_n),(E_1,\ldots,E_m),(A_1,\ldots,A_{m-1}) \mid\!\rightsquigarrow \varphi$ *then* $P,(D_1,\ldots,D_n),(E_1,\ldots,E_m),(A_1,\ldots,A_{m-1}) \rightsquigarrow \varphi$

Proof. Immediately from Lemma 5 and Lemma 6

Lemma 5. *Suppose there exists a successful consistency resolution with root φ with the first part of the sequence ending in $P, (D_1, \ldots, D_n), (E_1, \ldots E_m), (A_1, \ldots, A_{m-1}) \vdash' \psi_1 \otimes \ldots \otimes \psi_j$; and the second part of the sequence ending in $P, (D_n), (E_m, \ldots, E_{2m}), (A_{m-1}^{-1}, \ldots, A_1^{-1}) \vdash ()$. Then $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_{2m})(A_1, \ldots, A_{m-1}, A_{m-1}^{-1}, \ldots, A_1^{-1}) \rightsquigarrow \varphi$*

Proof. From rules (1-4) of Definition 14 $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m), (A_1, \ldots, A_{m-1}) \vdash' \psi_1 \otimes \ldots \otimes \psi_j$ is only a valid sequence if $(\psi_1 \otimes \ldots \otimes \psi_j) \rightarrow \varphi$. Furthermore, if the first sequence ends in $\psi_1 \otimes \ldots \otimes \psi_j$, then no rule can be applied to ψ_1. Since $\vdash'$ does not use compensating techniques, then it has a direct correspondence to $\mathcal{TR}$ inference system which is proven to be sound and complete. As a result, since no rule can be applied, then φ fails in the path $(D_1, \ldots, D_n)$. Moreover, if the second sequence ends in $P, (D_n), (E_m, \ldots, E_{2m}), (A_{m-1}^{-1}, \ldots, A_1^{-1}) \vdash ()$ then it means that all the compensations of the external actions can be performed evolving E_m into E_{2m}. Then by Definition 8:
$P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m})(A_1, \ldots, A_{m-1}, A_{m-1}^{-1}, \ldots, A_1^{-1}) \rightsquigarrow \varphi$ holds. □

Lemma 6. *Given a derivation $\mid\!\rightsquigarrow$ with goal φ.*

1. *Suppose a rule $\phi_1 \leftarrow \psi_1 \otimes \ldots \otimes \psi_j$ exists in P.*
   If $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \psi_1 \otimes \ldots \otimes \psi_j \otimes \phi_2 \otimes \ldots \otimes \phi_k$ implies
   $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \rightsquigarrow \varphi$
   then $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_1 \otimes \ldots \otimes \phi_k$ also implies
   $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \rightsquigarrow \varphi$
2. *Suppose that $\mathcal{O}^d(D_1) \models \phi_1$.*
   If $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_2 \otimes \ldots \otimes \phi_k$ implies
   $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \rightsquigarrow \varphi$
   then $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_1 \otimes \ldots \otimes \phi_k$ also implies
   $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \rightsquigarrow \varphi$
3. *Suppose $\mathcal{O}^t(D_1, D_2) \models \phi_1$.*
   If $P, (D_2, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_2 \otimes \ldots \otimes \phi_k$ implies
   $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \rightsquigarrow \varphi$
   then $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_1 \otimes \ldots \otimes \phi_k$ also implies
   $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \rightsquigarrow \varphi$
4. *Suppose that $\mathcal{O}^e(E_1, E_2) \models \phi_1$.*
   If $P, (D_1, \ldots, D_n), (E_2, \ldots, E_m) \vdash' \phi_2 \otimes \ldots \otimes \phi_k$ implies
   $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \rightsquigarrow \varphi$
   then $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_1 \otimes \ldots \otimes \phi_k$ also implies
   $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \rightsquigarrow \varphi$

Proof. We prove each item in turn

1. Note that since $\phi_1 \leftarrow \psi_1 \otimes \ldots \otimes \psi_j$ is in P, then every model of P is also a model of $\phi_1 \leftarrow \psi_1 \otimes \ldots \otimes \psi_j$.
   So assume that $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \psi_1 \otimes \ldots \otimes \psi_j \otimes \phi_2 \otimes \ldots \otimes \phi_k$ implies $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \rightsquigarrow \varphi$. Then assume

$P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$. If this is the case, then Rule 1 from Definition 14 can be applied, and thus $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \psi_1 \otimes \ldots \otimes \psi_j \otimes \phi_2 \otimes \ldots \otimes \phi_k$ which implies $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \leadsto \varphi$.

2. Note that since $\mathcal{O}^d(D_1) \models \phi_1$, then for every model $M\langle D_1\rangle\langle\rangle \models \phi_1$.
   So assume that $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_2 \otimes \ldots \otimes \phi_k$ implies $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \leadsto \varphi$. Then assume $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$. If this is the case, then Rule 2 from Definition 14 can be applied, and thus $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \psi_1 \otimes \ldots \otimes \psi_j \otimes \phi_2 \otimes \ldots \otimes \phi_k$ which implies $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \leadsto \varphi$.
3. So assume that $P, (D_2, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_2 \otimes \ldots \otimes \phi_k$ implies $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \leadsto \varphi$. Then assume $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$. If this is the case, then Rule 3 from Definition 14 can be applied, and thus $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \psi_1 \otimes \ldots \otimes \psi_j \otimes \phi_2 \otimes \ldots \otimes \phi_k$ which implies $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \leadsto \varphi$.
4. So assume that $P, (D_1, \ldots, D_n), (E_2, \ldots, E_m) \vdash' \phi_2 \otimes \ldots \otimes \phi_k$ implies $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \leadsto \varphi$. Then assume $P, (D_1, \ldots, D_n), (E_2, \ldots, E_m) \vdash' \phi_1 \otimes \phi_2 \otimes \ldots \otimes \phi_k$. If this is the case, then Rule 4 from Definition 14 can be applied, and thus $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash' \psi_1 \otimes \ldots \otimes \psi_j \otimes \phi_2 \otimes \ldots \otimes \phi_k$ which implies $P, (D_1, \ldots, D_n, D_1), (E_1, \ldots, E_m, E_{m+1}, \ldots, E_{2m}) \leadsto \varphi$.

□

Lemma 7. *Let $(D_1, \ldots, D_n), (E_1, \ldots, E_m)$ be a path that satisfies a formula ϕ without the execution of compensating strategies.*

If $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \models \phi$ *then* $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash \phi$

Proof. Without the execution of compensating strategies, $\mathcal{ETR}$ Proof Theory corresponds with $\mathcal{TR}$ Proof Theory with an additional case for point 4 of Definition 14. Since $\mathcal{TR}$ Proof Theory is complete w.r.t. $\models$, so is $\mathcal{ETR}$ Proof theory without the execution of compensations with the addition of point 4.

Lemma 8. *Let $(D_1, \ldots, D_k, D_1), (E_1, \ldots, E_{2j})(A_1, \ldots, A_{j-1}, A_{j-1}^{-1}, \ldots, A_1^{-1})$ be a consistency preserving path for ϕ of length 1, i.e., only one compensation is performed for ϕ in the given path.*
If $P, (D_1, \ldots, D_k, D_1), (E_1, \ldots, E_{2j}), (A_1, \ldots, A_{j-1}, A_{j-1}^{-1}, \ldots, A_1^{-1}) \leadsto \phi$ *then*
$P, (D_1, \ldots, D_k, D_1), (E_1, \ldots, E_{2j}), (A_1, \ldots, A_{j-1}, A_{j-1}^{-1}, \ldots, A_1^{-1}) \mid\!\!\leadsto \phi$

Proof. This proof comes almost immediately from Definition 8, Definition 15 and Lemma 7:
$P, (D_1, \ldots, D_k, D_1), (E_1, \ldots, E_j), (A_1, \ldots, A_{j-1}, A_{j-1}^{-1}, \ldots, A_1^{-1}) \leadsto \phi$ is true if
$\exists b_1 \otimes \ldots \otimes b_i \otimes \ldots \otimes b_n$ such that:
$P, (D_1, \ldots, D_k), (E_1, \ldots, E_j), (A_1, \ldots, A_{j-1}) \models b_1 \otimes \ldots \otimes b_i \otimes \ldots \otimes b_n \rightarrow \phi$
$P, (D_1, \ldots, D_k), (E_1, \ldots, E_j), (A_1, \ldots, A_{j-1}) \models b_1 \otimes \ldots \otimes b_{i-1}$
$P, (D_1, \ldots, D_k), (E_1, \ldots, E_j), (A_1, \ldots, A_{j-1}) \not\models b_i$
$P, (D_1, \ldots, D_k), (E_j, \ldots, E_{2j}), (A_{j-1}^{-1}, \ldots, A_1^{-1}) \models A_{j-1}^{-1} \otimes \ldots \otimes A_1^{-1}$

Since $\models$ without compensations is complete w.r.t. $\vdash$ then:
$P,(D_1,\ldots,D_k),(E_1,\ldots,E_j),(A_1,\ldots,A_{j-1}) \vdash b_1 \otimes \ldots \otimes b_i \otimes \ldots \otimes b_n \rightarrow \phi$
$P,(D_1,\ldots,D_k),(E_1,\ldots,E_j),(A_1,\ldots,A_{j-1}) \vdash b_1 \otimes \ldots \otimes b_{i-1}$
$P,(D_1,\ldots,D_k),(E_1,\ldots,E_j),(A_1,\ldots,A_{j-1}) \not\vdash b_i$
$P,(D_1,\ldots,D_k),(E_j,\ldots,E_{2j}),(A_{j-1}^{-1},\ldots,A_1^{-1}) \vdash A_{j-1}^{-1} \otimes \ldots \otimes A_1^{-1}$
Now by Definition 15 we know that
$P,(D_1,\ldots,D_k,D_1),(E_1,\ldots,E_{2j})(A_1,\ldots,A_{j-1},A_{j-1}^{-1},\ldots,A_1^{-1}) \mid\!\!\leadsto \phi$. $\square$

Definition 16. *$P,(D_1,\ldots,D_j,D'_1,\ldots,D'_k),(E_1,\ldots,E_p,E'_1,\ldots,E'_q) \mid\!\!\leadsto^* \phi$ if $P,(D_1,\ldots,D_j),(E_1,\ldots,E_p) \mid\!\!\leadsto \phi$ and $P,(D'_1,\ldots,D'_k),(E'_1,\ldots,E'_q) \mid\!\!\leadsto \phi$; or $P,(D_1,\ldots,D_j),(E_1,\ldots,E_p) \mid\!\!\leadsto^* \phi$ and $P,(D'_1,\ldots,D'_k),(E'_1,\ldots,E'_q) \mid\!\!\leadsto \phi$*

Lemma 9. *Completeness $\leadsto$ Let $(D_1,\ldots,D_n),(E_1,\ldots,E_m)$ be a consistency preserving path of an arbitrary length n for ϕ. We say that*
If $P,(D_1,\ldots,D_n),(E_1,\ldots,E_m) \leadsto \phi$ then $P,(D_1,\ldots,D_n),(E_1,\ldots,E_m) \mid\!\!\leadsto^* \phi$

Proof. We prove by induction on the length of the compensation. The base case comes immediately from Lemma 8. For the inductive case assume that the claim is true for n compensations of ϕ. We show that if it exists, then the claim is true for $n+1$. So, if there exists $n+1$ compensations for ϕ:
$P,(D_{1_1},\ldots,D_{j_1},\ldots,D_{1_{n+1}},\ldots,D_{j_{n+1}}),(E_{1_1},\ldots,E_{k_1},\ldots,E_{1_n},\ldots,E_{1_{n+1}},\ldots,E_{k_{n+1}}) \leadsto \phi$ However, we have $P,(D_{1_1},\ldots,D_{j_1},\ldots,D_{1_n},\ldots,D_{j_n},E_{1_1},\ldots,E_{k_1},\ldots,E_{1_n},\ldots,E_{k_n}) \leadsto \phi$ and so $P,(D_{1_1},\ldots,D_{j_1},\ldots,D_{1_n},\ldots,D_{j_n},E_{1_1},\ldots,E_{k_1},\ldots,E_{1_n},\ldots,E_{k_n}) \mid\!\!\leadsto^* \phi$. Furthermore since $P,(D_{1_{n+1}},\ldots,D_{j_{n+1}}),(E_{1_{n+1}},\ldots,E_{k_{n+1}}) \leadsto \phi$ is a compensation of length 1, then by Lemma 8 it exists $P,(D_{1_{n+1}},\ldots,D_{j_{n+1}})(E_{1_{n+1}},\ldots,E_{k_{n+1}}) \mid\!\!\leadsto \phi$ and thus by Definition 16 we know that the following formula holds:
$P,(D_{1_1},\ldots,D_{j_1},\ldots,D_{1_{n+1}},\ldots,D_{j_{n+1}}),(E_{1_1},\ldots,E_{k_1},\ldots,E_{1_{n+1}},\ldots,E_{k_{n+1}}) \mid\!\!\leadsto^* \phi$ $\square$

Theorem 5 (Completeness). *Let P be a ground transaction program with external actions (possibly infinite) and ϕ is a serial-Horn goal:*
If $P,(D_1,\ldots,D_n),(E_1,\ldots,E_m) \models \phi$ then $P,(D_1,\ldots,D_n),(E_1,\ldots,E_m) \vdash \phi$

Proof. ϕ has the form $b_1 \otimes \ldots \otimes b_k$, where $k \geq 0$ and each b_i is a ground atomic formula. Our proof is by induction on k. In the base case, $k = 0$ and ϕ is the empty clause $()$. If the expression $P,(D_1,\ldots,D_n),(E_1,\ldots,E_m) \models ()$ is true, then $n = 1$ and $m = 0$, since the empty clause is true only on states. But, the sequent $P,D_1,() \vdash ()$ is trivially true, since it is an axiom.

For the inductive case, assume the claim is true for all values of k from 0 to j. We show that it is true for $k = j + 1$:

$P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \models b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$
$M, \langle D_1, \ldots, D_n \rangle \langle E_1, \ldots, E_m \rangle \models b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$
Now by Definition 7 there are two cases:
a) Serial Conjunction Case:
And thus no compensation is performed. Since $\mathcal{TR}$ proof theory is complete, then:
$P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$

b) Compensating Case:
$\langle D_1, \ldots, D_n \rangle = \langle D_1, \ldots, D_p, D_1, \ldots, D_k \rangle$
and $\langle E_1, \ldots, E_m \rangle = \langle E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q}, E_{2q+1}, \ldots, E_l \rangle$
$M, \langle D_1, \ldots, D_p, D_1 \rangle \langle E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q} \rangle \rightsquigarrow b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$
and $M, \langle D_1, \ldots, D_p \rangle \langle E_{2q}, \ldots, E_l \rangle \models b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$

Now we can apply induction on the length of the path that satisfies the formula $b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$. In fact, we need to show exactly the same property, but now for a path $\langle D_1, \ldots, D_k \rangle$ that is strictly smaller than the initial path $\langle D_1, \ldots, D_n \rangle$. Again in this case, there are two possibilities that can be applied. But now, since we are reducing the size of the path we know that in order to satisfy the formula we will eventually fall into the case a), and thus $P, (D_1, \ldots, D_k), (E_{2q}, \ldots, E_l) \vdash b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$.

Furthermore, since operator $\rightsquigarrow$ is complete w.r.t. $\vdash\!\rightsquigarrow^*$ (Lemma 9): if $M, \langle D_1, \ldots, D_p, D_1 \rangle \langle E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q} \rangle \rightsquigarrow b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$ then $P, (D_1, \ldots, D_p, D_1), (E_1, \ldots, E_q, E_{q+1}, \ldots, E_{2q}) \vdash\!\rightsquigarrow^* b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$. This means that we can obtain the compensating path $\langle D_1, \ldots, D_p, D_1 \rangle \langle E_1 \ldots E_{2q} \rangle$ by applying the operator $\vdash\!\rightsquigarrow$ a finite amount of times[5]. So by Definition 14, we know that $P, (D_1, \ldots, D_n), (E_1, \ldots, E_m) \vdash b_1 \otimes \ldots \otimes b_j \otimes b_{j+1}$ is true. □

6 Related Work and Conclusions

$\mathcal{ETR}$ can be compared to many logics that reason about state change or about the related phenomena of time and action. These include action languages, the situation calculus, the event calculus, process logic and many others. However, although one can compare how these formalisms can model and represent transactions with Transaction Logic (and for an extensive comparison of these formalisms with $\mathcal{TR}$ the interested reader is referred to [2]), such comparisons are mainly unfair. Particularly, this kind of logics was not design to reason about database programs but rather intended to describe changes in dynamic systems where one has little or no control such as external domains. As a result, although these formalisms provide powerful tools to specify changes and reason about their causalities in a very general and abstract way, they are inappropriate to model database transactions [2]. Nevertheless, as argued throughout the paper, these formalisms are orthogonal to $\mathcal{ETR}$ theory. As a result, it is our opinion that rather than an alternative to $\mathcal{ETR}$, these solutions should be seen as a possible built-in component.

[5] particularly, the same as the amount of the compensations performed

On the other hand, one can also compare $\mathcal{ETR}$ to formalisms that involve the notion of long-running transactions or sagas. Generally such formalisms are based on process algebras, a family of algebraic systems for modeling concurrent communicating processes, as Milner's Calculus of Communicating Systems (CCS) and Hoare's Communicating Sequential Processes (CSP), among others. One clear difference between $\mathcal{ETR}$ and such systems is that $\mathcal{ETR}$ does not support concurrency and synchronization. However, extending $\mathcal{ETR}$ to provide such features represents a next obvious step and is in line with what has been done in Concurrent Transaction Logic [3].

Notwithstanding the major difference between $\mathcal{ETR}$ and other proposals based on process algebras as [9,12] is mainly conceptual. In fact, the semantics of these latter systems are mostly focused about the correct execution and synchronization of processes whilst $\mathcal{ETR}$ semantics emphasizes knowledge base states. As a result, process algebras solutions are interested in modeling the correctness evolution of each transaction, thereby possessing a powerful operational semantics, but they disregard the evolution of the knowledge base itself as there is no model theory available. In this sense, such solutions enclose powerful operators that in some cases even allow the system to construct the correct compensation for each action "on-the-fly" as in [14]. However, these solutions are not suitable to be used as a knowledge representation formalism. Consequently, it is not possible to model what is true at each step of the execution of these processes nor to specify constraints on their execution based on this knowledge.

This work represents a first step towards a unifying logical framework able to combine a strict ACID transactions with long-running/relaxed model of transactions for hybrid evolving systems. That is, systems that have both an internal and an external updatable component and require properties on the outcomes of the updates. Examples of these systems range from an intelligent agent that has an internal knowledge base where he executes reasoning, but is integrated in an evolving external world where he can execute actions; to web-based systems with an internal knowledge base that follows the strict ACID model, but also need to interact with other systems, for instance to request a web-service. Closely related to this latter domain, is that of reactive (semantic) web languages. The Semantic Web initiative of W3C has recently proposed RIF-PRD [5] as a recommendation of a reactive (prodution-rule-like) rule language. This languages is intended to exchange rules that execute actions in hybrid web systems reactively, but still without concerns on guaranteeing transaction properties on the outcome of these actions. Given $\mathcal{ETR}$ declarative semantics and its natural model theory, we believe that it can be suitable to provide transaction properties for web-reactive systems.

Another important line of research is to further study the flexibility provided by having the oracles as a parameter of the theory. Particularly, how to take advantage of augmenting the program P with rules from the oracles. This adoption provides new challenges and difficulties, as it comes with the demand for new reasoning algorithms that go beyond the serial-Horn limitations.

References

1. J. J. Alferes, A. Brogi, J. A. Leite, and L. M. Pereira. Evolving logic programs. In *JELIA*, pages 50–61, 2002.

2. A. J. Bonner and M. Kifer. Transaction logic programming (or a logic of declarative and procedural knowledge). Technical Report CSRI-323, Computer Systems Research Institute, University of Toronto, 1995.
3. A. J. Bonner and M. Kifer. Concurrency and communication in transaction logic. In *JICSLP*, pages 142–156, 1996.
4. A. J. Bonner and M. Kifer. Results on reasoning about updates in transaction logic. In *Transactions and Change in Logic Databases*, pages 166–196, 1998.
5. C. de Sainte Marie, G. Hallmark, and A. Paschke. RIF Production Rule Dialect, June 2010. W3C Recommendation `http://www.w3.org/TR/rif-prd/`.
6. H. Garcia-Molina and K. Salem. Sagas. *SIGMOD Rec.*, 16:249–259, December 1987.
7. M. Gelfond and V. Lifschitz. Action languages. *Electron. Trans. Artif. Intell.*, 2:193–210, 1998.
8. D. Harel, D. Kozen, and R. Parikh. Process logic: Expressiveness, decidability, completeness. In *FOCS*, pages 129–142, 1980.
9. C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
10. R. A. Kowalski and M. J. Sergot. A logic-based calculus of events. *New Generation Comput.*, 4(1):67–95, 1986.
11. J. McCarthy. Situations, actions, and causal laws. Technical report, Stanford University, 1963. Reprinted in Semantic Information Processing, MIT Press, Cambridge, Mass., 1968 pages 410-417.
12. R. Milner. Calculi for synchrony and asynchrony. *Theor. Comput. Sci.*, 25:267–310, 1983.
13. A. Van Gelder, K. A. Ross, and J. S. Schlipf. The well-founded semantics for general logic programs. *J. ACM*, 38:619–649, July 1991.
14. C. Vaz and C. Ferreira. Towards compensation correctness in interactive systems. In *WS-FM*, pages 161–177, 2009.