World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:1, No:9, 2007

Specifying Strict Serializability of Iterated Transactions in Propositional Temporal Logic

Walter Hussak

International Science Index, Mathematical and Computational Sciences Vol:1, No:9, 2007 waset.org/Publication/5190

Abstract— **We present an operator for a propositional linear temporal logic over infinite schedules of iterated transactions, which, when applied to a formula, asserts that any schedule satisfying the formula is serializable. The resulting logic is suitable for specifying and verifying consistency properties of concurrent transaction management systems, that can be defined in terms of serializability, as well as other general safety and liveness properties. A strict form of serializability is used requiring that, whenever the read and write steps of a transaction occurrence precede the read and write steps of another transaction occurrence in a schedule, the first transaction must precede the second transaction in an equivalent serial schedule. This work improves on previous work in providing a propositional temporal logic with a serializability operator that is of the same PSPACE-complete computational complexity as standard propositional linear temporal logic without a serializability operator.**

Index Terms— **Temporal Logic, Iterated Transactions, Serializability.**

I. Introduction

The model of concurrent iterated transactions, where transactions repeat infinitely often, was originally considered in [4] because of its applicability to the scheduling problem of service processes in operating systems. The behavior of such systems is an infinite schedule and the consistency condition a generalization of the familiar serializability condition for finite schedules of database transactions [3]. In fact, even in the case of concurrent transaction management for standard database systems, it is more accurate and assumption-free to model the output of schedulers as sets of infinite schedules. Infinite schedules have also acquired a greater significance with the advent of the newer technologies of web and mobile transactions in which transactions are continuously accessing data items. Despite this, there have been only a few attempts to address the problem of proving serializability of infinite schedules. Existing approaches advocate the use of temporal logic [12] for specifying infinite schedules generated by a scheduler, as models of temporal logic formulae. For example, the work [11] defines a partial-order temporal logic over trace models for specifying properties of schedules such as serializability. Also, the work [7] allows infinite schedules to be specified in a linear temporal logic. The problem with both of these approaches is that the only viable method of proof of conditions such as serializability is one using proof rules. In the work [11] an axiomatization is given for this purpose. Although no explicit axiomatization is given in the work [7], serializability is encoded into the Quantified Propositional Temporal Logic (*QPTL*) which is axiomatizable - however, no practical alternative based on a decision procedure is possible as *QPTL* has non-elementary computational complexity. The drawback of conducting proofs using proof rules is that they require considerable expertise by the person who is to carry out the proof manually, perhaps with the help of a 'proof assistant' tool. One of the attractions of certain temporal logics in computer science is their favorable computational complexity as compared to classical (non-temporal) logics that have the same expressiveness. For example, the validity problem for Propositional Linear Temporal Logic (*PTL*) is PSPACE-complete whereas the validity problem for a classical equivalent is non-elementary. This has led to the development of industrial-strength fully automatic theorem provers, such as NuSMV [1] and SPIN [6], for commonly used such temporal logics. With this in mind, the ideal solution to proving serializability of infinite schedules would be one that could utilize these logics efficiently.

Numerous variants of serializability have been proposed as the appropriate consistency condition in various circumstances and for various reasons in the case of finite schedules of concurrent transactions, for example [10], [15], [13] and [8]. In the case of the infinite schedules that result from concurrent iterated transactions an extension of conflict serializability to unbounded schedules, based on that used for the case of finite schedules of fixed length, is defined in [4] and weaker versions given in the work [5]. Conflict serializability is characterized by the commutativity of non-conflicting operations and forms of commutativity-based serializability are discussed in [11] and [9] with regard to the partial-order temporal logics that can be used to specify them. Some non-commutative forms of serializability for infinite schedules are specified in [7] making use of propositional quantification which, however, is responsible for the non-elementary complexity of the logic. In this paper, we seek a notion of serializability for infinite schedules, that can be expressed easily and efficiently in a temporal logic for which fully automatic theorem provers exist. To this end, we will consider the notion of 'serializability in the strict sense' from [10] or 'strict serializability' as we shall refer to it. Strict serializability has the following motivation. It is observed in [10], that certain schedules have a curious, maybe undesirable, property. Consider the following schedule:

$$R_1[x, y]R_2[y]W_2[y]R_3[x, z]W_3[z]W_1[x]$$

where the R's denote read steps, the W's write steps, subscripts identify transactions and the brackets denote the data items

Walter Hussak is a lecturer in the Department of Computer Science, Loughborough University, UK (Email : W.Hussak@lboro.ac.uk, Tel. : +44(0)1509 222937, Fax: +44(0)1509 211586).

accessed. This schedule serializes to the schedule:

$$R_3[x,z]W_3[z]R_1[x,y]W_1[x]R_2[y]W_2[y]$$

In the first schedule, transaction 2 has completed execution before transaction 3 has even started execution, yet the only serialized order has transaction 3 appearing before transaction 2. This undesirable property could be compounded in the case of an infinite schedule where any number of iterations of transaction 2 could execute before an occurrence of transaction 3, yet the only serialized order would have all those occurrences of transaction 2 coming after that single occurrence of transaction 3. Strict serializability does not allow such a serialization.

This paper is structured as follows. In section II, we extend strict serializability to the case of infinite schedules of infinitely repeating 2-step transactions and we give a test for a schedule to be strictly serializable that involves selecting an occurrence of each of the iterating transactions. This test is improved in section III by showing that only occurrences of a bounded subset of the iterating transactions, have to be considered. A strict serializability operator is then defined for propositional linear temporal logic in section IV and the extended logic is shown to be PSPACE-complete. We give concluding remarks in section V.

II. Strict serializability

In this section strict serializability is defined (Definitions 1-4), a condition that provides a test for strict serializability is given (Definitions 5,6), and then this condition is proved to correspond to strict serializability (Lemma 7 and Theorem 8).

The assumptions and notation for our 2-step transaction model are largely as in [7]. We assume n transactions $T_1,\dots,T_n$ where each T_i comprises a read step and a write step accessing finite sets of data items or variables denoted by $S(R_i)$ and $S(W_i)$ such that $S(W_i)\subseteq S(R_i)$, i.e. the write set is a subset of the read set. If $S(R_i)=\{y_1,\dots,y_p\}$ and $S(W_i)=\{y'_1,\dots,y'_q\}$ we shall display the read and write steps as $R_i[y_1\dots y_p]$ and $W_i[y'_1\dots y'_q]$ respectively. We shall omit the [ ] brackets if the variables accessed are of no interest and use the notation $R_i[x.]$ and $W_i[x.]$ to indicate a step that accesses x and may access other variables. The finite set of all variables accessed by the T'_is will be $\{x_1,\dots,x_m\}$. A *schedule* or *history* for $T_1,\dots,T_n$ is an interleaved sequence h of the read and write steps of infinitely many occurrences of the T'_is, such that the subsequence of h comprising steps of T_i is the infinitely repeating sequence

$$R_iW_iR_iW_i\dots$$

Different occurrences of steps will be labelled by adding an extra subscript as in the following history

$$R_{11}R_{21}W_{11}W_{21}R_{12}R_{22}W_{12}W_{22}\dots$$

The occurrence R_{ij} (respectively W_{ij}) will be called the *read (respectively write) step of the j-th occurrence T_{ij} of T_i*. In a history h, for each i there will be a positive integer e, not necessarily equal to 1, such that occurrences of T_i in h are labelled by consecutive integers starting at e. Then, T_{ie} will be referred to as the *earliest* occurrence of T_i in h. We shall write $T_{ij}\in h$ when occurrence T_{ij} belongs to h. For a history h, $<_h$ will be the (irreflexive) total order between all the read and write steps of h. If T_{ie} is the earliest occurrence of T_i in h, then $h-T_{ie}$ will denote the history with R_{ie} and W_{ie} removed. The history comprising R_{ie} followed by W_{ie} followed by the sequence $h-T_{ie}$ will be denoted $T_{ie}(h-T_{ie})$.

Strict serializability of an infinite history h means that it is 'equivalent', i.e. its read steps read the same write steps, to a serial history h_S such that, if the write step of a transaction occurrence precedes the read step of another transaction occurrence in h, those two transaction occurrences must be in the same order in h_S. We formalize this as follows.

Definition 1 Histories h_1 and h_2 are *equivalent*, written $h_1\sim h_2$, iff for $x\in\{x_1,\dots,x_m\}$ and read and write occurrences $R_{i_1j_1}$ and $W_{i_2j_2}$

$$sees^x_{h_1}(R_{i_1j_1},W_{i_2j_2})\ \text{ iff }\ sees^x_{h_2}(R_{i_1j_1},W_{i_2j_2})$$

where $sees^x_h(R_{i_1j_1},W_{i_2j_2})$ holds if h is of the form

$$\dots W_{i_2j_2}[x.]\underbrace{\dots\dots\dots\dots}_{\text{no writes to }x}R_{i_1j_1}[x.]\dots$$

Definition 2 A history h_S is *serial* iff it is of the form

$$R_{i_1j_1}W_{i_1j_1}R_{i_2j_2}W_{i_2j_2}\dots R_{i_mj_m}W_{i_mj_m}\dots$$

Definition 3 A history h_S is *strictly serial* with respect to h iff:

(i) h_S is serial
(ii) h_S has the same occurrences as h
(iii) if $W_{i_1j_1}<_h R_{i_2j_2}$ then $W_{i_1j_1}<_{h_S} R_{i_2j_2}$

Definition 4 A history h is *strictly serializable* iff there is a strictly serial history h_S such that $h\sim h_S$. It is easy to show that

$$R_{i_1j_1}[x.]<_h W_{i_2j_2}[x.]\ \text{ iff }\ R_{i_1j_1}[x.]<_{h_S} W_{i_2j_2}[x.] \qquad (1)$$

and

$$W_{i_1j_1}[x.]<_h W_{i_2j_2}[x.]\ \text{ iff }\ W_{i_1j_1}[x.]<_{h_S} W_{i_2j_2}[x.] \qquad (2)$$

The test for serializability that is encoded into temporal logic in [7] requires that any chosen set of occurrences of transactions in the history h has a 'detachable' occurrence. For strict serializability, the corresponding test requires an additional condition to produce a 'strictly detachable' occurrence, i.e. one whose read step cannot come after a write step in the chosen set of occurrences (see (iv) of Definition 5 below).

Definition 5 Let h be a history, p be an integer such that $1\le p\le n$ and $\{T_{i_1j_1},\dots,T_{i_pj_p}\}\subseteq\{T_1,\dots,T_n\}$. Then, (the sequence of read and write steps of) $T_{i_1j_1},\dots,T_{i_pj_p}$ is *strictly detachable* or *s-detachable* in h iff one of the occurrences $T_{i_kj_k}$, called a *s-detachable occurrence* in $T_{i_1j_1},\dots,T_{i_pj_p}$ is such that, for $1\le g\le p$, $g\ne k$, $x\in\{x_1,\dots,x_m\}$

(i) $\neg(R_{i_gj_g}[x.] <_h W_{i_kj_k}[x.])$
(ii) $\neg(W_{i_gj_g}[x.] <_h R_{i_kj_k}[x.])$
(iii) $\neg(W_{i_gj_g}[x.] <_h W_{i_kj_k}[x.])$
(iv) $\neg(W_{i_gj_g} <_h R_{i_kj_k})$

Definition 6 The condition *ssercond(h)* holds iff every sequence of occurrences $T_{i_1j_1}, \dots, T_{i_nj_n}$ as in Definition 5, such that $\{T_{i_1j_1}, \dots, T_{i_nj_p}\} = \{T_1, \dots, T_n\}$, is s-detachable.

We show that *ssercond* is indeed a necessary and sufficient condition for strict serializability to hold.

Lemma 7 Let h be a history with earliest occurrences $T_{i_1e_1}, \dots, T_{i_ne_n}$ such that *ssercond(h)* holds. Then, for some k with $1 \leq k \leq n$,

(i) $T_{i_ke_k}$ is a s-detachable occurrence
(ii) $h \sim T_{i_ke_k}(h - T_{i_ke_k})$
(iii) *ssercond*$(h - T_{i_ke_k})$ holds

Proof As *ssercond(h)* holds, it is immediate from Definition 5 that $T_{i_ke_k}$ satisfying (i) can be chosen. Now, let $h' = T_{i_ke_k}(h - T_{i_ke_k})$. To prove (ii) we show that $sees^x_h(R_{ij}, W_{i'j'})$ iff $sees^x_{h'}(R_{ij}, W_{i'j'})$ for any read and write steps R_{ij} and $W_{i'j'}$ respectively. Consider the non-trivial case that $x \in S(W_{i_ke_k})$. As $T_{i_1e_1}, \dots, T_{i_ne_n}$ are the earliest occurrences in h and $T_{i_ke_k}$ is s-detachable then, by Definition 5(ii) and (iii), h is of the form

$$\underbrace{\dots R_{i_ke_k}[x.] \dots}_{\text{no writes to } x} W_{i_ke_k}[x.] \dots$$

If $(i, j) = (i_k, e_k)$, then $\neg sees^x_h(R_{ij}, W_{i'j'})$ and $\neg sees^x_{h'}(R_{ij}, W_{i'j'})$ as R_{ij} is then the first step in h'. If $(i', j') = (i_k, e_k)$, h is of the form

$$\underbrace{\dots R_{i_ke_k}[x.] \dots}_{\text{no writes to } x} W_{i_ke_k}[x.] \dots R_{ij}[x.]$$

and, as h' only moves $R_{i_ke_k}[x.]$ and $W_{i_ke_k}$ to the left, $sees^x_{h'}(R_{ij}, W_{i_ke_k})$ will be the same as $sees^x_h(R_{ij}, W_{i_ke_k})$. If $(i, j) \neq (i_k, e_k)$ and $(i', j') \neq (i_k, e_k)$, then h cannot be of the form

$$\underbrace{\dots R_{ij}[x.] \dots}_{\text{no writes to } x} W_{i_ke_k}[x.] \dots$$

as $T_{i_ke_k}$ is detachable and Definition 5(i) would be breached as the read step of the earliest occurrence of T_i would precede $R_{ij}[x.]$ and therefore $W_{i_ke_k}[x.]$. So, h is of the form

$$\underbrace{\dots\dots\dots\dots}_{\text{no writes to } x} W_{i_ke_k}[x.] \dots R_{ij}[x.] \dots$$

in which case $sees^x_h(R_{ij}, W_{i'j'})$ iff $sees^x_{h'}(R_{ij}, W_{i'j'})$ as h only moves $R_{i_ke_k}[x.]$ and $W_{i_ke_k}[x.]$ to the left.

For (iii), let $T_{i_1j_1}, \dots, T_{i_nj_n}$ be a sequence of (not necessarily the earliest) occurrences of $T_1, \dots, T_n$ in $h'' = h - T_{i_ke_k}$. As $T_{i_ke_k} \notin \{T_{i_1j_1}, \dots, T_{i_nj_n}\}$ then, by the definition of h'', for $1 \leq f < g \leq n$,

$$R_{i_fj_f} \leq_h R_{i_gj_g} \text{ iff } R_{i_fj_f} \leq_{h''} R_{i_gj_g}$$

and

$$W_{i_fj_f} \leq_h W_{i_gj_g} \text{ iff } R_{i_fj_f} \leq_{h''} W_{i_gj_g}$$

From this, it is clear that a s-detachable $T_{i_kj_k}$ of $T_{i_1j_1}, \dots, T_{i_nj_n}$ in h is also s-detachable in h''. It follows that *ssercond*(h'') holds. ■

Theorem 8 A history h is strictly serializable iff *ssercond(h)* holds.

Proof Let h be strictly serializable. Choose a strictly serial history h_S such that $h \sim h_S$. Let $T_{i_1j_1}, \dots, T_{i_nj_n}$ be occurrences in h. As h is strictly serializable then, by Definition 3, one of the occurrences $T_{i_kj_k}$ is such that, for $1 \leq g \leq n$, $g \neq k$, $R_{i_kj_k} <_h W_{i_gj_g}$ and h_S is of the form

$$\dots R_{i_kj_k} W_{i_kj_k} \dots R_{i_gj_g} W_{i_gj_g} \dots$$

Thus, $T_{i_kj_k}$ satisfies Definition 5(iv). By (1) and (2), for $x \in \{x_1, \dots, x_m\}$

$$W_{i_kj_k}[x.] <_h R_{i_gj_g}[x.],$$

$$R_{i_kj_k}[x.] <_h W_{i_gj_g}[x.],$$

$$W_{i_kj_k}[x.] <_h W_{i_gj_g}[x.]$$

and so the conditions Definition 5(i), (ii) and (iii) are also satisfied. Therefore, $T_{i_kj_k}$ is s-detachable. It follows that *ssercond(h)* holds.

Conversely, suppose that *ssercond(h)* holds. We show that h is strictly serializable. Define a sequence $h_0, \dots, h_m, \dots$ of histories, inductively, as follows

$$h_0 = h, \;\; h_{m+1} = h_m - T_{i_{m_k}j_{m_k}} \;\; (m \geq 0) \tag{3}$$

where $T_{i_{m_k}j_{m_k}}$ is defined to be a s-detachable member of the earliest occurrences of h_m. Now define the sequence h_S whose $2m$-th and $(2m+1)$-th $(m \geq 0)$ steps are

$$h_S(2m) = R_{i_{m_k}j_{m_k}}, \;\; h_S(2m+1) = W_{i_{m_k}j_{m_k}}$$

We show that h_S is strictly serial by showing that conditions (i), (ii) and (iii) of Definition 3 are satisfied. Condition (i) is satisfied as h_S is serial by construction. For condition (ii), we need to show that h_S has the same occurrences as h. Assume, on the contrary, that there is an occurrence, $T_{i_1j_1}$ say, in h that is not in h_S. Without loss of generality, we can choose j_1 to be the smallest value for which $T_{i_1j_1}$ is in h but not in h_S, i.e.

$$T_{i_1j_1'} \in h \text{ and } T_{i_1j_1'} \notin h_S \text{ implies } j_1 \leq j_1'$$

Now, as h_S is infinite, there is some transaction, T_{i_2} say, which has infinitely many occurrences in h_S. Therefore, we can choose an occurrence $T_{i_2j_2}$ in h_S such that

$$W_{i_1j_1} <_h R_{i_2j_2} \tag{4}$$

By (3), $T_{i_2j_2}$ belongs to h_S because there is an integer $l \geq 0$ such that

$$h_{l+1} = h_l - T_{i_2j_2}$$

and $T_{i_2j_2}$ is a s-detachable member of the earliest occurrences of $T_1, \dots, T_n$ in h_l. Consider the earliest occurrence of T_{i_1} in h_l. As $T_{i_1j_1}$ is not in h_S, by the inductive definition of h_S

(3), $T_{i_1j_1}$ must be in h_l and, as h_l is a subsequence of h, the earliest occurrence $T_{i_1j_1''}$ in h_l is such that

$$W_{i_1j_1''} \leq_h R_{i_1j_1} <_h W_{i_1j_1} \tag{5}$$

By (4) and (5), we have that

$$W_{i_1j_1''} <_h R_{i_2j_2} \tag{6}$$

But, $T_{i_2j_2}$ is a s-detachable member of the earliest occurrences of $T_1, \dots, T_n$ in h_l which includes the occurrence $T_{i_1j_1''}$. Therefore, by Definition 5(iv),

$$\neg W_{i_1j_1''} <_h R_{i_2j_2} \tag{7}$$

The contradiction between (6) and (7) means that all occurrences in h will appear in h_S.

To show that condition (iii) of Definition 3 holds, suppose that

$$W_{i_fj_f} <_h R_{i_gj_g}$$

and assume, on the contrary, that

$$R_{i_gj_g} <_{h_S} W_{i_fj_f} \tag{8}$$

Then, there is a $l \geq 0$ such that

$$h_{l+1} = h_l - T_{i_gj_g} \text{ and } T_{i_fj_f} \in h_{l+1}$$

If $T_{i_fj_f'}$ is the earliest occurrence of T_{i_f} in h_l then we have that:

$$W_{i_fj_f'} <_h W_{i_fj_f} <_h R_{i_gj_g} \tag{9}$$

But, $T_{i_gj_g}$ is a s-detachable member of the earliest occurrences of h_l and therefore, by Definition 5(iv),

$$R_{i_gj_g} <_h W_{i_fj_f'}$$

contrary to (9). Thus (8) cannot hold.

We have now shown that h_S is strictly serial. It remains to show that $h \sim h_S$. By (3) and Lemma 7(ii), for $m \geq 0$,

$$R_{i_{m_k}j_{m_k}} W_{i_{m_k}j_{m_k}} h_{m+1} = T_{i_{m_k}j_{m_k}}(h_m - T_{i_{m_k}j_{m_k}}) \;\sim\; h_m$$

and so

$$h = h_0 \;\sim\; R_{i_{0_k}j_{0_k}} W_{i_{0_k}j_{0_k}} \dots R_{i_{m_k}j_{m_k}} W_{i_{m_k}j_{m_k}} h_{m+1} \tag{10}$$

If R_{ij} and $W_{i'j'}$ are read and write occurrences in h, choose m sufficiently large so that in (10)

$$R_{i_{0_k}j_{0_k}} W_{i_{0_k}j_{0_k}} \dots R_{i_{m_k}j_{m_k}} W_{i_{m_k}j_{m_k}}$$

includes both R_{ij} and $W_{i'j'}$. From this, it is clear that $sees_h(R_{ij}, W_{i'j'})$ iff $sees_{h_s}(R_{ij}, W_{i'j'})$ and it follows that $h \sim h_S$. ■

III. Representatives of refutations

In view of Theorem 8, a history h is strictly serializable iff whenever occurrences of all of $T_1, \dots, T_n$ in h are selected, the resulting subsequence of h of $2n$ steps is s-detachable. Therefore, in order to prove that h is not strictly serializable, a sequence of matching read and write steps of all of $T_1, \dots, T_n$ in h, that is not s-detachable, has to be found. The number of different sequences (permutations) of the 2n steps of $T_1, \dots, T_n$ such that a read step comes before a write step is $(2n)!/2^n$ which is greater than 2^n for $n > 1$. Now, strict serializability can be encoded into temporal logic by locating all such possible sequences of steps occurring in h and asserting their s-detachability. However, if all possible sequences of $2n$ steps are encoded, the temporal logic formula is exponential in the number of transactions n. This presents a major obstacle to proving strict serializability in the cases of large numbers of transactions. Fortunately, this problem can be overcome as the number of data items places a bound on the number of steps of sequences that have to be considered. In this section, we define a 'representative' to be a sequence of steps of transactions that occur in a history h and refute the strict serializability of h.

Definition 9 Let h be a history, p be an integer such that $1 \leq p \leq n$, $\{i_1, \dots, i_p\} \subseteq \{1, \dots, n\}$, and ρ be a bijection

$$\rho \;:\; \{1, \dots, 2p\} \;\rightarrow\; \{R_{i_1}, W_{i_1}, \dots, R_{i_p}, W_{i_p}\}$$

Then, a subsequence Σ of h comprising the steps $R_{i_1}, W_{i_1}, \dots, R_{i_p}, W_{i_p}$ occurring in the order

$$\Sigma \;=\; \rho(1) \dots \rho(2p)$$

is a *representative of (a refutation of strict serializability for)* h iff there is a sequence of transaction occurrences $T_{i_1j_1}, \dots, T_{i_pj_p}$, whose steps in h occur in the order of the steps in Σ, that is not s-detachable. The following theorem places a bound on the number of steps of representatives that need to be considered to refute strict serializability, independent of n if n is sufficiently large.

Theorem 10 If $n \geq 2^{m+2}$, then a history h has a representative with $2n$ steps iff h has a representative with 2^{m+2} steps.

Proof

If

Suppose that h has a representative Σ of 2^{m+2} steps. Then, by Definition 9, there is a corresponding sequence of transaction occurrences $T_{i_1j_1}, \dots, T_{i_pj_p}$, where $p = 2^{m+1}$, that is not s-detachable and whose steps occur in h in the same order as in Σ. Choose occurrences $T_{i_{p+1}j_{p+1}}, \dots, T_{i_nj_n}$ such that $\{T_{i_1}, \dots, T_{i_n}\} = T_1, \dots, T_n$ and that

$$W_{i_gj_g} <_h R_{i_fj_f} \;\; (1 \leq g \leq p,\; p+1 \leq f \leq n) \tag{11}$$

We show that $T_{i_1}, \dots, T_{i_n}$ is not s-detachable. Now, no $T_{i_kj_k}$ such that $1 \leq k \leq p$ is s-detachable in $T_{i_1j_1}, \dots, T_{i_nj_n}$ as $T_{i_1j_1}, \dots, T_{i_pj_p}$ is not s-detachable, and so $T_{i_kj_k}$ will not satisfy (i)-(iv) of Definition 5 for $1 \leq g \leq p$ let alone for $1 \leq g \leq n$. But, also, no $T_{i_k j_k}$ with $p+1 \leq k \leq n$ is s-detachable as, by (11), for $1 \leq g \leq p$,

$$W_{i_g j_g} <_h R_{i_k j_k}$$

and (iv) of Definition 5 cannot be satisfied by such a k. Thus, $T_{i_1}, \ldots, T_{i_n}$ is not s-detachable and the representative whose steps occur in the order that the steps of the occurrences $T_{i_1 j_1}, \ldots, T_{i_p j_p}, T_{i_{p+1} j_{p+1}}, \ldots T_{i_n j_n}$ occur in h is the required representative of $2n$ steps.

Only if

Suppose that h has a representative of $2n$ steps, corresponding to the occurrences $T_{i_1 j_1}, \ldots, T_{i_n j_n}$. Put

$$\mathcal{S}_R = \{S(R_{i_f}) \mid 1 \leq f \leq n\},\ \mathcal{S}_W = \{S(W_{i_g}) \mid 1 \leq g \leq n\}$$

Clearly, $\mathcal{S}_R$ and $\mathcal{S}_W$ each have at most 2^m elements as there are only 2^m subsets of the set of data items $\{x_1, \ldots, x_m\}$. Choose

$$T_{i_{f_1} j_{f_1}}, \ldots, T_{i_{f_{2^m}} j_{f_{2^m}}}$$

to be such that

$$\{S(R_{i_{f_1}}), \ldots, S(R_{i_{f_{2^m}}})\} = \mathcal{S}_R \tag{12}$$

and that, for all $1 \leq f \leq n$, there is a l, with $1 \leq l \leq 2^m$, such that

$$S(R_{i_f}) = S(R_{i_{f_l}}) \text{ and } R_{i_{f_l} j_{f_l}} \leq_h R_{i_f j_f} \tag{13}$$

Basically, (12) states that the read sets of the chosen 2^m transaction occurrences $T_{i_{f_1} j_{f_1}}, \ldots, T_{i_{f_{2^m}} j_{f_{2^m}}}$ span all the read sets of the possibly greater number of transaction occurrences $T_{i_1 j_1}, \ldots, T_{i_n j_n}$. The condition (13) states that the earliest occurrences spanning those read sets, should be chosen. In a similar way, we can choose

$$T_{i_{g_1} j_{g_1}}, \ldots, T_{i_{g_{2^m}} j_{g_{2^m}}}$$

to be such that

$$\{S(W_{i_{g_1}}), \ldots, S(W_{i_{g_{2^m}}})\} = \mathcal{S}_W \tag{14}$$

and that, for all $1 \leq g \leq n$, there is a l, with $1 \leq l \leq 2^m$, such that

$$S(W_{i_g}) = S(W_{i_{g_l}}) \text{ and } W_{i_{g_l} j_{g_l}} \leq_h W_{i_g j_g} \tag{15}$$

We show that the sequence of the $2(2^m + 2^m) = 2^{m+2}$ steps in h of the occurrences

$$T_{i_{f_1} j_{f_1}}, \ldots, T_{i_{f_{2^m}} j_{f_{2^m}}}, T_{i_{g_1} j_{g_1}}, \ldots, T_{i_{g_{2^m}} j_{g_{2^m}}} \tag{16}$$

is a representative. This means showing that the sequence (16) is not s-detachable. Now, the sequence $T_{i_1 j_1}, \ldots, T_{i_n j_n}$ is certainly not s-detachable as its steps form a representative. Assume, on the contrary, that the sequence (16) is s-detachable. Then, one of its occurrences, $T_{i_k j_k}$ say, satisfies (i)-(iv) of Definition 5. We derive the contradiction that $T_{i_k j_k}$ is a s-detachable occurrence of $T_{i_1 j_1}, \ldots, T_{i_n j_n}$. Let $1 \leq g \leq n$, $g \neq k$ and $x \in \{x_1, \ldots, x_m\}$. We have, by (13), that, for some l with $1 \leq l \leq 2^m$,

$$S(R_{i_g}) = S(R_{i_{f_l}}) \text{ and } R_{i_{f_l} j_{f_l}} \leq_h R_{i_g j_g} \tag{17}$$

As $T_{i_k j_k}$ is a s-detachable occurrence in (16), then, by Definition 5(i),

$$W_{i_k j_k}[x.] \leq_h R_{i_{f_l} j_{f_l}}[x.] \tag{18}$$

By (17) and (18),

$$W_{i_k j_k}[x.] \leq_h R_{i_g j_g}[x.]$$

Thus, Definition 5(i) is satisfied by $T_{i_k j_k}$ for occurrences $T_{i_1 j_1}, \ldots, T_{i_n j_n}$. Next, by (15), we have that, for some l with $1 \leq l \leq 2^m$,

$$S(W_{i_g}) = S(W_{i_{g_l}}) \text{ and } W_{i_{g_l} j_{g_l}} \leq_h W_{i_g j_g} \tag{19}$$

As $T_{i_k j_k}$ is a s-detachable occurrence in (16), then, by Definition 5(ii),

$$R_{i_k j_k}[x.] \leq_h W_{i_{g_l} j_{g_l}}[x.] \tag{20}$$

By (19) and (20),

$$R_{i_k j_k}[x.] \leq_h W_{i_g j_g}[x.]$$

Thus, Definition 5(ii) is satisfied by $T_{i_k j_k}$ for occurrences $T_{i_1 j_1}, \ldots, T_{i_n j_n}$. Next, as $T_{i_k j_k}$ is a s-detachable occurrence in (16), then, by Definition 5(iii),

$$W_{i_k j_k}[x.] \leq_h W_{i_{g_l} j_{g_l}}[x.] \tag{21}$$

By (19) and (21),

$$W_{i_k j_k}[x.] \leq_h W_{i_g j_g}[x.]$$

Thus, Definition 5(iii) is satisfied by $T_{i_k j_k}$ for occurrences $T_{i_1 j_1}, \ldots, T_{i_n j_n}$. Finally, as $T_{i_k j_k}$ is a s-detachable occurrence in (16), by Definition 5(iv),

$$R_{i_k j_k} \leq_h W_{i_{g_l} j_{g_l}} \tag{22}$$

By (19) and (22),

$$R_{i_k j_k} \leq_h W_{i_g j_g}$$

Thus, Definition 5(iv) is satisfied by $T_{i_k j_k}$ for occurrences $T_{i_1 j_1}, \ldots, T_{i_n j_n}$. We have now derived the contradiction that $T_{i_k j_k}$ is a s-detachable occurrence of $T_{i_1 j_1}, \ldots, T_{i_n j_n}$. Therefore, the assumption that (16) is detachable is untenable and it follows that the sequence of 2^{m+2} steps of the occurrences (16) is a representative as required. ■

IV. A temporal logic

We define propositional linear temporal logic with a strict serializability operator, and denote the logic by *PTL+sser*. The alphabet of *PTL+sser* consists of a list of propositional symbols $P_0, P_1, \ldots$, a list of special read/write step propositional symbols $R_1, R_2, \ldots$ and $W_1, W_2, \ldots$, booleans $\neg, \wedge, \top, \bot$, and temporal operators $\bigcirc$ and $\mathcal{U}$. Formulae in *PTL+sser* are either 'top-level' formulae τ or bottom-level formulae ψ generated by:

$$\tau \ ::= \ \neg\tau \mid \tau_1 \wedge \tau_2 \mid sser(\psi)$$

$$\psi \ ::= \ P_i \mid R_i \mid W_i \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \top \mid \bot \mid \bigcirc\psi \mid \psi_1 \mathcal{U} \psi_2$$

We use the standard abbreviations for $\vee$, $\Rightarrow$ and $\Leftrightarrow$, and

$$\Diamond\psi = \top\mathcal{U}\psi, \ \ \Box\psi = \neg\Diamond\neg\psi$$

A. Semantics

The semantics for *PTL+sser* is given with respect to a given set of data items $X = \{x_1, \ldots, x_m\}$ being accessed by transactions and, for each positive integer i, a given set of data items read by transaction i, $S(R_i)$, and a given set of data items written to by transaction i, $S(W_i)$, such that $X \supseteq S(R_i) \supseteq S(W_i)$.

A *model* for *PTL+sser* is an assignment M of the propositions that are true at each point in time $a \in \mathbb{N}$, i.e.

$$M : \mathbb{N} \to \wp(\{P_0, P_1, \ldots, R_1, R_2, \ldots, W_1, W_2, \ldots\})$$

where $M(a)$ gives the set of propositions equal to $\top$ (true) at time $a \in \mathbb{N}$ ($\wp$ is the powerset constructor) such that:

(i) for each $a \in \mathbb{N}$,

$$M(a) \cap \{R_1, R_2, \ldots, W_1, W_2, \ldots\} = \{Q_a\}$$

is a singleton

(ii) the sequence of (read/write) step propositions

$$Q_0, Q_1, \ldots \tag{23}$$

is a history for $T_1, \ldots, T_n$ where T_i comprises the read and write steps R_i and W_i $(1 \leq i \leq n)$

A model M is *strictly serializable* iff the history corresponding to the sequence of propositions (23) is strictly serializable. The semantics of bottom-level formulae is given as for standard propositional linear temporal logic, by the truth relations $(M, a) \vDash \psi$ $(a \in \mathbb{N})$ defined inductively on the construction of ψ as follows:

$(M, a) \vDash P_i$ iff $P_i \in M(a)$
$(M, a) \vDash R_i$ iff $R_i \in M(a)$
$(M, a) \vDash W_i$ iff $W_i \in M(a)$
$(M, a) \vDash \neg\psi$ iff $(M, a) \nvDash \psi$
$(M, a) \vDash \psi_1 \wedge \psi_2$ iff $(M, a) \vDash \psi_1$ and $(M, a) \vDash \psi_2$
$(M, a) \vDash \bigcirc\psi$ iff $(M, a+1) \vDash \psi$
$(M, a) \vDash \psi_1 \mathcal{U} \psi_2$ iff, for some $b \geq a$, $(M, b) \vDash \psi_2$ and, for $a \leq c < b$, $(M, c) \vDash \psi_1$

A formula ψ is said to be *satisfied* by the model M at a iff $(M, a) \vDash \psi$. The semantics of top-level formulae is given by the truth relation $(M, 0) \vDash \tau$ defined as follows:

$(M, a) \vDash \neg\tau$ iff $(M, a) \nvDash \tau$
$(M, a) \vDash \tau_1 \wedge \tau_2$ iff $(M, a) \vDash \tau_1$ and $(M, a) \vDash \tau_2$
$(M, a) \vDash \mathit{sser}(\psi)$ iff $(M, 0) \vDash \psi$ implies that M is strictly serializable

A *PTL+sser* formula ϕ is *valid* written

$$\vDash \phi$$

iff $(M, 0) \vDash \phi$ for all models M. It is clear that $\vDash \mathit{sser}(\psi)$ asserts that all models satisfying ψ (at 0) are strictly serializable.

B. An encoding of the sser operator

We encode the *sser* operator into plain propositional linear temporal logic (*PTL*) without the *sser* operator, by encoding the representatives of Theorem 10. We consider the interesting case when $n \geq 2^{m+2}$. Suppose that ψ has the read/write step propositions $R_1, W_1, \ldots, R_n, W_n$. Let $\underline{\rho}$ be the set of bijections:

$$\rho : \{1, \ldots, 2^{m+2}\} \to \{R_{i_1}, W_{i_1}, \ldots, R_{i_{2m+1}}, W_{i_{2m+1}}\}$$

where $\{i_1, \ldots, i_{2^{m+1}}\} \subseteq \{1, \ldots, n\}$,

$$\rho^{-1}(R_{i_g}) < \rho^{-1}(W_{i_g}) \;\; (1 \leq g \leq 2^{m+1})$$

and such that there is no k with $1 \leq k \leq 2^{m+1}$ satisfying, for $1 \leq g \leq 2^{m+1}$, $g \neq k$ and $x \in \{x_1, \ldots, x_m\}$, the following:

(i') $\neg(\rho^{-1}(R_{i_g}[x.]) < \rho^{-1}(W_{i_k}[x.]))$
(ii') $\neg(\rho^{-1}(W_{i_g}[x.]) < \rho^{-1}(R_{i_k}[x.]))$
(iii') $\neg(\rho^{-1}(W_{i_g}[x.]) < \rho^{-1}(W_{i_k}[x.]))$
(iv') $\neg(\rho^{-1}(W_{i_g}) < \rho^{-1}(R_{i_k}))$

As (i')-(iv') correspond to (i)-(iv) of Definition 5, it is clear that a model M is not strictly serializable iff, for some $\rho \in \underline{\rho}$, the read and write propositions of M are of the form

$$\ldots, \rho(1), \ldots, \rho(2), \ldots, \rho(2^{m+2}), \ldots \tag{24}$$

and are not of the form, for any $1 \leq u < v \leq 2^{m+1}$ and $1 \leq i \leq n$,

$$\ldots, \rho(u) = R_i, \ldots, R_i, \ldots, \rho(v) = W_i, \ldots \tag{25}$$

Condition (24) is essentially the condition that $\rho(1), \ldots, \rho(2^{m+2})$ is a representative, although we need the extra condition (25) to guarantee that if $\rho(u) = R_i$ then the later $\rho(v) = W_i$ is the write step for the same occurrence of T_i. We can now encode $\mathit{sser}(\psi)$ as follows:

$$\mathit{sser}(\psi) \;\; = \;\; \psi \to \neg \bigvee_{\rho \in \underline{\rho}} (\tag{26}$$

$$\Diamond(\rho(1)\mathcal{U}(\rho(2)\mathcal{U}(\ldots\rho(2^{m+2})\ldots))) \wedge \tag{27}$$

$$\bigwedge_{1 \leq i \leq n} \bigwedge_{1 \leq u \leq 2^{m+2}} \bigvee_{1 \leq v \leq 2^{m+2}} (\rho(u) \wedge R_i \to \neg W_i \mathcal{U}(\rho(v) \wedge W_i))) \tag{28}$$

Here, (27) and (28) encode the conditions (24) and (25) respectively.

Theorem 11 The validity problem for *PTL+sser* is PSPACE-complete.

Proof As *PTL+sser* contains *PTL*, and *PTL* is PSPACE-hard [14], it follows that *PTL+sser* is PSPACE-hard. On the other hand, *PTL+sser* can be encoded into PTL by encoding the *sser* operator as above. This involves computing fewer than $(2n)!/(2n - 2^{m+2})!(2^{m+2})!$ $\rho \in \underline{\rho}$ satisfying (i')-(iv'), i.e. a polynomial in n number of ρ. The encoding of $\mathit{sser}(\psi)$ into *PTL*, as given by (26), (27) and (28) is therefore achieved with at most a polynomial increase in the size of ψ. Thus, any formula ϕ in *PTL+sser* containing subformulae of the form $\mathit{sser}(\psi)$ can be rewritten by a *PTL* formula without any occurrences of the *sser* operator, incurring at worst a polynomial increase in size of formula. Therefore, *PTL+sser* is in PSPACE. It follows that *PTL+sser* is PSPACE-complete. ■

V. Conclusions

The importance of modelling infinite schedules of concurrent transactions is growing with the appearance new technologies such as mobile transactions. A natural way of modelling such schedules is to use temporal logic. The few existing approaches that have considered this problem, use temporal logics that rely on the manual use of proof rules to produce correctness proofs of the main consistency property of serializability. In this paper, we have presented a version of serializability that can be easily realized as an additional operator to one of the most common temporal logics of all - propositional linear temporal logic. We have shown that the validity problem of the resulting extended logic is of the same PSPACE-complete computational complexity. Further to this, regarding *PTL+sser* model-checking, we note from [2] that the algorithm that checks whether a finite state machine satisfies a *PTL* formula has time complexity $O((|S|+|R|).2^{O(|f|)}$ where $|S|$ is the number of states, $|R|$ is the number of transitions and $|f|$ is the length of the formula. Given a *PTL+sser* formula g, the encoding in section IV which removes instances of the *sser* operator produces a *PTL* formula f whose length is a polynomial in the length of g. It follows that the model-checking algorithm for *PTL+sser* is of comparable time complexity to that for *PTL*. Therefore, in every respect, proofs in *PTL+sser* should be as efficient as proofs in plain *PTL*. So, the logic *PTL+sser* is suitable for use with the well-known powerful model-checkers [1] and [6]. This opens up the possibility of conducting proofs of correctness of infinite schedules using fully automated means avoiding the drawbacks of manual proofs.

Further work will look to extend these results to the case of infinite schedules of concurrent multi-step transactions.

References

[1] A. Cimatti, E. Clarke, F. Giunchiglia, and M. Roveri, "NuSMV: a new symbolic model verifier," *Lecture Notes in Computer Science*, vol. 1633, pp. 495-499, 1999.

[2] E. Clarke, O. Grumberg and D. Peled, *Model Checking*, MIT Press, 1999.

[3] C.J. Date, *An Introduction to Database Systems*, Addison Wesley, 2004.

[4] M.P. Fle and G. Roucairol, "On serializability of iterated transactions," *Proc, 1st ACM SIGACT-SIGOPS Symp. on Principles of Distributed Computing*, pp. 194-200, 1982.

[5] M.P. Fle and G. Roucairol, "Multiserialization of iterated transactions," *Information Processing Letters*, vol. 18, pp. 243-247, 1984.

[6] G.J. Holzmann, *The SPIN model checker*. Addison-Wesley, 2004.

[7] W. Hussak, "Serializable Histories in Quantified Propositional Temporal Logic," *International Journal of Computer Mathematics*, vol. 81(10), pp. 1203-1211, 2004.

[8] W. Hussak and J.A. Keane, "Algebraic Specification of Serializability for Partitioned Transactions," *International Journal of Computer Systems Science and Engineering*, vol. 1(1), pp. 60-67, 2007.

[9] S. Katz and D. Peled, "Defining conditional independence using collapses," *Theoretical Computer Science*, vol. 101, pp. 337-359, 1992.

[10] C. Papadimitriou, "The Serializability of Concurrent Database Updates," *Journal of ACM*, vol. 26(4), pp. 631-653, 1979.

[11] D. Peled and A. Pnueli, "Proving partial order properties," *Theoretical Computer Science*, vol. 126, pp. 143-182, 1994.

[12] A. Pnueli, "Temporal logic of programs," *Proc. 18th IEEE Symp. on Foundations of Computer Science*, pp. 46-57, IEEE Computer Society Press, 1977.

[13] L. Sha, J.P. Lehoczky and E.D. Jensen, "Modular Concurrency Control and Failure Recovery," *IEEE Transactions on Computers*, vol. 37(2), pp. 146-159, 1988.

[14] A.P. Sistla and E.M. Clarke, "The Complexity of Propositional Linear Temporal Logics," *Journal of the ACM*, vol. 32, pp. 733-49, 1985.

[15] K. Vidyasankar, "Generalized Theory of Serializability," *Acta Informatica*, vol. 24, pp. 105-119, 1987.