A Notion of Serializability for Document Editing and Corresponding Optimal Locking Protocols

Stijn Dekeyser
University of Southern Queensland

Jan Hidders
Delft University of Technology

Contents

1 Introduction 1
2 General graph locking 2
3 Operation accumulation 5
4 Cyclic instances 6
5 Accumulation of cyclic operations 10
6 Locking cyclic operations 15

Abstract

This report describes the theoretical results underlying the PIECEMEAL concurrency management technique for collaborative editing. It presents a formalization of a document and edit operations on a document based on a graph model. Starting from these we propose notions of transactions and serializability that are related to how conflicts are defined in version-management systems such as CVS and Subversion. Then it is investigated how optimal certain locking strategies are that are based on edge-locking and that realize the aforementioned serializability. It is also investigated if and how in this setting series of operations can be combined into a single operation.

1 Introduction

The basic assumption in this work is that documents are modeled as linear graphs where the nodes represent parts of the document such as paragraphs and the edges represent the immediate precedence relationship, i.e., their order in the document. Operations on a document are modeled as triples of sets of edges, \((P, D, A)\) where \(P\) represents the context of the operation, i.e., the part of the document that is assumed to exists if the operation is to be correct, the set \(D\) represents the edges that have to be deleted and the set \(A\) contains the edges that have to be added. We will require for proper document operations that \(D \subseteq P\) and \(A \cap P = \emptyset\).

The main locking protocol that we will investigate is the following. For edges in \(P\) read-locks must be requested and for edges in \(D\) and \(A\) write-locks must be requested. As usual we will assume that write-locks cannot be obtained if there is already a write-lock or a read-lock on an item.
As is shown in the following sections, this protocol guarantees serializability but it is not optimal in the sense that it does not allow the maximum amount of concurrency, if we consider arbitrary graphs. An alternative protocol is presented, or rather a corresponding notion of conflict between operations, that is in fact optimal for that case. However, it is also shown that if we add the typical restrictions that would hold for graphs that represent documents and operations that manipulate documents, then it is in fact optimal.

2 General graph locking

Definition 2.1 (Instance). An instance is a set \( I \subseteq \mathcal{N} \times \mathcal{N} \). The set of all instances is \( \mathcal{I} \).

Definition 2.2 (Operation). An operation is a tuple \( o = (P, D, A) \) where \( P, D, A \subseteq \mathcal{N} \times \mathcal{N} \) are respectively the pattern edges, deleted edges and added edges, such that \( D \cap A = \emptyset \).

The semantics of of an operation \( o = (P, D, A) \) is defined as a partial function \( o : \mathcal{I} \to \mathcal{I} \) such that \( o(I) = (I - D) \cup A \) if \( P \subseteq I \) and undefined otherwise. The concatenation of two such partial functions \( o_1 \) and \( o_2 \) is denoted as \( o_1 \circ o_2 \) and defined such that \( (o_1 \circ o_2)(I) = o_1(o_2(I)) \) if \( o_1(o_2(I)) \) is defined and undefined otherwise.

Definition 2.3 (Disabling operations). Operation \( o_1 = (P_1, D_1, A_1) \) is said to disable operation \( o_2 = (P_2, D_2, A_2) \) if \( P_2 \cap D_1 \neq \emptyset \).

Theorem 2.1. Operation \( o_1 \) disables \( o_2 \) iff \( o_2 \circ o_1 \) is undefined for all instances.

Proof. Clearly if \( P_2 \cap D_1 \neq \emptyset \) then \( o_2 \circ o_1 \) is undefined for all instances. Assume that \( P_2 \cap D_1 = \emptyset \). Then let \( I = P_1 \cup (P_2 - A_1) \). Clearly \( o_1(I) \) is defined since \( P_1 \subseteq P_1 \cup (P_2 - A_1) \), and so \( o_1(I) = (I - D_1) \cup A_1 \). We now show that \( o_2(o_1(I)) \) is defined by showing that \( P_2 \subseteq o_1(I) \) because

\[
\begin{align*}
o_1(I) &= (I - D_1) \cup A_1 \\
&= ((P_1 \cup (P_2 - A_1)) - D_1) \cup A_1 \\
&= (P_1 - D_1) \cup (P_2 - A_1) \cup A_1 \quad \text{since} \ P_2 \cap D_1 = \emptyset \\
&= (P_1 - D_1) \cup P_2 \cup A_1
\end{align*}
\]

Corollary. Two operations \( o_1 \) and \( o_2 \) are mutually disabling iff both \( o_1 \circ o_2 \) and \( o_2 \circ o_1 \) are undefined for all instances.

Definition 2.4 (Conflicting operations). Two operations \( o_1 = (P_1, D_1, A_1) \) and \( o_2 = (P_2, D_2, A_2) \) are said to conflict if at least one of the following holds:

1. \( P_1 \cap D_2 \neq \emptyset \)
2. \( P_2 \cap D_1 \neq \emptyset \)
3. \( P_1 \cap A_2 \neq \emptyset \)
4. \( P_2 \cap A_1 \neq \emptyset \)
5. \( D_1 \cap A_2 \neq \emptyset \)
6. \( D_2 \cap A_1 \neq \emptyset \)
Theorem 2.2 (Commutativity). For all operations $o_1$ and $o_2$ that are not mutually disabling it holds that $o_1$ and $o_2$ do not conflict iff $o_1 \circ o_2 = o_2 \circ o_1$.

Proof. We first show that if $o_1$ and $o_2$ do not conflict then $o_1 \circ o_2 = o_2 \circ o_1$. This means that for every instance $I$ it holds that (i) $o_1(o_2(I))$ is defined iff $o_2(o_1(I))$ is defined and (ii) if both are defined then $o_1(o_2(I)) = o_2(o_1(I))$.

- Assume that $o_1(o_2(I))$ is defined. Then (a) $P_2 \subseteq I$ and (b) $P_1 \subseteq (I \setminus D_2) \cup A_2$. Since $c1$ not holds $P_1 \cap D_2 = \emptyset$ and since $c3$ not holds $P_1 \cap A_2 = \emptyset$, so it follows that $P_1 \subseteq I$, and so $o_1(I)$ is defined. Since $c2$ not holds $P_2 \cap D_1 = \emptyset$ and therefore it follows from (a) that $P_2 \subseteq (I \setminus D_1) \cup A_1$, and so $o_2(o_1(I))$ is defined. By symmetry and using that rules $c1$, $c2$ and $c4$ do not hold, it can be shown that if $o_2(o_1(I))$ is defined then $o_1(o_2(I))$ is defined.

- Assume that both $o_1(o_2(I))$ and $o_2(o_1(I))$ are defined. Since $c5$ and $c6$ do not apply it holds that $D_1 \cap A_2 = \emptyset$ and $D_2 \cap A_1 = \emptyset$. We can then show the following:

$$o_2(o_1(I)) = (((I - D_1) \cup A_1) - D_2) \cup A_2$$
$$o_1(o_2(I)) = (((I - D_1) - D_2) \cup A_1) \cup A_2$$

since $D_2 \cap A_1 = \emptyset$

We now show that if $o_1$ and $o_2$ conflict then $o_1 \circ o_2 \neq o_2 \circ o_1$. If $o_1$ disables $o_2$ but not vice versa, or if $o_2$ disables $o_1$ but not vice versa, then it follows by Theorem 2.1 that $o_1 \circ o_2 \neq o_2 \circ o_1$. Since we assume that $o_1$ and $o_2$ are not mutually disabling, the only remaining case is where $o_1$ does not disable $o_2$ and vice versa. We consider all possible ways there can be a conflict:

**c1** If $P_1 \cap D_2 \neq \emptyset$ then $o_2$ disables $o_1$, which leads to a contradiction.

**c2** Symmetrical to the case for **c1**.

**c3** If $P_1 \cap A_2 \neq \emptyset$ then let $I = (P_1 - A_2) \cup P_2$. Since $P_1 \cap A_2 \neq \emptyset$ and $A_2 \cap P_2 = \emptyset$ it holds that $P_1 \subseteq (P_1 - A_2) \cup P_2$ and so $o_1(I)$ is not defined and therefore also $o_2(o_1(I))$. However, it can be shown that $o_1(o_2(I))$ is defined as follows. Clearly $o_2(I)$ is defined since $P_2 \subseteq (P_1 - A_2) \cup P_2$. We now show that $o_1(o_2(I))$ is defined by showing that $P_1 \subseteq o_2(I)$ because

$$o_2(I) = ([I - D_2] \cup A_2)$$
$$= (((P_1 - A_2) \cup P_2) - D_2) \cup A_2$$
$$= ((P_1 \cup P_2) - A_2 - D_2) \cup A_2$$
$$= (P_1 \cup P_2 - D_2) \cup A_2$$
$$= P_1 \cup (P_2 - D_2) \cup A_2$$

since $P_1 \cap A_2 = \emptyset$ and $P_1 \cap D_2 = \emptyset$.

**c4** Symmetrical to the case for **c3**.

**c5** If $D_1 \cap A_2 \neq \emptyset$ then let $I = P_1 \cup P_2$. If $P_1 \cap A_2 \neq \emptyset$ or $P_2 \cap A_1 \neq \emptyset$ then we can proceed as in the preceding item, so we can assume here that $P_1 \cap A_2 = \emptyset$ and $P_2 \cap A_1 = \emptyset$. Clearly $P_1 \subseteq I$ so $o_1(I)$ is defined and $o_1(I) = ((P_1 \cup P_2) - D_1) \cup A_1$. Since $P_2 \cap D_1 = \emptyset$ it follows that $P_2 \subseteq o_1(I)$ and therefore $o_2(o_1(I))$ is defined. By symmetry it also follows that $o_1(o_2(I))$ is defined. However, $o_2(o_1(I)) \neq o_1(o_2(I))$ since $D_1 \cap A_2 \neq \emptyset$.  

The conflict rules also have to deal with the case where both \( o_1 \circ o_2 \) and \( o_2 \circ o_1 \) are never defined, in which case the operations commute even though they conflict.

**Proposition 2.3.** For all operations \( o_1 \) and \( o_2 \) it holds that \( o_1 \circ o_2 = o_2 \circ o_1 \) iff \( o_1 \) and \( o_2 \) do not conflict or are mutually disabling.

Since we will usually only consider schedules where all operations are defined for at least one instance the scheduler can ignore this condition. This is shown in the following.

**Definition 2.5** (Schedule). A schedule is a non-empty sequence \( S = \langle o_1, \ldots, o_n \rangle \) of operations.

**Definition 2.6** (Sound schedule). A schedule is said to be sound if there is an instance \( I \) such that \( (o_n \circ \ldots \circ o_1)(I) \) is defined.

**Definition 2.7** (Well-defined schedule). A schedule \( S = \langle o_1, \ldots, o_n \rangle \) is said to be well-defined if it holds that for all two operations \( o_i = (P_i, D_i, A_i) \) and \( o_j = (P_j, D_j, A_j) \) in \( S \) such that \( i < j \) and edges \((v_1, v_2) \in D_i \cap P_j\) there is an operation \( o_k = (P_k, D_k, A_k) \) such that \( i < k < j \) and \((v_1, v_2) \in A_k\).

**Theorem 2.4** (Well-definedness and soundness). A schedule is sound iff it is well-defined.

**Proof.** Clearly a schedule that is not well-defined will not have a defined result for any instance \( I \) and is therefore not sound.

For the other direction we start with assuming that \( S = \langle o_1, \ldots, o_n \rangle \) with \( o_i = (P_i, D_i, A_i) \) for all \( 1 \leq i \leq n \) is well-defined. Let \( I = \bigcup_{1 \leq i \leq n} P_i \). We show with induction that for each \( 1 \leq i \leq n \) it holds that the result of \( (o_i \circ \ldots \circ o_1)(I) \) is defined. Clearly for \( o_1(I) \) this is the case. Assume it holds for \( i \). For every edge \((v_1, v_2)\) in \( P_{i+1} \) it holds that it is either not deleted or deleted in \( o_i \circ \ldots \circ o_1 \). In the first case it is in \( (o_i \circ \ldots \circ o_1)(I) \). In the second case let \( o_j = (P_j, D_j, A_j) \) be the last operation in \( \langle o_1, \ldots, o_i \rangle \) that deletes \((v_1, v_2)\). By the condition on well-definedness there is an operation \( o_k \) with \( j < k < i + 1 \) that adds \((v_1, v_2)\), so also then this edge is in \( (o_i \circ \ldots \circ o_1)(I) \). It therefore holds that \( P_{i+1} \subseteq (o_i \circ \ldots \circ o_1)(I) \) and therefore \((o_{i+1} \circ o_i \circ \ldots \circ o_1)(I) \) is defined.

**Definition 2.8** (Conflict graph). The conflict graph of a schedule \( S = \langle o_1, \ldots, o_n \rangle \) is \( G_S = (V, E) \) where \( V = \{1, \ldots, n\} \) and \( E \) contains the edge \((i, j)\) iff \( i < j \) and \( o_i \) and \( o_j \) conflict.

**Definition 2.9** (Restricted conflict graph). The restricted conflict graph of a schedule \( S = \langle o_1, \ldots, o_n \rangle \) is \( G_{S}^r = (V, E) \) where \( V = \{1, \ldots, n\} \) and \( E \) contains the edge \((i, j)\) iff \( i < j \) and \( o_i \) and \( o_j \) conflict and are not mutually disabling.

**Note.** It is the restricted conflict graph that indicates whether operations commute or not.

**Proposition 2.5.** For every schedule \( S \) the conflict graph \( G_S \) is a super-graph of the restricted conflict graph \( G_{S}^r \).

**Theorem 2.6.** For every sound schedule \( S \) and edge \((i, j)\) in \( G_S \) there is a path from \( i \) to \( j \) in \( G_{S}^r \).
Proof. We proceed with a proof of the theorem with induction upon \( j - i \).

We first consider the base case where \( j - i = 1 \). Assume that \((i, j) \in G_S\). Since \( S \) is sound it holds that \( P_{i+1} \cap D_i = \emptyset \) and so \( o_1 \) and \( o_2 \) are not mutually disabling. It follows that \((i, j) \in G_S\).

We now consider the case where \( j - i > 1 \). Assume that \((i, j) \in G_S\). Since \( S \) is sound it follows by Theorem 2.4 that either (a) \( P_j \cap D_i = \emptyset \) or (b) there is an operation \( o_k = (P_k, D_k, A_k) \) in \( S \) such that \( i < k < j \) and \( A_k \cap D_i \neq \emptyset \) and \( A_k \cap P_j \neq \emptyset \). In case (a) it holds that \( o_i \) and \( o_j \) are not mutually disabling, and therefore \((i, j) \in G_S\). For case (b) we observe that \( o_i \) and \( o_k \) conflict by rule \( c5 \), and also \( o_k \) and \( o_j \) by rule \( c4 \). It then follows by induction that there is in \( G_S \) a path from \( i \) to \( k \) and from \( k \) to \( j \), and therefore also a path from \( i \) to \( j \).

\[ \Box \]

3 Operation accumulation

The used notion of operation is very general and allows the accumulation of sound schedules into a single operation. We therefore define the sum of two operations.

**Definition 3.1** (operation merging). Let \( o_1 = (P_1, D_1, A_1) \) and \( o_2 = (P_2, D_2, A_2) \) be two operations then \((o_1 + o_2) = (P_3, D_3, A_3)\) where \( P_3 = P_1 \cup (P_2 - A_1) \), \( D_3 = (D_1 - A_2) \cup D_2 \) and \( A_3 = (A_1 - D_2) \cup A_2 \).

Observe that \( o_1 + o_2 \) is indeed an operation since we know that \( D_1 \cap A_1 = \emptyset \) and \( D_2 \cap A_2 = \emptyset \) and hence \((D_1 - A_2) \cup D_2 \cap ((A_1 - D_2) \cup A_2) = \emptyset \). The merging can be shown to be correct if \( o_1 \) and \( o_2 \) form a sound schedule.

**Theorem 3.1** (Correctness of operation merging). If \( \langle o_1, o_2 \rangle \) is a sound schedule then \( o_2 \circ o_1 = o_1 + o_2 \).

Proof. We show that \( o_1 + o_2 \) is defined on the same instances as \( o_2 \circ o_1 \) and returns always the same result.

The partial function \( o_2 \circ o_1 \) is defined for \( I \) iff \( P_1 \subseteq I \) and \( P_2 \subseteq (I - D_1) \cup A_1 \). Since the schedule is sound we know that \( D_1 \cap P_2 = \emptyset \). Since \( P_2 - A_1 \subseteq I \) iff \( P_2 \subseteq I \cup A_1 \) it follows that \( P_2 - A_1 \subseteq I \) iff \( P_2 \subseteq (I - D_1) \cup A_1 \). It follows that \( P_1 \cup (P_2 - A_1) \subseteq I \) iff \( P_1 \subseteq I \) and \( P_2 \subseteq (I - D_1) \cup A_1 \).

We assume that both \((o_2 \circ o_1)(I)\) and \((o_1 + o_2)(I)\) are defined. Then it holds that:

\[
(o_2 \circ o_1)(I) = (((I - D_1) \cup (A_1 - D_2)) \cup D_2) \cup A_2
\]

\[
= (((I - D_1) \cup (A_1 - D_2)) - D_2) \cup A_2
\]

\[
= (((I - (D_1 - A_2)) \cup (A_1 - D_2)) - D_2) \cup A_2
\]

\[
= (((I - (D_1 - A_2)) - D_2) \cup (A_1 - D_2)) \cup A_2
\]

\[
= (((I - (D_1 - A_2)) \cup D_2) \cup (A_1 - D_2)) \cup A_2
\]

\[
= (o_1 + o_2)(I)
\]

\[ \Box \]

Unfortunately operation merging is not always associative. Consider the following counterexample: \( o_1 = (\emptyset, \emptyset, \{(1, 2)\}) \), \( o_2 = (\emptyset, \{(1, 2)\}, \emptyset) \) and \( o_3 = (\{(1, 2)\}, \emptyset, \emptyset) \). Then \( \langle o_1, o_2, o_3 \rangle \) is indeed an operation since we know that \( D_1 \cap A_1 = \emptyset \) and \( D_2 \cap A_2 = \emptyset \) and hence \((D_1 - A_2) \cup D_2 \cap ((A_1 - D_2) \cup A_2) = \emptyset \). However, if the three operations form a sound schedule, then associativity can be shown.

**Theorem 3.2** (Associativity of operation merging). For every sound schedule \( \langle o_1, o_2, o_3 \rangle \) it holds that \((o_1 + o_2) + o_3 = o_1 + (o_2 + o_3)\).
The case for $A \subseteq$ this criterium.

The results above can be easily adapted if we restrict operations such that $D \cap A = \emptyset$, since for every operation $(P, D, A)$ there is an equivalent operation $(P, D, A - P)$ that satisfies this criterium.

Observe that it follows that the merging of all operations in a sound schedule is both uniquely defined and semantically equivalent with the schedule.

**Proof.** Let $(P_4, D_4, A_4) = (o_1 + o_2) + o_3$ and $(P_5, D_5, A_5) = o_1 + (o_2 + o_3)$. Then:

$$
P_4 = (P_1 \cup (P_2 - A_1)) \cup (P_3 - ((A_1 - D_2) \cup A_2))
= (P_1 \cup (P_2 - A_1)) \cup (P_3 - (A_1 \cup A_2))
= P_1 \cup (P_2 - A_1) \cup ((P_3 - A_2) - A_1)
= (P_1 \cup ((P_2 \cup (P_3 - A_2)) - A_1))
= P_5
$$

**Proof.** Let $(P_4, D_4, A_4) = (o_1 + o_2) + o_3$ and $(P_5, D_5, A_5) = o_1 + (o_2 + o_3)$. Then:

$$
P_4 = (P_1 \cup (P_2 - A_1)) \cup (P_3 - ((A_1 - D_2) \cup A_2))
= (P_1 \cup (P_2 - A_1)) \cup (P_3 - (A_1 \cup A_2))
= P_1 \cup (P_2 - A_1) \cup ((P_3 - A_2) - A_1)
= (P_1 \cup ((P_2 \cup (P_3 - A_2)) - A_1))
= P_5
$$

$$
D_4 = ((D_1 - A_2) \cup D_2) - A_3) \cup D_3
= ((D_1 - A_2) - A_3) \cup (D_2 - A_3) \cup D_3
= (D_1 - ((A_2 - D_3) \cup A_3)) \cup ((D_2 - A_3) \cup D_3)
= D_5
$$

The case for $A_4$ and $A_5$ is symmetric to the case for $D_4$ and $D_5$ but with $A$ and $D$ swapped. $\square$

Under different syntactic restrictions this may not be so. For example, if we require that $D \subseteq P$ then merging is not always possible. Consider for example a schedule like

$$
S = (\langle 0^2, 0^- \rangle, \langle \{1, 2\}^+, \{1, 2\}^- \rangle, \langle 0^+ \rangle)
$$

that adds an edge and then deletes it. Its behavior cannot be simulated with a single operation with $D \subseteq P$ because it deletes an edge but does not require it to exist.

Also changing failure semantics may make merging impossible. For example, define that $(P, D, A)$ fails also if $A \cap I \neq \emptyset$. Then consider a schedule like

$$
S = (\langle 0^2, 0^- \rangle, \langle \{1, 2\}^+ \rangle, \langle \{1, 2\}^- \rangle, \langle 0^+ \rangle)
$$

that adds an edge and then deletes it. Its behavior cannot be simulated with a single operation because it fails if a certain edge exists and does nothing otherwise.

## 4 Cyclic instances

As an intermediate step towards restricting instances to documents we first consider cyclic instances that consist of disjoint simple cycles. Note that we can model a document as a single cycle with one special edge that connects the end node with the begin node and cannot be removed.

The set of nodes in a set of edges $X$ is denoted as $N_X$. The set of incoming and outgoing edges of a node $v$ in a set of edges $X$ is denoted as $in_X(v)$ and $out_X(v)$, respectively. The indegree and outdegree of a node $v$ in $X$ are denoted as $|in_X(v)|$ and $|out_X(v)|$, respectively.

**Proposition 4.1.** For a set of edges $X \subseteq N \times N$ and a node $v \in N$ it holds that $v \in N_X$ iff $|in_X(v)| > 0$ or $|out_X(v)| > 0$. 

---

6
Definition 4.1 (Cyclic instance). An instance $I$ is said to be cyclic if it is finite and for every node $v$ in $N_I$ it holds that $|in_I(v)| = |out_I(v)| = 1$.

In the following we define what a cyclically sound operation is. Intuitively this is an operation where at least for one instance all additions are real additions, i.e., the edges are not in the instance, the deletions are real deletions, and the nodes not appearing in the pattern or the delete set are completely new. These restrictions are chosen such that they represent what might be expected of a correct operation that is specified by the user.

Definition 4.2 (Cyclically well-formed operation). An operation $o = (P, D, A)$ is said to be cyclically well-formed if there is at least one cyclic instance $I$ such that $I \cap A = \emptyset$, $D \subseteq I$, $N_I \cap (N_A - N_{P,D}) = \emptyset$ and $o(I)$ is defined and a cyclic instance.

The following definition is a syntactic approximation of a cyclically sound operation.

Definition 4.3 (Cyclically well-formed operation). An operation $o = (P, D, A)$ is said to be cyclically well-formed if it holds that $P$, $D$ and $A$ are finite, for every node $v \in N_{P,D}$ it holds that $|in_{P,D}(v)| \leq 1$ and $|out_{P,D}(v)| \leq 1$, $P \cap A = \emptyset$ and for every node $v \in N_D \cup N_A$ one of the following holds:

- **cwf1** $|in_A(v)| = 1$, $|out_A(v)| = 1$, $|in_{P,D}(v)| = 0$, $|out_{P,D}(v)| = 0$
- **cwf2** $|in_A(v)| = 0$, $|out_A(v)| = 0$, $|in_D(v)| = 1$, $|out_D(v)| = 1$
- **cwf3** $|in_A(v)| = 1$, $|out_A(v)| = 0$, $|in_D(v)| = 1$, $|out_D(v)| = 0$
- **cwf4** $|in_A(v)| = 0$, $|out_A(v)| = 1$, $|in_D(v)| = 0$, $|out_D(v)| = 1$
- **cwf5** $|in_A(v)| = 1$, $|out_A(v)| = 1$, $|in_D(v)| = 1$, $|out_D(v)| = 1$

We first show that cyclical well-formedness is a sufficient condition for cyclical soundness.

Lemma 4.2. If an operation is cyclically well-formed then it is cyclically sound.

Proof. Assume that the operation $o = (P, D, A)$ is cyclically well-formed. We construct $I$ from $P \cup D$ by adding edges from each node with outdegree 0 to the node with indegree 0 if there is in $P \cup D$ a path from the last to the first. Since the indegree and outdegree of each node in $P \cup D$ is at most one, this results in a cyclic instance. It can be shown that $I \cap A = \emptyset$ as follows. By definition of an operation $D \cap A = \emptyset$ and we know that $P \cap A = \emptyset$, so $(P \cup D) \cap A = \emptyset$. Assume that an edge $(v_1, v_2)$ that was added to $P \cup D$ to construct $I$ is also in $A$. Then $|out_A(v_1)| \geq 1$ and so **cwf1**, **cwf4** or **cwf5** holds for $v_1$. Since $|in_{P,D}(v_1)| = 0$ rule **cwf1** cannot hold for $v_1$, so $|out_D(v_1)| = 1$. Let the outgoing edge of $v_1$ in $D$ be $(v_1, v_2)$ then $v_2 \neq v_3$ since $D \cap A = \emptyset$. Since $D \subseteq I$ it follows that $(v_1, v_3) \in I$. Since $I$ is cyclic it then holds that $(v_1, v_2) \notin I$. By the construction of $I$ it holds that $D \subseteq I$. It is also clear that $N_I \cap (N_A - N_{P,D}) = \emptyset$ since by construction $N_I = N_{P,D}$. By construction of $I$ it also holds that $P \subseteq I$, and so $o(I)$ is defined.

Finally we show that $o(I)$ is a cyclic instance. By construction $N_{o(I)} \subseteq N_P \cup N_D \cup N_A$. Let us first consider the case for nodes in $N_A - (N_P \cup N_D)$. These are not in $N_D$ and therefore must satisfy **cwf1** and so are not in $N_I$. It then follows that they have indegree and outdegree 1 in $o(I)$. Next we consider the nodes in $(N_P \cup N_D) - N_A$. These nodes then must satisfy **cwf2** and so, since $D \subseteq I$ and $I$ is cyclic, they will not be in $o(I)$. Finally we consider the nodes in $N_A \cap (N_P \cup N_D)$. These nodes must satisfy **cwf3**, **cwf4** or **cwf5**. Since $I \cap A = \emptyset$ and $D \subseteq I$ it is easily observed that in $o(I)$ their indegree and outdegree are 1 if they are 1 in $I$. 

\[\square\]
In the following we show that in fact a weaker condition than cyclical soundness is already sufficient condition for cyclical well-formedness.

**Lemma 4.3.** If for an operation $o = (P, D, A)$ there is at least one cyclic instance $I$ such that $I \cap A = \emptyset$, $D \subseteq I$ and $o(I)$ is defined and a cyclic instance, then $o$ is cyclically well-formed.

**Proof.** Assume that $o = (P, D, A)$ is not cyclically well-formed and $I$ is a cyclic instance such that $I \cap A = \emptyset$, $D \subseteq I$, and $o(I)$ is defined. We can then show that $o(I)$ is not a cyclic instance as follows. Clearly, $P$ is finite because $o(I)$ is defined and $I$ is a cyclic instance. The set $D$ is finite since $D \subseteq I$ and $I$ is a cyclic instance. If the set $A$ is finite then $o(I)$ is also infinite and therefore not a cyclic instance. In the remainder of this proof we consider the case where $A$ is finite. Since $o(I)$ is defined and $D \subseteq I$ it holds that $P \cup D \subseteq I$, and since $I$ is cyclical it follows that there is no node in $N_{P \cup D}$ that has in $P \cup D$ an indegree or outdegree larger than 1. Since $o(I)$ is defined and we assumed that $I \cap A = \emptyset$ it follows that $P \cap A = \emptyset$. Since operation $o$ is not cyclically well-formed, there must then be a node $v$ in $N_D \cup N_A$ that violates all of $cwf_1$, $\ldots$, $cwf_5$. If $|in_A(v)| > 1$ or $|out_A(v)| > 1$ then the same holds for $o(I)$ and so it will not be a cyclic instance. In the remainder of this proof we consider the case where $|in_A(v)| \leq 1$ and $|in_A(v)| \leq 1$. It holds that either $v \in N_A - N_D$, $v \in N_D - N_A$ or $v \in N_D \cap N_A$. We consider the three cases:

- **Assume that** $v \in N_A - N_D$. Since $cwf_1$ not holds for $v$ and $v \in N_A$, it follows that (a) $|in_A(v)| \neq |out_A(v)|$ or (b) $v \in N_{P} - N_{D}$. If (a) then since $|in_I(v)| = |out_I(v)|$, $I \cap A = \emptyset$ and $|in_D(v)| = |out_D(v)| = 0$ it follows that $|in_{o(I)}(v)| \neq |out_{o(I)}(v)|$, and therefore $o(I)$ is not a cyclic instance. If (b) then because $|in_I(v)| + |out_I(v)| = 2$, $I \cap A = \emptyset$, $|in_A(v)| + |out_A(v)| > 0$ and $|in_D(v)| = |out_D(v)| = 0$ it follows that $|in_{o(I)}(v)| + |out_{o(I)}(v)| > 2$, and therefore $o(I)$ is not a cyclic instance.

- **Assume that** $v \in N_D - N_A$. Since $cwf_2$ not holds for $v$ and $v \in N_D$ it follows that $|in_D(v)| \neq |out_D(v)|$, and so, since $|in_I(v)| = |out_I(v)|$ and $|in_A(v)| = |out_A(v)| = 0$, it follows that $|in_{o(I)}(v)| \neq |out_{o(I)}(v)|$, and therefore $o(I)$ is not a cyclic instance.

- **Assume that** $v \in N_A \cap N_D$. Since $cwf_3$, $cwf_4$ and $cwf_5$ not hold for $v$ it follows that $|in_A(v)| \neq |in_D(v)|$ or $|out_A(v)| \neq |out_D(v)|$. Then, since $|in_I(v)| = |out_I(v)|$ and $I \cap A = \emptyset$, it follows that $|in_{o(I)}(v)| \neq |in_I(v)|$ or $|out_{o(I)}(v)| \neq |out_I(v)|$, and therefore $o(I)$ is not a cyclic instance.

**Theorem 4.4.** An operation is cyclically sound iff it is cyclically well-formed.

**Proof.** The if direction follows from Lemma 4.2, and the only-if direction from Lemma 4.3 since the condition mentioned there for $I$ is weaker then the condition for $I$ in the definition of cyclic soundness.

It follows that we also might have used the weaker condition of Lemma 4.3 to define cyclical soundness, i.e., there must be an instance such that the additions are real additions, the deletions are real deletions, and the result is defined and a cyclic instance.

**Corollary.** An operation $o = (P, D, A)$ is cyclically sound iff there is at least one cyclic instance $I$ such that $I \cap A = \emptyset$, $D \subseteq I$ and $o(I)$ is defined and a cyclic instance.

The following theorem states that the a cyclically well-formed operation will always return a cyclic instance if its deletions are real deletions, its additions are real additions and the new nodes are indeed new nodes.
**Theorem 4.5.** If an operation \( o = (P, D, A) \) is cyclically well-formed and \( I \) a cyclic instance such that \( D \subseteq I \), \( A \cap I = \emptyset \), \( N_I \cap (N_A - N_{P \cup D}) = \emptyset \) and \( o(I) \) is defined, then \( o(I) \) is a cyclic instance.

**Proof.** All nodes in \( o(I) \) are either in \( D \cup A \) or not. In the latter case their indegree and outdegree is the same in \( I \) and \( o(I) \). In the first case it holds that these nodes are in \( N_D \cup N_A \). Let us first consider the case for nodes in \( N_A - N_D \). These are not in \( N_D \) and therefore must satisfy \text{cwf1} \ and therefore, because \( N_I \cap (N_A - N_{P \cup D}) = \emptyset \), it is not in \( I \) and so they have indegree and outdegree 1 in \( o(I) \).

Next we consider the nodes in \( N_D - N_A \). These nodes then must satisfy \text{cwf2} \ and so \( o \) removes the only two edges containing these nodes and they will not be in \( o(I) \).

Finally we consider the nodes in both \( N_A \cap N_D \). These nodes must satisfy \text{cwf3, cwf4} \ or \text{cwf5}.

Then, using \( A \cap I = \emptyset \) and \( D \subseteq I \), it is easily observed that in \( o(I) \) their indegree and outdegree are 1 if they are 1 in \( I \).

The following theorem states that if the result of cyclically well-formed operation is defined and it deletions are real deletions and the new nodes are indeed new nodes, then its additions are real additions.

**Theorem 4.6.** For every cyclic instance \( I \) and cyclically well-formed operation \( o = (P, D, A) \) it holds that if \( D \subseteq I \), \( N_I \cap (N_A - N_{P \cup D}) = \emptyset \) and \( o(I) \) is defined then \( A \cap I = \emptyset \).

**Proof.** Consider an edge \( (v_1, v_2) \in A \). If \( v_1 \) or \( v_2 \) is not in \( N_{P \cup D} \) then \( (v_1, v_2) \notin I \) since \( N_I \cap (N_A - N_{P \cup D}) = \emptyset \). Assume that \( v_1 \) and \( v_2 \) are in \( N_{P \cup D} \). It follows that \( v_1 \) must satisfy \text{cwf4} \ or \text{cwf5}.

It follows that \( |\text{out}_D(v_1)| = 1 \). Let the outgoing edge of \( v_1 \) in \( D \) be \((v_1, v_3) \) then, since \( A \cap D = \emptyset \), \( v_3 \neq v_2 \). Since \( D \subseteq I \) it follows that \( (v_1, v_3) \in I \). Since \( I \) is cyclical it then holds that \( (v_1, v_2) \notin I \).

An important restriction seems to be where the delete set is a subset of the pattern, i.e., it is checked whether all edges that are deleted are indeed present in the instance. We will therefore give this property a name.

**Definition 4.4 (Well-guarded operation).** An operation \( (P, D, A) \) is said to be well-guarded if \( D \subseteq P \).

The following theorem states that if the result of a well-guarded cyclically well-formed operation is defined and the new nodes are indeed new nodes, then the result is a cyclic instance.

**Theorem 4.7.** For every cyclic instance \( I \) and well-guarded cyclically well-formed operation \( o = (P, D, A) \) such that \( N_I \cap (N_A - N_{P \cup D}) = \emptyset \) and \( o(I) \) is defined then \( o(I) \) is a cyclic instance.

**Proof.** If \( D \subseteq P \) and \( o(I) \) is defined then \( D \subseteq I \). It then follows by Theorem 4.6 that \( A \cap I = \emptyset \).

We can then apply Theorem 4.5 and derive that \( o(I) \) is a cyclic instance.

Finally we show that for well-guarded operations the notion of soundness might even have been defined by a weaker condition that states that there is an instance such that the new nodes are indeed new nodes and the result is defined and a cyclic instance.

**Theorem 4.8.** For every well-guarded operation \( o = (P, D, A) \) it holds that \( o \) is cyclically well-formed if there is an instance \( I \) such that \( N_I \cap (N_A - N_P) \), \( o(I) \) is defined and a cyclic instance.

**Proof.** We first prove the if part. Since \( D \subseteq P \) and \( o(I) \) is defined it holds that \( D \subseteq I \). By Theorem 4.6 it follows that \( A \cap I = \emptyset \). It therefore holds that \( o \) is cyclically sound, and therefore by Theorem 4.4 it is cyclically well-formed. The only-if part follows straightforwardly from Theorem 4.4 in the other direction.
5 Accumulation of cyclic operations

As pointed out before well-guardedness seems problematic for aggregation, but at the same time it seems very natural for document operations, and aggregation of document operations also seems natural. So how can this be resolved?

We begin with a notion of soundness for schedules that is much more restrictive than the previous notion of soundness. Intuitively it requires that each step of the schedule is strict in the sense that all additions and deletions are real additions and real deletions, respectively, and that the new nodes are indeed new nodes.

Definition 5.1 (Cyclically sound schedule). A schedule $S = \langle o_1, \ldots, o_n \rangle$ with $o_i = (P_i, D_i, A_i)$ for all $1 \leq i \leq n$, is said to be sound if there is a sequence of cyclic instances $\langle I_0, \ldots, I_n \rangle$ such that for all $1 \leq i \leq n$ it holds that (1) $I_{i-1} \cap A_i = \emptyset$, (2) $D_i \subseteq I_{i-1}$, (3) for all $0 \leq j < i$ it holds that $N_{I_j} \cap (N_{A_i} - N_{P_i \cup D_i}) = \emptyset$ and (4) $o_i(I_{i-1})$ is defined and equal to $I_i$.

Proposition 5.1. If a schedule is cyclically sound then it is sound.

Proof. Follows from the definition of a sound schedule.

Definition 5.2 (Cyclically well-formed schedule). A schedule $S = \langle o_1, \ldots, o_n \rangle$ with $o_i = (P_i, D_i, A_i)$ for all $1 \leq i \leq n$, is said to be cyclically well-formed if

- **cwfs1** for all $1 \leq i \leq n$ each $o_i$ is a cyclically well-formed operation,
- **cwfs2** for all $1 \leq i < j \leq n$ it holds that $N_{P_i \cup D_i \cup A_i} \cap (N_{A_j} - N_{P_j \cup D_j}) = \emptyset$ and
- **cwfs3** for all $1 \leq i < j \leq n$ and edges $(v_1, v_2) \in P_j \cup D_j$, if
  - (a) $(v_1, v_2) \in D_i$,
  - (b) there is an edge $(v_1, v_3) \in P_{i+1} \cup D_{i+1} \cup A_i$ with $v_2 \neq v_3$ or
  - (c) there is an edge $(v_4, v_2) \in P_{i+1} \cup D_{i+1} \cup A_i$ with $v_4 \neq v_1$,

then there is a $i < k < j$ such that $(v_1, v_2) \in A_k$.

Theorem 5.2. A schedule is cyclically sound iff it is cyclically well-formed.

Proof. It is easy to verify that a not cyclically well-formed schedule is not cyclically sound.

Assume that a schedule $S = \langle o_1, \ldots, o_n \rangle$ with $o_i = (P_i, D_i, A_i)$ for all $1 \leq i \leq n$, is cyclically well-formed. We construct a sequence of instances $\langle I_0, \ldots, I_{n-1} \rangle$ that intuitively represent the minimal intermediate instances required by a sound schedule. It is defined such that $I_{n-1} = P_n \cup D_n$ and $I_{i-1} = (I_i - A_i) \cup (P_i \cup D_i)$ for all $1 \leq i \leq n - 1$. It then can be shown by induction on $n - 1 - i$ that for every $0 \leq i \leq n - 1$ and every edge $(v_1, v_2) \in N \times N$ that $(v_1, v_2) \in I_i$ iff there is $i < j \leq n$ such that $(v_1, v_2) \in P_j \cup D_j$ and there is no $i < k < j$ such that $(v_1, v_2) \in A_k$.

It follows by **cwfs3** (a) that $D_i \cap I_i = \emptyset$ for all $1 \leq i \leq n - 1$. We now define a sequence $\langle I'_0, \ldots, I'_n \rangle$ such that $I'_0 = I_0$ and $I'_i = (I'_{i-1} - D_i) \cup A_i$ for all $1 \leq i \leq n$. We can then show by induction on $i$ that $I'_i \supseteq I_i$: for $i = 0$ it clearly holds that $I'_0 \supseteq I_0$, and for $0 < i < n$ it holds that

$$I'_i = (I'_{i-1} - D_i) \cup A_i$$
$$\supseteq (I_{i-1} - D_i) \cup A_i$$
$$\supseteq ((I_i - A_i) \cup (P_i \cup D_i)) - D_i \cup A_i$$
$$= (I_i - A_i) \cup (P_i - D_i) \cup A_i$$
using $D_i \cap I_i = \emptyset$
$$= I_i \cup (P_i - D_i) \cup A_i$$
$$\supseteq I_i$$
It then follows that \( S(P'_i) \) is defined and \( P_i \cup D_i \subseteq I'_{i-1} \) for all \( 1 \leq i \leq n \). It also holds that \( (N_{A_i} - N_{P_i \cup D_i}) \cap N_{I'_{i-1}} = \emptyset \) which can be shown as follows. First we show that \( (N_{A_i} - N_{P_i \cup D_i}) \cap N_{I_0} = \emptyset \) as follows. Assume that there is an edge \((v_1, v_2) \in P_j \cup D_j \) for some \( i < j \leq n \), such that \( \{v_1, v_2\} \cap (N_{A_i} - N_{P_i \cup D_i}) \neq \emptyset \). Then there is either an edge \((v_1, v_2) \in A_i \) or, if not then, by \textbf{cwsf3 (b), (c)} there is an \( i < k < j \) such that \( (v_1, v_2) \in A_k \). It follows that for every such \((v_1, v_2) \) it holds that \( (v_1, v_2) \notin I'_{i-1} \), and hence \( N_{I'_{i-1}} \cap (N_{A_i} - N_{P_i \cup D_i}) = \emptyset \). Moreover, by \textbf{cwsf2} it holds that \( N_{P_i \cup D_i} \cap (N_{A_i} - N_{P_i \cup D_i}) = \emptyset \) for every \( 1 \leq j < i \) and so \( N_{I_0} \cap (N_{A_i} - N_{P_i \cup D_i}) = \emptyset \).

By \textbf{cwsf2} it holds that \( N_{A_i} \cap (N_{A_i} - N_{P_i \cup D_i}) = \emptyset \) for all \( 1 \leq i < j \), and so it follows that \( N_{I'_{i-1}} \cap (N_{A_i} - N_{P_i \cup D_i}) = \emptyset \).

By \textbf{cwsf3 (b), (c)} it holds that for every node \( v \in N_{I_0} \) that \( |in_{I_0}(v)| \leq 1 \) and \( |out_{I_0}(v)| \leq 1 \). We then construct \( I''_0 \) from \( I_0 \) by closing each path with an edge from \( v \) to a fresh node. We then again define \( I''_i = (I''_{i-1} - D_i) \cup A_i \) for all \( 1 \leq i \leq n \). Since the new edges in \( I''_0 \) concern fresh nodes not in \( S \) it holds that for every \( 1 \leq i \leq n \) it holds that \( I''_i = \emptyset \). Since \( I''_n \) is a cyclic instance and every operation is a cyclically well-formed operation it then follows by Theorem 4.6 that \( A_i \cap I''_{i-1} = \emptyset \) for all \( 1 \leq i \leq n \). It then follows by Theorem 4.5 that \( I''_i \) is a cyclic instance for all \( 1 \leq i \leq n \).

**Lemma 5.3.** For a cyclically well-formed schedule \( S = \langle o_1, \ldots, o_n \rangle \) with \( o_i = (P_i, D_i, A_i) \) for all \( 1 \leq i \leq n \), it holds that for all \( 1 \leq i < j \leq n \) and edges \((v_1, v_2) \in A_i \cap A_j \) there is a \( i < k < j \) such that \( (v_1, v_2) \in D_k \).

**Proof.** The proof proceeds by reductio ad absurdum. Assume that \((v_1, v_2) \in A_i \cap A_j \) and there is no \( i < k < j \) such that \((v_1, v_2) \in D_k \). If \( S \) is cyclically well-formed then by Theorem 5.2 it is cyclically sound and an appropriate sequence \( \langle I_0, \ldots, I_n \rangle \) of cyclic instances exists. However, in that sequence it will hold that \((v_1, v_2) \in I_i \) and since it is not deleted by \( o_{i+1}, \ldots, o_{j-1} \) it also holds that \((v_1, v_2) \in I_{j-1} \). However, since \((v_1, v_2) \in A_j \) this contradicts that \( \langle I_0, \ldots, I_n \rangle \) is a sequence that shows that \( S \) is cyclically sound.

We now introduce a new notion of operation merging that is slightly different from the preceding one. It is designed such that it will result under certain circumstances in a cyclically well-formed operation if the original operations are well-formed.

**Definition 5.3** (Strict operation merging). Let \( o_1 = (P_1, D_1, A_1) \) and \( o_2 = (P_2, D_2, A_2) \) be two operations then \( (o_1 \oplus o_2) = (P_3, D_3, A_3) \) where \( P_3 = P_1 \cup (P_2 - A_1), D_3 = (D_1 - A_2) \cup (D_2 - A_1) \) and \( A_3 = (A_1 - D_2) \cup (A_2 - D_1) \).

**Proposition 5.4.** If \( o_1 \) and \( o_2 \) are operations then \( o_1 \oplus o_2 \) is an operation.

**Proof.** This follows straightforward from the following observations: \( (D_1 - A_2) \cap (A_1 - D_2) = \emptyset \) since \( D_1 \cap A_1 = \emptyset \), \( (D_1 - A_2) \cap (A_2 - D_1) = \emptyset \), \( (D_2 - A_1) \cap (A_1 - D_2) = \emptyset \), and \( (D_2 - A_1) \cap (A_2 - D_1) = \emptyset \) since \( D_2 \cap A_2 = \emptyset \).

**Proposition 5.5.** If \( o_1 \) and \( o_2 \) are well-guarded then \( o_1 \oplus o_2 \) is well-guarded.

**Proof.** If \( D_1 \subseteq P_1 \) and \( D_2 \subseteq P_2 \) then \( (D_1 - A_2) \cup (D_2 - A_1) \subseteq P_1 \cup (P_2 - A_1) \).

It is not the case that this type of merging maintains the exact semantics, even if we only consider pairs that form a cyclically sound schedule with well-guarded operations. A counterexample is where \( o_1 = (\emptyset, \emptyset^*, \{(1, 2), (2, 1)\})^* \) and \( o_2 = (\{(1, 2), (2, 1)\}^*, \{(1, 2), (2, 1)\})^* \), and so \( o_1 \oplus o_2 = (\emptyset^*, \emptyset^*, \emptyset^*) \), which is not the same as \( o_1, o_2 \) since that removes the cycle if it exists.

The following theorem establishes that the definedness of a sound schedule of two operations does not change if we merge them into one operation.
Lemma 5.6. If \( \langle o_1, o_2 \rangle \) is a sound schedule then for all instances \( I \) it holds that \( (o_1 \oplus o_2)(I) \) is defined iff \( \langle o_1, o_2 \rangle(I) \) is defined.

**Proof.** The result of \( (o_2 \circ o_1)(I) \) is defined iff \( P_1 \subseteq I \) and \( P_2 \subseteq (I - D_1) \cup A_1 \). Since the schedule is sound we know that \( D_2 \cap P_2 = \emptyset \). Since \( P_2 - A_1 \subseteq I \) iff \( P_2 \subseteq I \cup A_1 \) it follows that \( P_2 - A_1 \subseteq I \) iff \( P_2 \subseteq (I - D_1) \cup A_1 \). It follows that \( P_1 \cup (P_2 - A_1) \subseteq I \) iff \( P_1 \subseteq I \) and \( P_2 \subseteq (I - D_1) \cup A_1 \). □

The following theorem establishes that the result of two operations does not change if we merge them, under certain reasonable assumptions (which are satisfied always if the operation is well-guarded and the new nodes are indeed new nodes).

Lemma 5.7. For operations \( o_1 = (P_1, D_1, A_1) \) and \( o_2 = (P_2, D_2, A_2) \) it holds for every instance \( I \) that if \( (o_1 \circ o_2)(I) \) and \( \langle o_1, o_2 \rangle(I) \) are both defined, \( D_1 \subseteq I \) and \( A_1 \cap I = \emptyset \) then \( (o_1 \circ o_2)(I) = \langle o_1, o_2 \rangle(I) \).

**Proof.**

\[
\langle o_1, o_2 \rangle(I) = (((I - D_1) \cup A_1) - D_2) \cup A_2
= (((I \cup A_1) - D_1) - D_2) \cup A_2
= (((I \cup A_1) - D_2) - D_1) \cup A_2
= (((I - D_2) \cup (A_1 - D_2)) - D_1) \cup A_2
= (((I - (D_2 - A_1)) \cup (A_1 - D_2)) - D_1) \cup A_2
= (((I - (D_2 - A_1)) \cup D_1) \cup (A_1 - D_2)) \cup A_2
= (((I - (D_1) - (D_2 - A_1)) \cup (A_1 - D_2)) \cup A_2
= (((I - (D_1)) \cup (A_1 - D_2)) \cup (A_1 - D_2)) \cup A_2
= (((I - (D_1) \cup A_2) - (D_2 - A_1)) \cup (A_1 - D_2)) \cup A_2
= (((I \cup A_2) - (D_1 - A_2)) - (D_2 - A_1)) \cup (A_1 - D_2)
= (((I \cup (A_2 - D_1)) - (D_1 - A_2)) - (D_2 - A_1)) \cup (A_1 - D_2)
= (((I - (D_1) - (A_2 - D_1)) - (D_2 - A_1)) \cup (A_1 - D_2)) \cup (A_1 - D_2)
= (((I - (D_1) - (D_2 - A_1)) \cup (A_1 - D_2)) \cup (A_1 - D_2)) \cup (A_1 - D_2)
= (o_1 \circ o_2)(I)
\]

We can now show that operation merging does not change the semantics if we consider sound schedules of two well-guarded cyclically well-formed operations and the nodes that are added by the first operation are fresh nodes. Note that this implies that this holds also for cyclically sound schedules of well-guarded operations that add only fresh nodes.

**Theorem 5.8.** If \( \langle o_1, o_2 \rangle \) is a sound schedule with cyclically well-formed, well-guarded operations \( o_1 = (P_1, D_1, A_1) \) and \( o_2 = (P_2, D_2, A_2) \), and \( I \) is a cyclic instance such that \( N_I \cap (N_{A_1} - N_{P_1 \cup D_1}) = \emptyset \), then it holds that \( (I) \langle o_1, o_2 \rangle(I) \) is defined iff \( (o_1 \circ o_2)(I) \) is defined and (2) if \( \langle o_1, o_2 \rangle(I) \) and \( (o_1 \circ o_2)(I) \) are defined then \( \langle o_1, o_2 \rangle(I) = (o_1 \circ o_2)(I) \).
Proof. Clearly (1) follows directly from Lemma 5.6. To see that (2) holds consider the following. Since the operations are well-guarded and \((o_1 \oplus o_2)(I)\) is defined, it follows that \(D_1 \subseteq P_1 \subseteq I\). Since \(o_1\) is a cyclically well-formed operation and \(N_I \cap (N_{A_1} - P_{\cup D_1}) = \emptyset\) it then follows by Theorem 4.6 that \(A_1 \cap I = \emptyset\). It then follows by Lemma 5.7 that \(\langle o_1, o_2 \rangle(I) = (o_1 \oplus o_2)(I)\).

**Corollary.** If \((o_1, o_2)\) is a cyclically sound schedule with well-guarded operations \(o_1 = (P_1, D_1, A_1)\) and \(o_2 = (P_2, D_2, A_2)\), and \(I\) is a cyclic instance such that \(N_I \cap (N_{A_1} - P_{\cup D_1}) = \emptyset\) and \(N_I \cap (N_{A_2} - P_{\cup D_2}) = \emptyset\), then it holds that (1) \((o_1, o_2)(I)\) is defined iff \((o_1 \oplus o_2)(I)\) is defined and (2) \((o_1, o_2)(I)\) and \((o_1 \oplus o_2)(I)\) are defined then \((o_1, o_2)(I) = (o_1 \oplus o_2)(I)\).

**Lemma 5.9.** Let \(o_1 = (P_1, D_1, A_1)\) and \(o_2 = (P_2, D_2, A_2)\) be two operations, \(I\) an instance such that \(D_1 \subseteq I\) and \(D_2 \subseteq o_1(I)\), then if \(o_1 \oplus o_2 = (P_3, D_3, A_3)\) then \(D_3 \subseteq I\).

**Lemma 5.10.** Let \(o_1 = (P_1, D_1, A_1)\) and \(o_2 = (P_2, D_2, A_2)\) be two operations, \(I\) an instance such that \(A_1 \cap I = \emptyset\) and \(A_2 \cap o_1(I) = A_2 \cap ((I - D_1) \cup A_1) = \emptyset\). From the first it follows that \((A_1 - D_2) \cap I = \emptyset\). From the second it follows that \(A_2 \cap (I - D_1) = \emptyset\) and therefore \((A_2 - D_1) \cap I = (A_2 \cap I) - D_1 = A_2 \cap (I - D_1) = \emptyset\). It then follows that \(A_3 \cap I = ((A_1 - D_2) \cup (A_2 - D_1)) \cap I = \emptyset\).

**Lemma 5.11.** If \(o = (P, D, A)\) is a cyclically well-formed operation, \(I\) a cyclic instance, \(D \subseteq I\), \(A \cap I = \emptyset\) and \(o(I)\) is defined and a cyclic instance, then \(N_I \cap (N_A - P_{\cup D}) = \emptyset\).

**Proof.** Assume that there is a node \(v \in N_I \cap (N_A - P_{\cup D})\). Since \(o\) is cyclically well-formed it holds that one of \(\text{cwf}1, \ldots, \text{cwf}5\) applies. Since \(v \not\in N_{\cup D_{1\cup A}}\) it must be \(\text{cwf}1\) that applies and so there are two edges \((v_1, v)\) and \((v, v_2)\) in \(A\). However, since \(v \in N_I\) and \(I\) is cyclic, there are edges \((v_1', v)\) and \((v, v_2')\) in \(I\). Since \(A \cap I = \emptyset\) it follows that \(v_1 \neq v_1'\) and \(v_2 \neq v_2'\). Since \(\text{cwf}1\) applies it also holds that \((v_1, v)\) and \((v, v_2)\) are not in \(D\). It then follows that in \(o(I)\) we have the edges \((v_1, v), (v_1', v), (v, v_2)\) and \((v, v_2')\), and so \(o(I)\) is not a cyclic instance, which leads to a contradiction. Hence the assumption that there is a node in \(N_I \cap (N_A - P_{\cup D})\) must be false.

**Lemma 5.12.** If \(o_1 = (P_1, D_1, A_1)\) and \(o_2 = (P_2, D_2, A_2)\) are cyclically well-formed operations and \(o_1 \oplus o_2 = (P_3, D_3, A_3)\) then \((N_{A_1} - P_{\cup D_1}) \subseteq (N_{A_1} - P_{\cup D_1}) \cup (N_{A_1} - P_{\cup D_1})\).

**Proof.** For a cyclically well-formed operation \((P, D, A)\) it holds that for every node \(v \in N_{\cup D_{1\cup A}}\) that either \(v \in N_{\cup D_{1\cup A}}\) or \(v\) satisfies one of \(\text{cwf}1, \ldots, \text{cwf}5\). We assume that \(\text{cwf}1\) does not apply to \(v\) in \(o_1\) or in \(o_2\) and consider all remaining combinations for \(o_1\) and \(o_2\):

- If \(v \not\in N_{D_{1\cup A_1}}\) and \(v \not\in N_{D_{2\cup A_2}}\) then \(v \not\in N_{A_1} - P_{\cup D_1}\) since \(v \not\in N_{A_1}\).
- If \(v \not\in N_{D_{1\cup A_1}}\) and \(v\) satisfies one of \(\text{cwf}1, \ldots, \text{cwf}5\) in \(o_2\) then \(|\text{in}_{D_{1\cup A_1}}(v)| > 0\) or \(|\text{out}_{D_{1\cup A_1}}(v)| > 0\), and so \(v \not\in N_{A_2} - P_{\cup D_2}\).
- If \(v\) satisfies one of \(\text{cwf}1, \ldots, \text{cwf}5\) in \(o_1\) and \(v \not\in N_{D_{2\cup A_2}}\) then by a similar argument as before it follows that \(v \not\in N_{A_3} - P_{\cup D_3}\).
- If \(\text{cwf}2\) holds for \(v\) in \(o_1\) and one of \(\text{cwf}2, \ldots, \text{cwf}4\) holds for \(v\) in \(o_2\) then at least one of the two edges in \(D_1\) that contain \(v\) will still be present in \(D_3\), and so \(v \in N_{D_{3\cup A_3}}\), and therefore \(v \not\in N_{A_3} - P_{\cup D_3}\).
Theorem 5.14. For every schedule \( S = \langle o_1, \ldots, o_n \rangle \) with \( o_i = (P_i, D_i, A_i) \) for all \( 1 \leq i \leq n \), and schedule \( S' = \langle o_1, \ldots, o_{j-1}, (o_j \oplus o_{j+1}), o_{j+2}, \ldots, o_n \rangle \) and I an instance such that \( N_I \cap (N_{A_i} - N_{P_{j} \cup D_{j}}) = 0 \) for all \( 1 \leq i \leq n \), it holds that (1) \( S(I) \) is defined iff \( S'(I) \) is defined and (2) if \( S(I) \) and \( S'(I) \) are defined then \( S(I) = S'(I) \).
Proof. Up to operation $o_j$ the intermediate results will be the same for $S$ and $S'$. Moreover, we know by assumption that $N_I \cap (N_{A_j} - N_{P_j \cup D_j}) = \emptyset$ and since $S$ is cyclically well-formed and by Lemma 5.13 it follows that $N_{I_{j-1}} \cap (N_{A_j} - N_{P_j \cup D_j}) = \emptyset$ where $I_{j-1} = (o_1, \ldots, o_{j-1})(I)$. Then by Theorem 5.8 it follows that the result after $o_{j+1}$ in $S$ and after $(o_j \oplus o_{j+1})$ in $S'$ is in both cases defined or undefined and the same if they are both defined. Since the remainders of the schedules are the same it follows that the same holds for $S'(I)$ and $S(I)$. \hfill \Box

Theorem 5.15. For every cyclically well-formed schedule $S = \langle o_1, \ldots, o_n \rangle$ it holds that $S' = \langle o_1, \ldots, o_{j-1}, (o_j \oplus o_{j+1}), o_{j+2}, \ldots, o_n \rangle$ is also cyclically well-formed.

Proof. If $S$ is cyclically well-formed then by Theorem 5.2 it is cyclically sound. Then there is a sequence of cyclic instances $(I_0, \ldots, I_n)$ such that for all $1 \leq i \leq n$ it holds that (1) $I_{i-1} \cap A_i = \emptyset$, (2) $D_i \subseteq I_{i-1}$, (3) for all $0 \leq j < i$ it holds that $N_{I_j} \cap (N_{A_i} - N_{P_i \cup D_i}) = \emptyset$ and (4) $o_i(I_{i-1})$ is defined and equal to $I_i$. We can then show that the sequence $(I_0, \ldots, I_{j-1}, I_{j+1}, \ldots, I_n)$ has also these properties for $S'$ as follows.

We first show that (4) holds. Since the first $j-1$ operations are the same in $S$ and $S'$ this holds for $o_1, \ldots, o_{j-1}$. Since $S$ is cyclically well-formed, it is also sound, and therefore $(o_j, o_{j+1})$ is sound, and then by Lemma 5.6 and the fact that $(o_{j+1} \circ o_j)(I)$ is defined it follows that $(o_j \oplus o_{j+1})(I_{j-1})$ is defined. Moreover, we know that $D_j \subseteq I_{j-1}$ and $A_j \cap I_{j-1} = \emptyset$, and so by Lemma 5.7 it follows that $(o_j \oplus o_{j+1})(I_{j-1}) = I_{j+1}$. Since the remainder of $S$ and $S'$ are the same, there (4) also holds.

Next we show that (1) holds. Clearly it holds for $I_0, \ldots, I_{j-1}$ and $I_{j+1}, \ldots, I_n$ since (1) holds for $S$ and $(o_1, \ldots, o_n)$. By Lemma 5.10 it also holds for $I_j$ and $o_j \oplus o_{j+1}$. By a similar argument and using Lemma 5.9 it follows that (2) holds.

Finally we show that (3) holds. This follows from the fact that (3) holds for $S$ and $(o_1, \ldots, o_n)$, and Lemma 5.13.

We have now shown that $S'$ is cyclically sound and it then follows by Theorem 5.2 that it is cyclically well-formed. \hfill \Box

Corollary. Let $(o_1, o_2)$ be a cyclically well-formed schedule then $o_1 \oplus o_2$ is a cyclically well-formed operation.

Note. Other things that might still be interesting to prove: Merging of operations is associative.

6 Locking cyclic operations

Definition 6.1 (Lock-Conflicting operations). Two operations $o_1 = (P_1, D_1, A_1)$ and $o_2 = (P_2, D_2, A_2)$ are said to conflict if at least one of the following holds:

lc1. $P_1 \cap D_2 \neq \emptyset$

lc2. $P_2 \cap D_1 \neq \emptyset$

lc3. $P_1 \cap A_2 \neq \emptyset$

lc4. $P_2 \cap A_1 \neq \emptyset$

lc5. $D_1 \cap A_2 \neq \emptyset$

lc6. $D_2 \cap A_1 \neq \emptyset$

lc7. $A_1 \cap A_2 \neq \emptyset$
lc8. $D_1 \cap D_2 \neq \emptyset$

*Note.* The conditions lc1, . . . , lc6 are equal to c1, . . . , c6 of the definition of conflicting operations. Only the conditions lc7 and lc8 are new.

**Definition 6.2** (Lock-conflict graph). The lock-conflict graph of a schedule $S = \langle o_1, \ldots, o_n \rangle$ is $G'_S = (V,E)$ where $V = \{1, \ldots, n\}$ and $E$ contains the edge $(i,j)$ iff $i < j$ and $o_i$ and $o_j$ lock-conflict.

**Theorem 6.1.** For every cyclically sound schedule $S$ and edge $(i,j)$ in $G'_S$ there is a path from $i$ to $j$ in $G_S$.

**Proof.** If $S$ is cyclically sound the by Theorem 4.4 it is cyclically well-formed. Assume that $(i,j) \in G'_S$. If this is because one of lc1, lc2, lc3, lc4, lc5 or lc6 holds then because these conditions are those that define when two operations conflict and therefore whether there is an edge in $G_S$ it follows that $(i,j) \in G_S$. If lc7 holds because of an edge $(v_1, v_2) \in A_1 \cap A_2$ then by Lemma 5.3 there is a $i < k < j$ such that $(v_1, v_2) \in D_k$ and so there is in $G_S$ an edge $(i,k)$ because of c6 and an edge $(k,j)$ because of c5. If lc8 holds because of an edge $(v_1, v_2) \in D_1 \cap D_2$ then by cwfs3 (a) there is a $i < k < j$ such that $(v_1, v_2) \in A_k$ and so there is in $G_S$ an edge $(i,k)$ because of c5 and an edge $(k,j)$ because of c6. \qed