Mesochronal Structure Learning: Supplementary Materials

1 Proofs of Lemmas

These supplemental materials contain the proofs for the lemmas in the main paper, presented largely without any exposition. Throughout, we assume knowledge of the terminology and notation provided in the main paper.

Lemma 3.1. ***Conflict persistence***: *If a virtual node identification results in a conflict, then no further node identifications will eliminate that conflict.*

Proof: The identification of the virtual node for $A \rightarrow B$ with node X corresponds to the addition of $A \rightarrow X$ and $X \rightarrow B$ to $\mathcal{G}^1$. By definition, a conflict means that there is at least one edge $S \rightarrow T$ or $S \leftrightarrow T$ in $\mathcal{G}^2$ that is not found in $\mathcal{H}^2$. Edge addition in $\mathcal{G}^1$ monotonically (though not strictly) increases the sets of length-2 paths and balanced forks of length 1 in $\mathcal{G}^1$, and so monotonically increases the set of edges in $\mathcal{G}^2$. Thus, once $S \rightarrow T$ or $S \leftrightarrow T$ is in $\mathcal{G}^2$, further node identifications cannot remove that edge, and so the conflict will persist. ∎

Lemma 3.3. *The MSL algorithm is correct and complete: given $\mathcal{H}^2$, it finds all and only the $\mathcal{G}^1$ such that $\mathcal{G}^2 = \mathcal{H}^2$.*

Proof : The correctness of the algorithm follows immediately from the fact that every $\mathcal{G}^1$ in the output set was directly tested to determine whether $\mathcal{G}^2 = \mathcal{H}^2$. For completeness of the algorithm, consider some arbitrary $\mathcal{G}^1$ in the actual equivalence class. Since $\mathcal{G}^2 = \mathcal{H}^2$, there must be a length-2 path in $\mathcal{G}^1$ for every edge in $\mathcal{H}^2$. If there is only one such path for each $\mathcal{H}^2$ edge, then that $\mathcal{G}^1$ must eventually be formed through virtual node identifiation. Suppose there are multiple such paths for some $\mathcal{H}^2$ edge, but they are not formable through different virtual node identifiations, nor is this $\mathcal{G}^1$ a supergraph of some other member of the actual equivalence class. That implies that no virtual node identification leads to *any* of those length-2 paths being created, else this $\mathcal{G}^1$ would be a supergraph of some other member of the equivalence class. But there must be *some* virtual node identification for the $\mathcal{H}^2$ edge, since the algorithm ultimately identifies every virtual node, and there is one for that edge. Hence, we have a contradiction, and so the algorithm is complete. ∎

Lemma 3.4. *A virtual node V in $S \xrightarrow{V} E$ cannot be identified with node X if any of the following holds:*

1. $\exists W \in ch_{\mathcal{G}^1}(S) \setminus X$ *s.t.* $\nexists W \leftrightarrow X$ *in* $\mathcal{H}^2$
2. $\exists W \in ch_{\mathcal{G}^1}(X) \setminus E$ *s.t.* $\nexists W \leftrightarrow E$ *in* $\mathcal{H}^2$
3. $\exists W \in ch_{\mathcal{G}^1}(X)$ *s.t.* $\nexists S \rightarrow W$ *in* $\mathcal{H}^2$
4. $\exists W \in ch_{\mathcal{G}^1}(E)$ *s.t.* $\nexists X \rightarrow W$ *in* $\mathcal{H}^2$
5. $\exists W \in pa_{\mathcal{G}^1}(S)$ *s.t.* $\nexists W \rightarrow X$ *in* $\mathcal{H}^2$
6. $\exists W \in pa_{\mathcal{G}^1}(X)$ *s.t.* $\nexists W \rightarrow E$ *in* $\mathcal{H}^2$

Proof : V being identified with X just means that we add $S \rightarrow X$ and $X \rightarrow E$ to $\mathcal{G}^1$. Each constraint is now proven separately:

1. For every non-X child W of S, $\mathcal{G}^1$ now contains $W \leftarrow S \rightarrow X$, so $\mathcal{G}^2$ will contain $W \leftrightarrow X$. If that is not in $\mathcal{H}^2$, then we have a conflict.
2. For every non-E child W of X, $\mathcal{G}^1$ now contains $W \leftarrow X \rightarrow E$, so $\mathcal{G}^2$ will contain $W \leftrightarrow E$. If that is not in $\mathcal{H}^2$, then we have a conflict.
3. For every child W of X, $\mathcal{G}^1$ now contains $S \rightarrow X \rightarrow W$, so $\mathcal{G}^2$ will contain $S \rightarrow W$. If that is not in $\mathcal{H}^2$, then we have a conflict.
4. For every child W of E, $\mathcal{G}^1$ now contains $X \rightarrow E \rightarrow W$, so $\mathcal{G}^2$ will contain $X \rightarrow W$. If that is not in $\mathcal{H}^2$, then we have a conflict.
5. For every parent W of S, $\mathcal{G}^1$ now contains $W \rightarrow S \rightarrow X$, so $\mathcal{G}^2$ will contain $W \rightarrow X$. If that is not in $\mathcal{H}^2$, then we have a conflict.
6. For every parent W of X, $\mathcal{G}^1$ now contains $W \rightarrow X \rightarrow E$, so $\mathcal{G}^2$ will contain $W \rightarrow E$. If that is not in $\mathcal{H}^2$, then we have a conflict.

∎

Lemma 3.5. *A virtual node pair V_1, V_2 for a fork $E_1 \xleftarrow{V_1} S \xrightarrow{V_2} E_2$ cannot be identified with nodes X_1, X_2 if any of the following holds:*

1. V_1 *in* $E_1 \xleftarrow{V_1} S$ *cannot be identified with* X_1
2. V_2 *in* $S \xrightarrow{V_2} E_2$ *cannot be identified with* X_2
3. $V_1 \equiv E_2 \wedge V_2 \notin pa_{\mathcal{H}^2}(E_1)$

4. $V_2 \equiv E_1 \wedge V_1 \notin pa_{\mathcal{H}^2}(E_2)$
5. $S \equiv V_2 \wedge V_1 \neq V_2 \wedge V_1 \neq E_2$ *and* $\nexists E_2 \leftrightarrow V_1$ *in* $\mathcal{H}^2$
6. $S \equiv V_1 \wedge V_1 \neq V_2 \wedge V_2 \neq E_1$ *and* $\nexists V_2 \leftrightarrow E_1$ *in* $\mathcal{H}^2$
7. $S \equiv V_1 \wedge (V_1 \equiv V_2 \vee V_2 \equiv E_2)$ *and* $\nexists E_1 \leftrightarrow E_2$ *in* $\mathcal{H}^2$
8. $S \equiv V_2 \wedge (V_1 \equiv V_2 \vee V_1 \equiv E_1)$ *and* $\nexists E_1 \leftrightarrow E_2$ *in* $\mathcal{H}^2$
9. $V_1 \equiv V_2$ *and* $\nexists E_1 \leftrightarrow E_2$ *in* $\mathcal{H}^2$

Proof : V_1, V_2 being identified with X_1, X_2 just means that we add $E_1 \leftarrow X_1$, $X_1 \leftarrow S$, $S \rightarrow X_2$, and $X_2 \rightarrow E_2$ to $\mathcal{G}^1$. Each constraint is now proven separately:

1. If V_1 cannot be identified with X_1 on other grounds, then this constraint fails
2. If V_2 cannot be identified with X_2 on other grounds, then this constraint fails
3. The first conjunct implies $V_2 \rightarrow E_2 \equiv V_1 \rightarrow E_1$ in $\mathcal{G}^1$, so $V_2 \rightarrow E_1$ in $\mathcal{G}^2$, which conflicts with the second conjunct.
4. The first conjunct implies $V_1 \rightarrow E_1 \equiv V_2 \rightarrow E_2$ in $\mathcal{G}^1$, so $V_1 \rightarrow E_2$ in $\mathcal{G}^2$, which conflicts with the second conjunct.
5. The initial conjuncts imply $V_1 \leftarrow S \equiv V_2 \rightarrow E_2$ in $\mathcal{G}^1$, so $V_1 \leftrightarrow E_2$ in $\mathcal{G}^2$, which conflicts with the last conjunct
6. The initial conjuncts imply $E_1 \leftarrow V_1 \equiv S \rightarrow V_2$ in $\mathcal{G}^1$, so $E_1 \leftrightarrow V_2$ in $\mathcal{G}^2$, which conflicts with the last conjunct
7. The initial conjuncts imply $E_1 \leftarrow V_1 \equiv S$ and either $S \rightarrow V_2 \equiv E_2$ or $S \equiv V_2 \rightarrow E_2$ in $\mathcal{G}^1$, so $E_1 \leftrightarrow E_2$ in $\mathcal{G}^2$, which conflicts with the last conjunct
8. The initial conjuncts imply $S \equiv V_2 \rightarrow E_2$ and either $E_1 \equiv V_1 \leftarrow S$ or $E_1 \leftarrow V_1 \equiv S$ in $\mathcal{G}^1$, so $E_1 \leftrightarrow E_2$ in $\mathcal{G}^2$, which conflicts with the last conjunct
9. The initial conjunct implies $E_1 \leftarrow V_1 \equiv V_2 \rightarrow E_2$ in $\mathcal{G}^1$, so $E_1 \leftrightarrow E_2$ in $\mathcal{G}^2$, which conflicts with the second conjunct.

■

Lemma 3.6. *A virtual node pair V_1, V_2 for two-edge sequence $S \xrightarrow{V_1} M \xrightarrow{V_2} E$ cannot be identified with X_1, X_2 if any of the following holds:*

1. *V_1 in $S \xrightarrow{V_1} M$ cannot be merged to X_1*
2. *V_2 in $M \xrightarrow{V_2} E$ cannot be merged to X_2*
3. $V_1 \equiv V_2 \wedge (M \notin pa_{\mathcal{H}^2}(M) \vee V_1 \notin pa_{\mathcal{H}^2}(V_2) \vee S \notin pa_{\mathcal{H}^2}(E))$

Proof : V_1, V_2 being identified with X_1, X_2 just means that we add $S \rightarrow X_1$, $X_1 \rightarrow M$, $M \rightarrow X_2$, and $X_2 \rightarrow E$ to $\mathcal{G}^1$. Each constraint is now proven separately:

1. If V_1 cannot be identified with X_1 on other grounds, then this constraint fails
2. If V_2 cannot be identified with X_2 on other grounds, then this constraint fails
3. Let $V \equiv V_1 \equiv V_2$. The first conjunct implies $S \rightarrow V \rightarrow M \rightarrow V \rightarrow E$ in $\mathcal{G}^1$, so M and V both have self-loops in $\mathcal{G}^2$, and $S \rightarrow E$ in $\mathcal{G}^2$, each of which conflicts with part of the second conjunct.

■

2 Supergraphs in equivalence class algorithm

Algorithm 1: Find class-equivalent supergraphs of $\mathcal{G}^1$

```
Input: G^1
from G^1 compute G^2
initialize an empty set S
begin add edges G*, L, G^2
    if L has elements then
        forall the edges in L do
            if edge creates a conflict remove it from L
        if len(L) == 0 then
            stop
        forall the edges in L do
            add the edge to G* add the new G* to S
            recurse into add edges with new G* and L
            remove the edge to G*
put all edges missing from G^1 into list L
call add edges G^1, L, G^2
return S
```