A PROOF OF THEOREM 3.8

To provide the proof, we first show some definitions and prepositions from [34]:

Definition A.1 (Partial Order). Let S and L be two sequences. Then, if for some k, $L^k = S$ we say that S is a pre-sequence of L and write $S \sqsubseteq L$. If $k < n$, we say that S is a proper pre-sequence of L .

Then we have the Prepositions 3.5, 3.9, 4.1 from [34] as following:

PREPOSITION 1. *A function $F(S)$ on a sequence S can be computed using a non-blocking operator, if and only if F is monotonic with respect to the partial ordering $\sqsubseteq$ defined in Definition A.1.*

PREPOSITION 2. *Let F be a function that is valid on both sequences and sets. Then, if F is a function on sequences that is monotonic with respect to the pre-sequence partial order $\sqsubseteq$ in Definition A.1, then the set function F is monotonic with respect to set containment.*

PREPOSITION 3. *A function F on relations can be computed using a non-blocking operator if and only if F is monotonic with respect to the set containment ordering.*

Based on such prepositions, we can have the following proofs: Given operator G and $i < j$, we have $G^i(S) \subseteq G^j(S)$ based on the accumulative definition. If G is non-blocking operator, we then have $G^i(S) = G(S^i)$ and $G^j(S) = G(S^j)$, then we have $G(S^i) \subseteq G(S^j)$. If we have $G(S^i) \subseteq G(S^j)$ for any i, j, then we $G^i(S^i) \subseteq G^j(S^j)$. Thus, the operator G at step $i + 1$ returns all the tuples that are contained in $G^{i+1}(S^{i+1})$ but were not in $G^i(S^i)$, then G is non-blocking.

B PROOF OF THEOREM 3.10

Given a stream S, where the recursion operator G does not contain aggregates, we want to show G is monotonic w.r.t. the set containment ordering $\subseteq$. Before step i, the operator G already seen the tuples in S^{i-1} and the recursion must reach the fixpoint which output $G^{i-1}(S^{i-1})$. Similarity, after the step i, the operator comes the output $G^i(S^i)$. Thus, we could obtain that $G^{i-1}(S^{i-1}) \subseteq G^i(S^i)$ since $S^{i-1} \subseteq S^i$.

Based on the fixpoint semantics, we could assert that $G^{i-1}(S^{i-1}) = G(S^{i-1})$ and $G^i(S^i) = G(S^i)$ since the fixpoint only depends on the input rather than the calculation orders. Then, we have $G(S^{i-1}) \subseteq G(S^i)$. Thus, $G(S^i) \subseteq G(S^j)$ is established by mathematics induction. Based on the theorem 3.8, we know the recursion operator is non-blocking.

C THE PREM PROPERTY

To address the problems raised above, the Pre-Mappability(PREM) property [61] provides formal semantics for pushing extrema aggregates, i.e. `max` and `min`, into recursion while preserving the semantics of the original stratified program. As shown in Definition C.1, its definition is based on viewing a Datalog program as function $T(R)$ where T is a relational algebra expression, and R is the vector of relations used in the expression.

Definition C.1 (PREM). Given a function $T(R_1, \dots R_k)$ defined by relational algebra and a constraint γ, γ is said to be Pre-Mappable to T if the following property holds:

$$\gamma(T(R_1, \dots, R_k)) = \gamma(T(\gamma(R_1), \dots, \gamma(R_k))).$$

For instance, if T denotes the union operator, and γ denotes the `min` or `max` constraint, we can pre-map (i.e., push) γ to the relations taking part in the union. The PREM property that has proven so useful in parallel and distributed data processing of extrema, is also critical in resolving the non-monotonic conundrum created by their presence in recursion.

D PROOF OF THEOREM 3.11

The recursion operator G with aggregation satisfying the PreM properties is also a non-blocking operator.

PROOF. We consider the operator G consists of two part : a recursion operator T without aggregation and innner aggregate operation γ. Based on the Theorem 3.10, we already know that the recursive operator T following the non-blocking properties. During each window, the aggregation γ can be moved out of T based on the PreM properties, i.e. $\gamma(T(\gamma)) = \gamma(T)$. For the stream not expiring tuples, the aggregation γ out of the T can be further interpreted as the monotonic aggregation on the stream, thus is also a non-blocking operator. Moreover, the expiration of tuples on the streams can be considered as the negation, which therefore following the PCWA assumption and also won't influence the non-blocking properties. Thus, the recursion operator G with aggregation satisfying the PreM properties is also a non-blocking operator. □