Appendix A SSP processing based recursive computation with $\mathscr{P}reM$

Definition 3. *(γ-Cover). Let P be a positive recursive Datalog program with T as its corresponding ICO. Let a constraint γ be defined over the recursive predicate on a set of k group-by arguments, denoted by $G_1, G_2, ..., G_k$ with the cost-argument denoted as C. Let γ be also $\mathscr{P}reM$ to T and P. Let there be two sets S_1 and S_2, both of which contain tuples of the form $\{(g_1, g_2, ..., g_k, c) | g_i \in G_i \forall 1 \leq i \leq k, c \in \mathbb{R}\}$, where $\mathbb{R}$ represents the set of real numbers. Now, S_1 is defined as the γ-cover for S_2, if for every tuple $t \in S_2$, there exists only one tuple $t' \in S_1$ such that (i) $t'[G] = t[G]$ and (ii) $\gamma(t'[C], t[C]) = t'[C]$.*

It is important to note from the above definition that if S_1 is the γ-cover for S_2, then there can exist a tuple $t \in S_1$, such that $t[G] \neq t'[G] \; \forall t' \in S_2$ but the converse is never true.

Lemma 2. Let P be a recursive Datalog program, T be its corresponding ICO and let the constraint γ be $\mathscr{P}reM$ to T and P, resulting in the constrained ICO T_γ. Now, for any pair of positive integers m, n, where $m \geq n$, $T_\gamma^{\uparrow m}(\emptyset)$ is a γ-cover for $T_\gamma^{\uparrow n}(\emptyset)$.

Proof. This directly follows from the fact that any atom in $T_\gamma^{\uparrow n}(\emptyset)$ with cost c can only exist in $T_\gamma^{\uparrow m}(\emptyset)$ with updated cost c', if $c = c'$ or $\gamma(c, c') = c'$. Note if $c = c'$, then $\gamma(c, c') = c'$ is trivially true.

Lemma 3. Let P be a recursive Datalog program with ICO T and let the constraint γ be $\mathscr{P}reM$ to T and P. Let P also have a parallel decomposable evaluation plan that can be executed over $\mathscr{W}$ workers, where Q_i is the program executed at worker i and T_i is the corresponding ICO defined over Q_i. Let γ be also $\mathscr{P}reM$ to T_i and Q_i, for $1 \leq i \leq \mathscr{W}$. After r rounds of synchronization (r rounds of synchronization in SSP model means every worker has sent at least r updates), if I_b and I_s denote the interpretation of the recursive predicate under BSP and SSP models respectively for any worker i, then I_s is a γ-cover for I_b.

Proof. In a SSP based fixpoint computation, any worker i can produce an atom in three ways:

(1) From local computation not involving any of the updates sent by other workers.
(2) From a join with a new atom or an update sent by another worker j.
(3) From both cases (1) and (2) together.

Now, consider the base case, where before the first round of synchronization (i.e., at the 0^{th} round) each worker performs only local computation, since it has not received/sent any update from/to any other worker. Since, in a SSP model, each local computation may involve multiple iterations (as shown in step (6) in Figure 3), I_s is trivially a γ-cover for I_b (from lemma 2).

We next assume this hypothesis (lemma 3) to be true for some $r \geq 0$. Under this assumption, we find that each worker i in SSP model for its fixpoint computation operates based on the information from its own I_s and from the ones sent by other workers after the r^{th} round of synchronization. And since each of this I_s involved is a γ-cover for the corresponding I_b (when compared against the BSP model), the aforementioned cases (1)-(3) will also produce a γ-cover for the $(r+1)^{th}$ synchronization round.

Hence, by principle of mathematical induction, the lemma holds for all $r \geq 0$.

Theorem 2. *Let P be a recursive Datalog program with ICO T and let the constraint γ be $\mathscr{P}reM$ to T and P. Let P have a parallel decomposable evaluation plan that can be executed over $\mathscr{W}$ workers, where Q_i is the program executed at worker i and T_i is the corresponding ICO defined over Q_i. If γ is also $\mathscr{P}reM$ to T_i and Q_i, for $1 \leq i \leq \mathscr{W}$, then:*

(i). *The SSP processing yields the same minimal fixpoint of $\gamma(T^{\uparrow\omega}(\emptyset))$, as would have been obtained with BSP processing.*

(ii). *If any worker i under BSP processing requires r rounds of synchronization, then under SSP processing i would require $\leq r$ rounds to reach the minimal fixpoint, where r rounds of synchronization in SSP model means every worker has sent at least r updates.*

Proof. Theorem 1 guarantees that the BSP evaluation of the datalog program with $\mathscr{P}reM$ will yield the minimal fixpoint of $\gamma(T^{\uparrow\omega}(\emptyset))$. Note that in the SSP evaluation, for every tuple t produced by a worker i from the program Q_i, $t \in T^{\uparrow\omega}(\emptyset)$. In other words, if I represents the final interpretation of the recursive predicate under SSP evaluation, then $I \subseteq T^{\uparrow\omega}(\emptyset)$ i.e. I is bounded. It also follows from lemma 3, that I is a γ-cover for the final interpretation of the recursive predicate under BSP evaluation i.e. I is a γ-cover for $\gamma(T^{\uparrow\omega}(\emptyset))$. Since, $\gamma(T^{\uparrow\omega}(\emptyset))$ is the least fixpoint under the γ constraint, we also get $\gamma(T^{\uparrow\omega}(\emptyset)) \subseteq I$, as atoms in $\gamma(T^{\uparrow\omega}(\emptyset))$ must have identical cost in I.

Thus, we can write the following equation based on the above discussion,

$$\gamma(T^{\uparrow\omega}(\emptyset)) \subseteq I \subseteq T^{\uparrow\omega}(\emptyset) \tag{A1}$$

Also recall, since γ is $\mathscr{P}reM$ to each T_i and Q_i, under the SSP evaluation, each worker i also applies γ in every iteration in its fixpoint computation (step (4) in Figure 3). Thus, we have,

$$I \subseteq \gamma(T^{\uparrow\omega}(\emptyset)) \tag{A2}$$

Combining equations (A1) and (A2), we get $I = \gamma(T^{\uparrow\omega}(\emptyset)$. Thus, the SSP evaluation also yields the same minimal fixpoint as the BSP model.

Since, the interpretation of the recursive predicate in the least model obtained from BSP evaluation is identical to that in the least model obtained from SSP processing, it directly follows from lemma 3, that the number of synchronization rounds required by worker i in SSP evaluation will be at most r, where r is the number of rounds i takes under BSP model.