APPENDIX

A. PROOFS

Proofs of selected theorems are to be examined at the discretion of the programme committee.

A.1 Proofs for Section 2

Determinism of TIOAs. Remember that according to semantic rules, TIOAs are always time deterministic. In order to check for determinism of discrete action transitions check for each location q and each action $a \in Act$, whether all its guards are mutually exclusive. Formally, let $G_{q,a}$ be the set of strengthened guards of all $i?$ transitions leaving q:

$$G_{q,a} = \{\varphi \wedge Inv(q') \mid \text{whenever } (q, a, \varphi, c, q') \in E\} \quad (10)$$

To guarantee determinism check for each pair $\varphi, \psi \in G_{q,a}$ the conjunction $Inv(q) \wedge \varphi \wedge \psi$ is inconsistent, and do that for all locations.

Input-enabledness of Specification Automata. In order to check that a TIOA S induces an input-enabled TIOTS $[\![S]\!]_{\text{sem}}$ decide for each location $q \in Loc^S$ and each input action $i? \in Act$ if a disjunction of guards of outgoing transitions labelled by $i?$ is entailed by $Inv(q)$. Formally, if $G_{q,i?}$ is the set of strengthened guards (see above) of all $i?$–transitions leaving q then in order to check if $i?$ is enabled in location q, check if

$$Inv(q) \;\text{ entails }\; \bigvee_{\psi \in G_{g,i?}} \psi \quad (11)$$

To check if the entire specification automaton is input-enabled just repeat the check for all location–input pair.

Proof Theorem. 1. Let us begin with defining an auxiliary function δ which chooses a delay and an output for every locally consistent state s:

$$\delta_s = \begin{cases} d & \text{for some } d \text{ such that } s \xrightarrow{d}^S s' \text{ and } \exists o!.\, s' \xrightarrow{o!}^S \\ +\infty & \text{if } \forall d \geq 0.\, s \xrightarrow{d}^S \end{cases} \quad (12)$$

Note that δ is a function, so it always gives a unique value of a delay for any state s, thus in the first case we mean that an arbitrary fixed value is chosen out of possibly uncountably many possible. It is immaterial for the proof which of the many values is chosen, though. It is important however that δ is time additive in the following sense: if $s \xrightarrow{d} s'$ and $d \leq \delta_s$ then $\delta_{s'} + d = \delta_s$. It is always possible to choose such a function δ due to time additivity of $\rightarrow^S$, and local consistency of S.

We want to show for an arbitrary locally consistent specifications S that it has an implementation. This can be shown by synthesising an implementation $P = (St^P, p_{s_0}, \Sigma^P, \rightarrow^P)$, where $St^P = \{p_s \mid s \in St^S\}$, $\Sigma^P = \Sigma^S$ with the same partitioning into inputs and outputs, and $\rightarrow^P$ is the largest transition relation generated by the following rules:

$$\frac{s \xrightarrow{i?}^S s' \qquad i? \in \Sigma_{\text{i}}^S}{p_s \xrightarrow{i?}^P p_{s'}} \quad (13)$$

$$\frac{s \xrightarrow{o!}^S s' \qquad o! \in \Sigma_{\text{o}}^S \qquad \delta_s = 0}{p_s \xrightarrow{o!}^P p_{s'}} \quad (14)$$

$$\frac{s \xrightarrow{d}^S s' \qquad d \in \mathcal{R}_{\geq 0} \qquad d \leq \delta_s}{p_s \xrightarrow{d}^P p_{s'}} \quad (15)$$

Since P only takes a subset of transitions of S, the determinism of S implies determinism of P. The transition relation of P is time-additive due to time additivity of $\rightarrow^S$ and of δ. It is also time-reflexive due to the last rule ($0 \leq \delta_s$ for every state s and $\rightarrow^S$ was time reflexive). So P is a TIOTS.

The new transition relation is also input enabled as it inherits of input transitions from S which was input enabled. So P is a specification. The second rule guarantees that outputs are urgent (P only outputs when no further delays are possible). Moreover P observes independent progress. Consider a state p_s. Then if $\delta_s = +\infty$ clearly p_s can delay indefinitely. If δ_s is finite, then by definition of δ and of P, the state p_s can delay and hen produce an output. Thus P is an implementation in the sense of Definition 4.

Now it is easy to show that the following relation $R \subseteq St^P \times St^S$ witnesses P sat S:

$$R = \left\{(p_s, s) \mid p_s \in St^P \text{ and } s \in St^S\right\} \quad (16)$$

This is done using an unsurprising coinductive argument. □

Proof Theorem 2. Observe first that S is already locally consistent, so all its states warrant independent progress. We only need to argue that it satisfies output urgency.

Without loss of generality, assume that S only contains states which are reachable by (sequences of) discrete or timed transitions.

If S only contains reachable states, every state of S has to be related to some state of P in a relation R witnessing $S \leq P$ (output and delay transitions need to be matched in the refinement; input transitions also need to be matched as P is input enabled and S is deterministic). This can be argued for using a standard, though slightly lengthy argument, by formalizing reachable states as a fixpoint of a monotonic operator.

Now that we know that every state of S is related to some state of P consider an arbitrary $s \in St^S$ and let $p \in St^S$ be such that $(s, p) \in R$. Then if $s \xrightarrow{o!}^S s'$ for some state $s' \in St^S$ and an output $o! \in \Sigma_{\text{o}}^S$, it must be that also $p \xrightarrow{o!} p'$ for some state $p' \in St^P$ (and $(s', p') \in R$). But since P is an implementation, its outputs must be urgent, so $p \not\xrightarrow{d}^P$ for all $d > 0$, and consequently $s \not\xrightarrow{d}^S$ for all $s > 0$. We have shown that all states of S have urgent outputs (if any) and thus S is an implementation. □

Checking Implementation Automaton Axioms. Take a specification automaton P. We will formalize the conditions under which P is an implementation automaton (i.e. $[\![P]\!]_{\text{sem}}$ is an implementation).

First we need to check for output urgency. This requires that if an output transition is enabled the source location invariant must not allow any delay, so for each outgoing transition the transition guard conjoined with this invariant should describe a single valuation of clocks (not a zone). Formally for each output transition $(q, o!, \varphi, c, q')$ it must be that $Inv(q) \wedge \varphi \wedge Inv(q')$ has a unique solution.

We have described in Section 2 how to check for independent progress. Both checks need to be done for all location–transition pairs. □

We split the proof of Theorem 3 into two lemmas.

LEMMA 1 (SOUNDNESS). *Whenever $S \leq T$ and S, T locally consistent then also $[\![S]\!]_{\text{mod}} \subseteq [\![T]\!]_{\text{mod}}$*

PROOF. Assume existence of relations R_1 and R_2 witnessing satisfaction of S by P and refinement of T by S respectively. Use a standard co-inductive argument to show that

$$R = \{(p,t) \in St^P \times St^T \mid \exists\, s \in St^S.\, (p,s) \in R_1 \wedge (s,t) \in R_2\} \tag{17}$$

is a relation witnessing satisfaction of T by P. Also observe that $(p_0, t_0) \in R$. □

LEMMA 2 (COMPLETENESS). *Whenever $[\![S]\!]_{\text{mod}} \subseteq [\![T]\!]_{\text{mod}}$ and S, T locally consistent then $S \leq T$.*

PROOF LEMMA 2. Assume that $[\![S]\!]_{\text{mod}} \subseteq [\![T]\!]_{\text{mod}}$. In the following we write $p \;\mathsf{sat}\; s$ for states p and s meaning that there exists a relation R' witnessing $P \;\mathsf{sat}\; S$ that contains (p, s).

We construct a binary relation $R \subseteq St^S \times St^T$:

$$R = \{(s,t) \mid \forall P.\, (p_0 \;\mathsf{sat}\; s \implies p_0 \;\mathsf{sat}\; t)\}\ , \tag{18}$$

where p_0 is the initial state of P. We shall argue that R witnesses $S \leq T$. Consider a pair $(s,t) \in R$. There are two cases to be considered:

- For any input $i?$ there exists $t' \in St^T$ such that $t \xrightarrow{i?}{}^T t'$. We need to show existence of a state $s' \in St^S$ such that $s \xrightarrow{i?}{}^S s'$ and $(s', t') \in R$.

  Observe that due to input-enabledness, for the same $i?$ there exists a state $s' \in St^S$ such that $s \xrightarrow{i?}{}^S s'$. We need to show that $(s', t') \in R$. By Theorem 1 we have that there exists an implementation P and its state $p_0 \in St^P$ such that $p_0 \;\mathsf{sat}\; s'$ (technically speaking s may not be an initial state of S, but we can consider a version of S with initial state changed to s to apply Theorem 1, concluding existence of an implementation).

  Consider an arbitrary implementation $Q \;\mathsf{sat}\; S$ and its state $q_0 \in St^Q$ such that $q_0 \;\mathsf{sat}\; s'$. We need to show that also $q_0 \;\mathsf{sat}\; t'$.

  Create an implementation Q' by merging Q and P above and adding a fresh state q with transition $q \xrightarrow{i?}{}^{Q'} q_0$ and transitions $q \xrightarrow{j?}{}^{Q'} p_0$ for all $j? \neq i?$, $j? \in \Sigma_i$ if $p \xrightarrow{j?}{}^P p_0$. Now $q \;\mathsf{sat}\; s$ as $p \;\mathsf{sat}\; s$ and it follows all evolutions of p for $j? \neq i?$ and $q \xrightarrow{i?}{}^{Q'} q_0$ and $q_0 \;\mathsf{sat}\; s'$. By assumption, every implementation of s is also an implementation of t, so $q \;\mathsf{sat}\; t$ and consequently $q_0 \;\mathsf{sat}\; t'$ as q is deterministic on $i?$.

  Summarizing, for any implementation $q_0 \;\mathsf{sat}\; s'$ we were able to argue that $q_0 \;\mathsf{sat}\; t'$, thus necessarily $(s', t') \in R$.

- Consider any action a (which is an output or a delay) for which exists s' such that $s \xrightarrow{a}{}^S s'$. Using a construction similar to the above it is not hard to see that one can actually construct (and thus postulate existence) of an implementation P containing $p \in St^P$ such that $p \;\mathsf{sat}\; s$ which has a transition $p \xrightarrow{a}{}^P p'$. Since then also $p \;\mathsf{sat}\; t$ we have that there exists $t' \in St^T$ such that $t \xrightarrow{a}{}^T t'$. It remains to argue that $(s', t') \in R$. This is done in the same way as with the first case, by considering any model of s', then by extending it deterministically to a model of s, concluding that it is now a model of t and the only a-derivative, which is p', must be a model of t'. Consequently $(s', t') \in R$.

It follows directly from the definition of R with $[\![S]\!]_{\text{sem}} \subseteq [\![T]\!]_{\text{sem}}$ that $(s_0, t_0) \in R$ □

A.2 Proofs for Section 3

Before we prove Theorem 4 let us first formalize the Θ operator:

$$\begin{aligned}\Theta(X) = \overline{\mathsf{err}^S} \cap \{s \in St^S \mid \forall d \geq 0.\\ [\forall\, s' \in St^S.\, s \xrightarrow{d} s' \text{ implies } s' \in X \wedge \forall\, i? \in \Sigma_{\text{i}}.\, \exists\, s'' \in X.\, s' \xrightarrow{i?} s'']\\ \vee\ [\exists\, d' \leq d.\, \exists\, s', s'' \in X.\, \exists o! \in \Sigma_{\text{o}}.\, s \xrightarrow{d'} s' \wedge s' \xrightarrow{o!} s'']\}\end{aligned} \tag{19}$$

The Θ operator formalizes a two player game, when both players choose a delay, possibly zero, and then a move to be made. The move with a shorter delay is executed. If the two delays are equal then the move is nondeterministic, and thus the operator computing the strategy requires that in either of the moves has to be non-loosing.

Θ is a monotonic operator on a complete lattice, which means that it has a greatest fixpoint, which precisely characterizes the set of consistent states in S: $\mathsf{cons}^S = \Theta(\mathsf{cons}^S)$.

PROOF THEOREM 4. First, assume that $s_0 \in \mathsf{cons}^S$. Show that S is consistent in the sense of Definition 7. In a similar fashion to the proof of Theorem 1 we first postulate existence of a function δ, which chooses a delay and an output for every consistent state s:

$$\delta_s = \begin{cases} d' & \text{if } \exists s', s'' \in \mathsf{cons}^S.\ s \xrightarrow{d'}{}^S s' \text{ and } \exists o!.\, s' \xrightarrow{o!}{}^S s'' \\ +\infty & \textit{otherwise} \end{cases} \tag{20}$$

Note that δ is a function, so it always gives a unique value of a delay for any state s, thus in the first case we mean that an arbitrary fixed value is chosen out of possibly uncountably many d's possible. It is important however that δ is time additive in the following sense: if $s \xrightarrow{d} s'$ and $d \leq \delta_s$ then $\delta_{s'} + d = \delta_s$. It is always possible to choose such a function δ due to time additivity of $\rightarrow^S$, and the fact that cons^S is a fixpoint of Θ.

We show this by constructing an implementation $P = (St^P, p_0, \Sigma^P, \rightarrow^P)$ such that $St^P = \{p_s \mid s \in St^S\}$, $\Sigma^P = \Sigma^S$ with the same partitioning in the inputs and outputs, $p_0 = p_{s_0}$ and the transition relation is the largest relation generated by the following rules:

1. $p_s \xrightarrow{o!}{}^P p_{s'}$ iff $s \xrightarrow{o!}{}^S s'$ and $s' \in \mathsf{cons}^S$ and $\delta_s = 0$
2. $p_s \xrightarrow{i?}{}^P p_{s'}$ iff $s \xrightarrow{i?}{}^S s'$
3. $p_s \xrightarrow{d}{}^P p_{s'}$ iff $s \xrightarrow{d}{}^S s'$ and $d \leq \delta_s$

Observe that the construction of P is essentially identical to the one in the proof of Theorem 1 above. It can be

argued in almost the same way as in the above prove, that P satisfies the axioms of TIOAs and is an implementation. Here one has to use the definition of Θ in order to see that the side condition in the first rule, that is $s' \in \mathsf{cons}^S$, does not introduce a violation of independent progress.

It remains to argue that P sat S. This is done by arguing that the following relation R:

$$R = \left\{(p, s) \in St^P \times St^S \mid p_s = p\right\} \tag{21}$$

witnesses the refinement of S by P.

Consider now the other direction. Assume that S is consistent and show that $s_0 \in \mathsf{cons}^S$. In the following we write that a state s is consistent, meaning that a specification would be consistent if s was the initial state.

Let $X = \{s \in St^S \mid s \text{ is consistent }\}$. It suffices to show that X is a post-fixed point of Θ, thus $X \subseteq \Theta(X)$ (then $s_0 \in X = \mathsf{cons}^S$).

Since s is consistent, let us consider an implementation P and a state p such that p sat s. We will show that $s \in \Theta(X)$. Consider an arbitrary $d \geq 0$ and the first disjunct in the definition of Θ. If $p \xrightarrow{d} p^d$ then also $s \xrightarrow{d} s^d$ and p^d sat s^d, so $s^d \in X$. Consider an arbitrary input $i?$ such that $s^d \xrightarrow{i?} s'$. Then also $p^d \xrightarrow{i?} p'$ and p' sat s' (by satisfaction). But then $s' \in X$. So by the first disjunct of definition of Θ we have that $s \in \Theta(X)$.

If $p \not\xrightarrow{d}$ for our fixed value of d then by independent progress of p there exists a $d_{\max} < d$ such that $p \xrightarrow{d_{\max}} p'$ for some p' and $p' \xrightarrow{o!} p''$ for some p'' and some output $o!$. By p sat s there also exist s' and s'' such that $s \xrightarrow{d_{\max}} s'$ and $s' \xrightarrow{o!} s''$. Moreover p'' sat s'', so $s'' \in X$, which by the second disjunct in the definition of Θ implies that $s \in \Theta(X)$.

So we conclude that X is a fixpoint of Θ. Since s_0 is consistent by assumption, then $s_0 \in X \subseteq \mathsf{cons}^S$. □

Proof Thm. 6.1. We will prove that $S \wedge T$ refines S (the other refinement is entirely symmetric).

Intuitively the theorem holds because the pruning operator used to compute $(S \times T)^\Delta$ only removes output and delay transitions (which are allowed to be dropped by the refinement). It never removes input transitions.

Let $S \wedge T = (St^S \times St^T, (s_0, t_0), \Sigma, \rightarrow)$ constructed according to the definition of conjunction. We abbreviate the set of states of $S \wedge T$ as St. It is easy to see that the following relation on states of $S \wedge T$ and states of T witnesses refinement of T by $S \wedge T$:

$$R = \left\{((s_1, t), s_2) \in St \times St^S \mid s_1 = s_2\right\} \tag{22}$$

The argument is standard, and it takes into account that $St = \mathsf{cons}^{S \times T}$ is a fixpoint of Θ.

How Θ is taken into account is demonstrated in more detail in the proof for Theorem 6.2. □

In order to prove Theorem 6.2 we will need the following lemma:

Lemma 3. *For two specifications S, T, and their states s, respectively t, if there exists an implementation P and its state p such that simultaneously p sat s and p sat t then $(s, t) \in \mathsf{cons}^{S \times T}$.*

Proof of Lemma 3. This is shown by arguing that the following set X of states of $S \times T$ is a postfixed point of Θ (then $(s, t) \in X \subseteq \Theta(X) \subseteq \mathsf{cons}^{S \times T}$):

$$X = \{(s, t) \mid \exists P.\, \exists\, p \in St^P .\, p \text{ sat } s \wedge p \text{ sat } t\} \tag{23}$$

This is done by checking that $X \subseteq \Theta(X)$. Take $(s, t) \in X$, show that $(s, t) \in \Theta(X)$. So consider an arbitrary $d_0 \geq 0$. We know that there exists state p such that p sat s and p sat t. Since p is a state of an implementation it guarantees independent progress, so there exists a delay d^p such that $p \xrightarrow{d^p} p'$ for some state p'. Now the proof is split in two cases, proceeding by coinduction:

(a) $d^p \leq d_0$ is used to show that $(s, t) \in \Theta(X)$ using a standard argument with the second disjunct in definition of Θ (namely that p can delay and output leading to a refinement of successors of s and t, which again will be in X).

(b) $d^p > d_0$ is used to show that $(s, t) \in \Theta(X)$ using the same kind of argument with the first disjunct in the definition of Θ (namely that then p can delay d_0 time and by refinement for any input transition it can advanced to a state refining successors of s and t, which are in X). □

Proof Thm. 6.2. Assume that $U \leq S$ and $U \leq T$. Then $U \leq S \wedge T$. The first refinement is witnessed by some relation R_1, the second refinement by R_2. Then the third refinement is witnessed by the following relation $R \subseteq St^U \times St^{S \wedge T}$:

$$R = \left\{(u, (s, t)) \in St^U \times \mathsf{cons}^{S \times T} \mid (u, s) \in R_1 \wedge (u, t) \in R_2\right\}$$

The argument that R is a refinement is standard again, relying on the fact that $\mathsf{cons}^{S \times T}$ is a fixed point of Θ.

Consider an output case when $u \xrightarrow{o!}{}^U u'$ for some output $o!$ and the target state u'. Then $s \xrightarrow{o!}{}^S s'$ and $t \xrightarrow{o!}{}^T t'$ for some states s' and t' and $(u's') \in R_1$ and $(u', t') \in R_2$. This means that $(s, t) \xrightarrow{o!}{}^{S \times T} (s', t')$. In order to finish the case we need to argue that $(s', t') \in St^{S \wedge T} = \mathsf{cons}^{S \times T}$. This follows from Lemma 3 since U, and thus u', is locally consistent, and by transitivity any implementation satisfying u' would be a common implementation of s' and t'.

The case for delay is identical, while the case for inputs is unsurprising (since pruning in the computation of conjunction never removes input transitions from consistent to inconsistent states – there are no such transitions). □

Theorem 6.3 follows directly from parts 1 and 2 of the same theorem.

Proof Thm. 6.4. Again this follows easily from the facts shown above. Take $L = (S \wedge T) \wedge U$ and $R = S \wedge (T \wedge U)$. Then Then $L \leq U$ and $L \leq S \wedge T$, and consequently $L \leq U$ and $L \leq S$ and $L \leq T$. Now $L \leq T$ and $L \leq U$ implies $L \leq T \wedge U$, which with $L \leq S$ gives $L \leq S \wedge (T \wedge U)$. We have shown that $L \leq R$ which gives $[\![L]\!]_{\text{mod}} \subseteq [\![R]\!]_{\text{mod}}$. The argument for $[\![R]\!]_{\text{mod}} \subseteq [\![L]\!]_{\text{mod}}$ is entirely symmetric. □

A.3 Quotient

Lemma 4. *The prequotient $T \diagdown S$ is input enabled.*

PROOF SKETCH. Inputs of $T \diagdown S$ are $\Sigma_i = \Sigma_i^T \cup \Sigma_o^S$. The universal state u (respectively the inconsistent state e) is input-enabled for Σ_i due to the [universal] (resp. [inconsistent]) rule. For the remaining states input enabledness follows from the remaining rules. Let $a \in \Sigma_i$. For $a \in \Sigma_o^S$ we get that the transition exists by the [unreachable], [unsafe], or [all] rule. Otherwise, if $a \in \Sigma_i^T$ a transition is induced by the [dividend], or [all] rule. □

We now give the proof for Theorem 12. First observe that if X has same input and output alphabets as $T \diagdown S$, then Σ_o^X and Σ_o^S are disjoint and thus $S|X$ is defined. We split the argument for the two directions of the equivalence into two separate lemmas below.

LEMMA 5. *For any two specifications S and T such that $T \diagdown S$ is defined and an implementation X over the same alphabet as $T \diagdown S$:*

$$S|X \leq T \text{ implies } X \leq T \diagdown S$$

PROOF OF LEMMA 5. We have the refinement relation R_1 showing that $S|X \leq T$ and want to give a relation showing that $X \leq T \diagdown S$. We propose the following relation and prove that it is a refinement demonstrating $X \leq T \diagdown S$:

$$R_2 = \{(x, (t \diagdown s)) \mid s|x \leq t\} \cup \{(x, \mathsf{u}) \mid x \in St^X\}$$

We have to prove that R_2 is a refinement relation. We have three cases based on the three rules in the refinement definition:

Case 1: We have $(t \diagdown s) \xrightarrow{i?} (t \diagdown s)'$ and we need to show that $x \xrightarrow{i?} x'$ and $x' \leq (t \diagdown s)'$.

We have four subcases based on which rule was used to conclude that $(t \diagdown s) \xrightarrow{i?} (t \diagdown s)'$.

- **Case 1.1** Rule [output-to-input]. We have both $t \xrightarrow{i!} t'$ and $s \xrightarrow{i!} s'$ by the rule [output-to-input]. Because x is input enabled we have $x \xrightarrow{i?} x'$ and by rule [sync-io] we have $(s|x) \xrightarrow{i!} (s'|x')$ and thus $s'|x' \leq t'$ and from this we can conclude that $x' \leq (t', s')$.
- **Case 1.2** Rule [input]. We have both $t \xrightarrow{i?} t'$ and $s \xrightarrow{i?} s'$ by rule [input]. Because x is input enabled we have $x \xrightarrow{i?} x'$ and by rule [sync-in] we have $(s|x) \xrightarrow{i?} (s'|x')$ and thus $s'|x' \leq t'$ and from this we can conclude that $x' \leq (t', s')$.
- **Case 1.3** Rule [output-e]. We can conclude that this rule could never have been used to conclude that $(t \diagdown s) \xrightarrow{i?} (t \diagdown s)'$ because in this case we would have that $t \not\xrightarrow{i!}$ and $s \xrightarrow{i!} s'$. This ensures that the refinement on the left of the implication $(s|x \leq t)$ is impossible.
- **Case 1.4** Rule [output-u] Because x is input enabled we have $x \xrightarrow{i?} x'$ and we have that $x' \leq \mathsf{u}$.

Case 2: We have $x \xrightarrow{o!} x'$ and we need to show that $(t \diagdown s) \xrightarrow{o!} (t \diagdown s)'$ and $x' \leq (t \diagdown s)'$.

There are to ways in which $(t \diagdown s) \xrightarrow{o!} (t \diagdown s)'$ can be the case:

- **Case 2.1** Rule [dividend-output1]: In this case we need to prove $t \xrightarrow{o!} t'$. Since $x \xrightarrow{o!} x'$ we know by rule [indep-l] that $x|s \xrightarrow{o!} x'|s$ and $o \notin \Sigma_o^S \cup \Sigma_i^S$. From this we can conclude that $x'|s \leq t'$ and this exactly gives us $x', \leq t' \diagdown s$.
- **Case 2.2** Rule [dividend-output2]: In this case we need to prove $t \xrightarrow{o!} t'$. Since $x \xrightarrow{o!} x'$ and $o \in \Sigma_i^S$ we know by rule [sync-io] that $x|s \xrightarrow{o!} x'|s'$. From this we can conclude that $x'|s' \leq t'$ and this exactly gives us $x', \leq t' \diagdown s'$.

There is only one way in which $(t \diagdown s) \xrightarrow{o!} (t \diagdown s)'$ can be the case and this is due to rule [dividend-output] and thus we need to prove $t \xrightarrow{o!} t'$. Since $x \xrightarrow{o!} x'$ we know by rule [indep-l] that $x|s \xrightarrow{o!} x'|s$ and $o \notin \Sigma_o^S$. From this we can conclude that $x'|s \leq t'$ and this exactly gives us $x', \leq t' \diagdown s$.

Case 3: We have $x \xrightarrow{d} x'$ and we need to show that $(t \diagdown s) \xrightarrow{d} (t \diagdown s)'$ and $x' \leq (t \diagdown s)'$.

We have two cases:

- **Case 3.1** If $s \not\xrightarrow{d}$ then we can conclude that $(t \diagdown s) \xrightarrow{d} \mathsf{u}$ and $x' \leq \mathsf{u}$.
- **Case 3.2** If $s \xrightarrow{d} s'$ then we have that $s|x \xrightarrow{d} s'|x'$ and by $s|x \leq t$ we know that $t \xrightarrow{d} t'$ and $s'|x' \leq t'$ which gives us $x' \leq t' \diagdown s'$. From $t \xrightarrow{d} t'$ and $s \xrightarrow{d} s'$ we also have $(t \diagdown s) \xrightarrow{d} (t' \diagdown s')$

Finally we just mention that anything refines u and thus all pairs added by $\{(x, \mathsf{u}) \mid x \in St^X\}$ will continue to stay in the refinement.

This concludes the proof.

□

LEMMA 6. *For any two specifications S and T such that $T \diagdown S$ is defined and an implementation X over the same alphabet as $T \diagdown S$:*

$$S|X \leq T \Longleftarrow X \leq T \diagdown S$$

PROOF OF LEMMA 6. We have the refinement relation R_2 showing that $X \leq T \diagdown S$ and want to give a relation showing that $S|X \leq T$. We propose the following relation and prove that it is a refinement demonstrating $S|X \leq T$:

$$R_1 = \{(s|x, t) \mid x \leq t \diagdown s\}$$

We have to prove that R_1 is a refinement relation. We have three cases based on the three rules in the refinement definition:

Case 1: We have $t \xrightarrow{i?} t'$ and we need to show that $(s|x) \xrightarrow{i?} (s|x)'$ and $(s|x)' \leq t'$. Since we know that s is input enabled we have that $s \xrightarrow{i?} s'$ and by rule [input] we have that $(t \diagdown s) \xrightarrow{i?} (t' \diagdown s')$. By $x \leq t \diagdown s$ we know that $x \xrightarrow{i?} x'$ and $x' \leq t' \diagdown s'$. Since we have $s \xrightarrow{i?} s'$ and $x \xrightarrow{i?} x'$ we can also conclude that $s|x \xrightarrow{i?} s'|x'$.

Case 2: We have $s|x \xrightarrow{o!} (s|x)'$ and we need to show that $t \xrightarrow{o!} t'$ and $(s|x)' \leq t'$.

We have four cases:

- **Case 2.1:** If $o \in \Sigma_o^S \cap \Sigma_i^X$ we have $s|x \xrightarrow{o!} s'|x'$ with $s \xrightarrow{o!} s'$ and $x \xrightarrow{o?} x'$ by rule [sync-io]. In this case we have two subcases: Either $t \xrightarrow{o!} t'$ or $t \not\xrightarrow{o!}$. In the first case we have$t \xrightarrow{o!} t'$ by assumption and rule [output-to-input] can be used to conclude that $x' \leq t' \diagdown s'$ because x is input enabled. This in turn gives us $s'|x' \leq t'$. In the other case $t \not\xrightarrow{o!}$ and this leads us to conclude that $x' \leq \mathsf{e}$ which could not

have been the case in the first place, so we need not consider this case.

Case 2.2: It can never be the case that $o \in \Sigma_{\mathrm{o}}^S \setminus \Sigma_{\mathrm{i}}^X$ because we know that $o \in \Sigma_{\mathrm{o}}^T$ and thus we know that it will be in $\Sigma_{\mathrm{i}}^X = \Sigma_{\mathrm{i}}^S \cap \Sigma_{\mathrm{i}}^T \cup \Sigma_{\mathrm{o}}^S \cap \Sigma_{\mathrm{o}}^T$.

Case 2.3: If $o \in \Sigma_{\mathrm{o}}^X \cap \Sigma_{\mathrm{i}}^S$ we have $s|x \xrightarrow{o!} s'|x'$ with $s \xrightarrow{o?} s'$ and $x \xrightarrow{o!} x'$ by rule [sync-io]. In this case we can conclude from rule [dividend-output2] that $t \diagup s \xrightarrow{o!} t' \diagup s'$ which gives us $x' \leq t' \diagup s'$ which in turn gives us $s'|x' \leq t'$.

Case 2.4: If $o \in \Sigma_{\mathrm{o}}^X \setminus \Sigma_{\mathrm{i}}^S$ we have $s|x \xrightarrow{o!} s|x'$ with $x \xrightarrow{o!} x'$ by rule [indep-r]. In this case we can conclude from rule [dividend-output1] that $t \diagup s \xrightarrow{o!} t' \diagup s$ which gives us $x' \leq t' \diagup s$ which in turn gives us $s|x' \leq t'$.

Case 3: We have $s|x \xrightarrow{d} s'|x'$ and we need to show that $t \xrightarrow{d} t'$ and $s'|x' \leq t'$.

From $s|x \xrightarrow{d} s'|x'$ we have $s \xrightarrow{d} s'$ and $x \xrightarrow{d} x'$. Since we have $x \xrightarrow{d} x'$, $s \xrightarrow{d} s'$ and $x \leq t \diagup s$ we know that $t \diagup s \xrightarrow{d} t' \diagup s'$ (because only rule [delay] could have been used) and $x' \leq t' \diagup s'$. Thus we also have $t \xrightarrow{d} t'$ from the premise of rule [delay] and can conclude $s'|x' \leq t'$

This concludes the proof.

□