UNIVERSITÀ CA’ FOSCARI DI VENEZIA
Dipartimento di Informatica
Technical Report Series in Computer Science

Rapporto di Ricerca CS-2004-8
Ottobre 2004

S. Crafa, S. Rossi

A Theory of Non-Interference for the $\pi$-calculus

Dipartimento di Informatica, Università Ca’ Foscari di Venezia
Via Torino 155, 30172 Mestre–Venezia, Italy
A Theory of Non-Interference for the $\pi$-calculus

Silvia Crafa and Sabina Rossi

Dipartimento di Informatica Università Ca’ Foscari, Venezia
e-mail: {silvia,rossi}@dsi.unive.it

Abstract. We develop a theory of non-interference for a typed version of the $\pi$-calculus where types are used to assign secrecy levels to channels. We provide two equivalent characterizations of non-interference based on a typed behavioural equivalence relative to a security level $\sigma$, which captures the idea of external observers of level $\sigma$. The first characterization involves a universal quantification over all the possible active attacks, i.e., malicious processes which interact with the system possibly leaking secret information. The second definition of non-interference is expressed in terms of an unwinding condition, which deals with so-called passive attacks trying to infer confidential information just by observing the behaviour of the system. This unwinding-based characterization naturally leads to efficient methods for the verification and construction of (compositional) secure systems. Furthermore, we characterize non-interference in terms of bisimulation-like (partial) equivalence relations in the style of a stream of similar studies for other process calculi (e.g., CCS and CryptoSPa) and languages (e.g., imperative and multi-threaded languages).

1 Introduction

A central issue of multilevel security systems is the protection of sensitive data and resources from undesired access. Information flow security properties have been proposed as a means to provide strong guarantees of confidentiality of secret information. These properties impose constraints on information flow ensuring that no information can flow from a higher to a lower security level. Since Denning and Denning’s work [3], information flow analysis has been studied for various programming languages, including imperative languages [3, 17, 20], functional languages [7, 15] and concurrent languages [4, 11–14, 16, 19, 21, 2].

One of the most successful approaches to information flow security relies on the notion of Noninterference [6]. The basic idea is that a system is interference free if the low level observation of the system is independent from the behaviour of its high components. Recently, various type-based proof techniques for the $\pi$-calculus have been proposed [8, 11–14]. They usually rely on a type system together with a soundness theorem stating that if a system is well-typed, then no change in the behaviour of its high components can affect the low level view

* Supported by the EU Contract IST-2001-32617 “Models and Types for Security in Mobile Distributed Systems” (MyThS) and FIRB project RBAU018RCZ “Interpretazione astratta e model checking per la verifica di sistemi embedded”. 
of the system. In these works the notion of noninterference depends on strong typing constraints (see Section 5 for a brief overview).

In this paper we wish to define a general theory of noninterference for the $\pi$-calculus, where the use of types is much lighter. In particular, the only typing constraint we impose is that values at a given security clearance cannot flow through channels with a lower security level. Our typing discipline ensures that information does not explicitly flow from high to low. Implicit flows are not dealt with the type system and then we cannot use it as a proof technique for noninterference. On the contrary, we characterize noninterference in terms of the actions that typed processes may perform.

Our approach intends to generalize previous ideas, mainly developed for CCS, to the $\pi$-calculus, where new difficulties arise due to the presence of scope extrusion. The main contribution is the formalization of a rich and elegant theory of noninterference for the $\pi$-calculus, where a number of sound and complete characterizations of secure processes lead to efficient verification techniques.

The noninterference property we are going to study is based on the notion of process behaviour relative to a security level $\sigma$, taken from a complete lattice $\langle \Sigma, \preceq \rangle$ of security annotations. We define typed equivalences for the $\pi$-calculus relative to an observation level $\sigma$, namely $\sigma$-reduction barbed congruences [9], where the judgements have the form

$$\Gamma \vdash P \cong_{\sigma} Q$$

meaning that in the type environment $\Gamma$, the processes $P$ and $Q$ exhibit the same $\sigma$-level behaviour, i.e., they are indistinguishable for a $\sigma$-level observer.

A $\sigma$-level observer is formalized as a $\sigma$-context, i.e., a well typed context which can interact with the observed process only through channels of level at most $\sigma$. We require $\cong_{\sigma}$ to be a congruence for all $\sigma$-level contexts.

Building on [9], we develop a proof technique for $\cong_{\sigma}$ in terms of a quite natural bisimilarity on $\sigma$-actions defined on typed labelled transition systems. A typed LTS is built around typed actions of the form

$$\Gamma \triangleright P \xrightarrow{\alpha} \gamma \delta \Gamma' \triangleright P'$$

indicating that in the type environment $\Gamma$, the process $P$ performs the action $\alpha$ of level $\delta$ and evolves to $P'$ in the possibly modified environment $\Gamma'$. We prove that if two processes are bisimilar on typed actions of level $\sigma$, then they are also $\sigma$-barbed congruent. Relying on this equational theory for the $\pi$-calculus, we introduce the noninterference property $\mathcal{NI} (\cong_{\sigma})$ for typed processes, which is inspired by the $P_{BND}\Pi$DC property defined in [5] for CCS. We say that a process $P$ in a type environment $\Gamma$ satisfies the property $\mathcal{NI} (\cong_{\sigma})$, written $\Gamma \triangleright P \in \mathcal{NI} (\cong_{\sigma})$, if for every configuration $\Gamma' \triangleright P'$ reachable from $\Gamma \triangleright P$ in the typed LTS, and for every $\sigma$-high level source $H$ (that is a process which can perform only actions at level higher than $\sigma$) it holds

$$\Gamma' \triangleright P' \cong_{\sigma} \Gamma' \triangleright P' \mid H.$$
This definition involves a universal quantification over all the possible active attacks, i.e., high level malicious processes $H$ which interact with the system possibly leaking secret information. Moreover, it is persistent in the sense that if a configuration satisfies $\mathcal{NT}(\cong_\sigma)$ then also all the configurations reachable from it in the typed LTS satisfy $\mathcal{NT}(\cong_\sigma)$. As discussed in [5], persistence is technically useful since it allows us to apply inductive reasoning when proving security results (e.g., compositionality), but it is also intuitively motivated by the need for mobile processes to be secure at any computation step.

We provide a first characterization of $\mathcal{NT}(\cong_\sigma)$ in terms of an unwinding condition in the style of [1]. The unwinding condition aims at specifying local constraints on process transitions which imply the global security property. More precisely, we require that whenever a configuration $C$ performs a typed action of level higher than $\sigma$ moving to $C'$, then a configuration $C''$ can also be reached through an internal computation such that $C'$ and $C''$ are indistinguishable for a $\sigma$-level observer. In other words, the unwinding condition ensures that the $\sigma$-high actions are always simulated by internal computations, thus becoming invisible for the low level observers.

It is interesting to observe that the unwinding condition characterizes security with respect to the so-called passive attacks, which try to infer information about the classified behaviour ($\sigma$-high actions) just by observing the $\sigma$-level behaviour of the system. Thanks to this characterization, the noninterference property $\mathcal{NT}(\cong_\sigma)$ becomes decidable for finite state processes, i.e., processes whose typed LTS is finite. Furthermore, we show that $\mathcal{NT}(\cong_\sigma)$ is compositional with respect to most of the operators of the $\pi$-calculus. In particular, if $P$ and $Q$ satisfy $\mathcal{NT}(\cong_\sigma)$ then $P | Q$ and $!P$ also do.

We also develop two quantifier-free characterizations of noninterference based on bisimulation-like (partial) equivalence relations. More precisely, we first introduce a partial equivalence relation $\approx_\sigma$ (per model) over configurations and, inspired by the definitions in [17, 16] for imperative and multi-threaded languages, we prove that $\approx_\sigma$ is reflexive only on the set of secure processes. Hence, we obtain that a typed process $P$ is secure if and only if $P$ is $\approx_\sigma$-equivalent to itself. Then we investigate the impact of name restriction on noninterference. Let $(\nu^\sigma)P$ be the process $P$ where all its $\sigma$-high free names are restricted. We define the equivalence relation $\approx_\sigma$ and prove that a typed process $P$ is secure if and only if $P$ and $(\nu^\sigma)P$ are $\approx_\sigma$-equivalent (a similar definition is given in [5] for CCS). Finally we show that two well typed processes $P$ and $Q$ are equivalent on $\sigma$-actions if and only if $(\nu^\sigma)P$ and $(\nu^\sigma)Q$ are equivalent on every action. This property allows us to precisely relate the standard bisimulation equivalence $\approx$ for the $\pi$-calculus with our bisimulation on $\sigma$-actions and also to express our noninterference property in terms of the relation $\approx$ over restricted processes.

The paper is organized as follows. In Section 2 we present the language, its semantics and the type system. In Section 3 we study typed observation equivalences relative to a security level. In Section 4 we introduce the notion of $\sigma$-noninterference and provide a number of characterizations based on typed
actions. Section 5 concludes the paper discussing some related work. All the proofs are collected in two final appendixes.

2 Syntax and Semantics of typed π-calculus

In this section we introduce the language, its operational semantics and the type system with which we will be concerned.

We presuppose a countably-infinite set of names and a countably-infinite set of variables ranged over by \( n, ..., q \) and by \( x, ..., z \), respectively. We often use \( a, b, c \) to range over both names and variables. We also assume a complete lattice \( \langle \Sigma, \preceq \rangle \) of security annotations, ranged over by \( \sigma, \delta \), where \( \top \) and \( \bot \) represent the top and the bottom elements of the lattice. The syntax of processes and types is shown in Table 1. It is a synchronous, monadic, calculus with the match/mismatch operator. The choice of a synchronous and monadic calculus is just for the sake of simplifying the presentation; on the other hand, as explained in [9], the matching construct is essential for the coinductive characterization of the reduction barbed congruence shown in Section 3.

As usual, the input construct \( a(x : T).P \) acts as a binder for the variable \( x \) in \( P \), while the restriction \( (\nu n : T)P \) acts as a binder for the name \( n \) in \( P \). We identify processes up to \( \alpha \)-conversion. We use \( \text{fn}(P) \) and \( \text{fv}(P) \) to denote the set of free names and free variables, respectively, in \( P \). We write \( P[x := n] \) to denote the substitution of all free occurrences of \( x \) in \( P \) with \( n \), and we often write \( a(x:T),\pi(b) \) omitting trailing \( 0 \)'s. In this paper we restrict to closed processes, that are processes containing no free occurrences of variables; in Section 5 we discuss how to extend our theory also to open terms.

Types assign security levels to channels. More precisely, if \( \sigma \in \Sigma \), then \( \sigma[] \) is the type of channels of level \( \sigma \) which carry no values, while \( \sigma[T] \) is the type of channels of level \( \sigma \) which carry values of type \( T \). We consider the function \( A \) associating to types the corresponding level, that is \( A(\sigma[]) = \sigma = A(\sigma[T]) \).
### Semantics

The operational semantics of our language is given in terms of a labelled transition system (LTS) defined over processes. The set of labels, or actions, is the following:

- **Actions**
  - \( \pi(m) \) send a name
  - \((\nu m : T)\pi(m)\) send a fresh name
  - \(n(m)\) receive a name
  - \(\tau\) internal action

We write \(\text{fn}(\alpha)\) and \(\text{bn}(\alpha)\) to denote the set of free and bound names occurring in the action \(\alpha\), where \(\text{bn}(\alpha) = \{m\}\) if \(\alpha = (\nu m : T)\pi(m)\), and \(\text{bn}(\alpha) = \emptyset\) otherwise.

The LTS is defined in Table 2 and it is entirely standard; we just omitted the symmetric rules for (Sum), (Par), (Comm) and (Close) in which the role of the left and right components are swapped.

### Type System

Our type system corresponds to the basic type system for the \(\pi\)-calculus (see [18]). The main judgements take the form \(\Gamma \vdash P\), where \(\Gamma\) is a
type environment, that is a finite mapping from names and variables to types. Intuitively, $\Gamma \vdash P$ means that the process $P$ uses all channels as input/output devices in accordance with their types, as given in $\Gamma$. The other, auxiliary, judgements are $\Gamma \vdash a : T$ stating that the name/variable $a$ has type $T$ in $\Gamma$, and $\Gamma \vdash \cdot$ stating that the type environment $\Gamma$ is well formed. The typing rules are collected in Table 3, and they are based on the following rules of type formation, which prevent a channel of level $\delta$ from carrying values of level higher than $\delta$.

Notice that the type formation rules guarantee the absence of any explicit flow of information from a higher to a lower security level: for instance, the process $\text{pub}(\text{passwd})\cdot \mathbf{0}$ where a secret password is forwarded along a public channel, is not well-typed.

The following subject reduction property expresses the consistency between the operational semantics and the typing rules (see [18]).

**Proposition 1 (Subject Reduction).** Let $P$ be a process such that $\Gamma \vdash P$.

- if $P \xrightarrow{\tau} P'$ then $\Gamma \vdash P'$.
If $P \xrightarrow{n(m)} P'$ then $\Gamma \vdash n : \delta[T]$ and if $\Gamma \vdash m : T$ then $\Gamma \vdash P'$.

If $P \xrightarrow{\pi(m)} P'$ then $\Gamma \vdash n : \delta[T]$, $\Gamma \vdash m : T$ and $\Gamma \vdash P'$.

If $P \xrightarrow{(\nu m : T) \pi(m)} P'$ then $\Gamma \vdash n : \delta[T]$ and $\Gamma, m : T \vdash P'$.

Proof. By induction on the depth of the derivation of $P \xrightarrow{\alpha} P'$ and a case analysis on $\alpha$. See [18] for details.

3 Observational Equivalence for $\pi$-calculus

In this section we introduce the notion of $\sigma$-level observation equivalence and we develop an equational theory for the $\pi$-calculus which is parametric on the security level (i.e., the observational power) of the observers.

Our equivalences are reminiscent of the typed behavioural equivalences for $\pi$-calculus [9, 11, 18], that are equivalences indexed by a type environment $\Gamma$ ensuring that both the observed process and the observer associate the same security levels to the same names. Our equivalences, however, are much simpler than those in [9, 11] since we do not consider subtyping nor linearity/affinity.

Our type-indexed relations are based on the notion of configuration. We say that $\Gamma \triangleright P$ is a configuration if $\Gamma$ is a type environment and $P$ is a process such that $\Gamma \vdash P$. A type-indexed relation over processes is a family of binary relations between processes indexed by type environments. We write $\Gamma \models P \mathcal{R} Q$ to mean that $P$ and $Q$ are related by $\mathcal{R}$ at $\Gamma$ and $\Gamma \triangleright P$ and $\Gamma \triangleright Q$ are configurations.

To define our $\sigma$-level observation equivalences, we will ask for the largest type-indexed relation over processes which satisfies the following properties.

Reduction Closure. A type-indexed relation $\mathcal{R}$ over processes is reduction closed if $\Gamma \models P \mathcal{R} Q$ and $P \xrightarrow{\tau} P'$ imply that there exists $Q'$ such that $Q \Rightarrow Q'$ and $\Gamma \models P' \mathcal{R} Q'$, where $\Rightarrow$ denotes the reflexive and transitive closure of $\xrightarrow{\tau}$.

$\sigma$-Barb Preservation. Let $\sigma \in \Sigma$, $P$ be a process and $\Gamma$ a type environment such that $\Gamma \vdash P$. We write $\Gamma \models P \downarrow_{\sigma}^n$ if $P \xrightarrow{n(m)}$ with $\Lambda(\Gamma(n)) \leq \sigma$. Furthermore we write $\Gamma \models P \uparrow_{\sigma}^n$ if there exists some $P'$ such that $P \Rightarrow P'$ and $\Gamma \models P' \downarrow_{\sigma}^n$. A type-indexed relation $\mathcal{R}$ over processes is $\sigma$-barb preserving if $\Gamma \models P \mathcal{R} Q$ and $\Gamma \models P \downarrow_{\sigma}^n$ imply $\Gamma \models Q \uparrow_{\sigma}^n$.

$\sigma$-Contextuality. Let a typed context be a process with at most one typed hole $[\cdot : \Gamma]$. If $C[\cdot : \Gamma]$ is a typed context and $P$ is a process such that $\Gamma \vdash P$, then we write $C[P]$ for the process obtained by replacing the hole in $C[\cdot : \Gamma]$ by $P$. In order to type contexts, the type system of Table 3 is extended with the following rule:

$$
\text{(Ctx)} \quad \Gamma, \Gamma' \vdash [\cdot : \Gamma]
$$
Proposition 2. Let $\Gamma \vdash P$ and $\Gamma', \Gamma' \vdash C[\cdot]$, then $\Gamma, \Gamma' \vdash C[\cdot]$.

Proof. By induction on the derivation of $\Gamma, \Gamma' \vdash C[\cdot]$.

We are interested in $\sigma$-contexts that capture the idea of $\sigma$-level observers. Intuitively, a $\sigma$-context is an evaluation context which may interact with the process filling the hole just through channels of level at most $\sigma$.

Definition 1 ($\sigma$-context). Let $\sigma \in \Sigma$. A context $C[\cdot]$ is a $\sigma$-context if there exists a type environment $\Gamma'$ such that $\Gamma, \Gamma' \vdash C[\cdot]$ and $C[\cdot]$ is generated by the following grammar


where $P$ is a process such that $\forall n \in \text{fn}(P)$ we have $\Lambda(\Gamma, \Gamma'(n)) \preceq \sigma$.

Example 1. Let $\Gamma$ be the type environment $h : \top[\bot], \ell : \bot[\bot]$ and $\sigma \prec \top$. The context $(\nu h)(\ell(h) | [\cdot])$ is not a $\sigma$-context since the process $h(\ell)$ in parallel with the hole has a free occurrence of the high name $h$. This context does not represent a $\sigma$-level observer since it can interact with a process filling the hole through the high channel $h$.

We say that a type-indexed relation $R$ over processes is $\sigma$-contextual if $\Gamma \vdash P \sim R Q$ and $\Gamma, \Gamma' \vdash C[\cdot]$ imply $\Gamma, \Gamma' \vdash C[P] \sim R C[Q]$ for all $\sigma$-contexts $C[\cdot]$.  

Definition 2 ($\sigma$-Reduction Barbed Congruence $\cong_\sigma$). Let $\sigma \in \Sigma$. The $\sigma$-reduction barbed congruence, denoted by $\cong_\sigma$, is the largest type-indexed relation over processes which is symmetric, $\sigma$-contextual, reduction closed and $\sigma$-barb preserving.

3.1 A bisimulation-based proof technique

In this section we develop a proof technique for the equivalences $\cong_\sigma$ defined above. More precisely, following [8, 9], we define a LTS of typed actions (called typed LTS) over configurations. As in [8], actions are parameterized over security levels and take the form

$$\Gamma \triangleright P \xrightarrow{\alpha} \delta \Gamma' \triangleright P'$$

indicating that the process $P$ in the type environment $\Gamma$ can perform the action $\alpha$ to interact with some $\delta$-level observer. In this case, we say that $\alpha$ is a $\delta$-level action.

The rules of the typed LTS are obtained from those in Table 2 by taking into account the type environment $\Gamma$ which records the security levels of the channels used by the process. Differently from [8], our typed actions are built around just a single type environment $\Gamma$ constraining the observed process $P$. This differs from [8] where, due to the presence of subtyping, two distinct type environments are needed, one for the observer and the other for the observed process.
The rules of the typed LTS are reported in Table 4; note that there is an additional input action of the form \((\nu m:T)n\(m\)) occurring when the process receives a new name \(m\) generated by the environment.

A precise relationship between the untyped actions and the typed ones is established in the following proposition, whose proof is immediate.

**Proposition 3.** Let \(\Gamma \triangleright P\) be a configuration. Then

1. \(\Gamma \triangleright P \xrightarrow{\tau} Q\) if and only if \(P \xrightarrow{\tau} Q\).
2. \(\Gamma \triangleright P \xrightarrow{\alpha} P'\) with \(\alpha \in \{n\(m\), \pi\(m\)\}\) if and only if \(P \xrightarrow{\pi\(m\)} P'\), \(m \in \text{Dom}(\Gamma)\) and \(\Lambda(\Gamma(n)) \leq \delta\).
3. \(\Gamma \triangleright P \xrightarrow{\nu m:T} P'\) if and only if \(P'\) with \(\Gamma(n) = \delta_1[T]\) and \(\delta_1 \leq \delta\).
4. \(\Gamma \triangleright P \xrightarrow{n\(m\)} P'\) with \(\Gamma(n) = \delta_1[T]\), \(\delta_1 \leq \delta\) and \(m \notin \text{Dom}(\Gamma)\).

**Table 4.** Typed LTS for \(\pi\)-calculus

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Gamma \triangleright P \xrightarrow{\alpha} P')</td>
<td>(\Delta \triangleright P \xrightarrow{\alpha} P') if and only if (P \xrightarrow{\alpha} P'), (\alpha \in {\pi(m), \nu m:T}) and (\Delta(\Gamma(n)) \leq \delta).</td>
</tr>
<tr>
<td>(\Gamma \triangleright P \xrightarrow{\nu m:T} P')</td>
<td>(\Delta \triangleright P \xrightarrow{\nu m:T} P') if and only if (P') with (\Delta(\Gamma(n)) = \delta_1[T]) and (\delta_1 \leq \delta).</td>
</tr>
<tr>
<td>(\Gamma \triangleright P \xrightarrow{n(m)} P')</td>
<td>(\Delta \triangleright P \xrightarrow{n(m)} P') with (\Delta(\Gamma(n)) = \delta_1[T]), (\delta_1 \leq \delta) and (m \notin \text{Dom}(\Gamma)).</td>
</tr>
</tbody>
</table>
The next proposition shows how the type environment is modified after the execution of an action.

**Proposition 4.** Let be \( \Gamma \triangleright P \xrightarrow{\alpha} \delta \Gamma' \triangleright P' \), then

- if \( \alpha \in \{ \tau, \overline{\pi}(m), n(m) \} \) then \( \Gamma' = \Gamma \).
- if \( \alpha \in \{ (\nu m : T) \overline{\pi}(m), (\nu m : T) n(m) \} \) then \( \Gamma' = \Gamma, m : T \).

**Proof.** By induction on the depth of the derivation of \( \Gamma \triangleright P \xrightarrow{\alpha} \delta \Gamma' \triangleright P' \).

Relying on the typed LTS, we now introduce the bisimilarity on \( \sigma \)-actions which provides a coinductive characterization of \( \sigma \)-reduction barbed congruence \( \approx_{\sigma} \).

With an abuse of notation, we write \( \Rightarrow = \) for the reflexive and transitive closure of \( \xrightarrow{\tau} \). We also write \( \Rightarrow = \) for \( \Rightarrow = \) if \( \alpha = \tau \) and \( \Rightarrow = \) otherwise.

**Definition 3 (Bisimilarity on \( \sigma \)-actions \( \approx_{\sigma} \)).** Let \( \sigma \in \Sigma \). Bisimilarity on \( \sigma \)-actions is the largest symmetric relation \( \approx_{\sigma} \) over configurations, such that whenever \( (\Gamma \triangleright P) \approx_{\sigma} (\Gamma \triangleright Q) \), if \( \Gamma \triangleright P \xrightarrow{\alpha} \delta \Gamma' \triangleright P' \), then there exists \( Q' \) such that \( \Gamma \triangleright Q \xrightarrow{\alpha} \delta \Gamma' \triangleright Q' \) and \( (\Gamma' \triangleright Q') \approx_{\sigma} (\Gamma' \triangleright P') \).

In the rest of the paper, for a given relation \( \mathcal{R} \) over configurations, we write

\[
\Gamma \models P \mathcal{R} Q \quad \text{whenever} \quad (\Gamma \triangleright P) \mathcal{R} (\Gamma \triangleright Q).
\]

**Theorem 1.** Let \( \sigma \in \Sigma \), \( \Gamma \) be a type environment and \( P, Q \) be processes such that \( \Gamma \vdash P, Q \). \( \Gamma \vdash P \approx_{\sigma} Q \) if and only if \( \Gamma \vdash P \approx_{\sigma} Q \).

### 4 Information Flow in \( \pi \)-calculus

In this section we introduce a notion of noninterference for processes of the typed \( \pi \)-calculus which uses the \( \sigma \)-reduction barbed congruence \( \approx_{\sigma} \) as observation equivalence. This property, called \( \mathcal{NI}(\approx_{\sigma}) \), is inspired by the \( P_{\text{BNDC}} \) property defined in [5] for CCS processes; it requires that no information flow should occur even in the presence of active malicious processes, e.g., Trojan Horse programs, that run at the classified (higher than \( \sigma \)) level. We use the following notations:

- We say that a configuration \( \Gamma' \triangleright P' \) is **reachable** from a configuration \( \Gamma \triangleright P \), written \( \Gamma \triangleright P \leadsto \Gamma' \triangleright P' \), if \( \Gamma \triangleright P \xrightarrow{\alpha_1} \sigma_1 \xrightarrow{\alpha_2} \sigma_2 \ldots \xrightarrow{\alpha_n} \sigma_n \Gamma' \triangleright P' \) for some \( \alpha_i \) and \( \sigma_i \), for \( i = 1, \ldots, n \). (Notice that the concept of reachability is independent from the levels \( \sigma_i \).)
- Given a type environment \( \Gamma \), we say that a process \( P \) is a **\( \sigma \)-high level source** in \( \Gamma \), written \( P \in \mathcal{H}^\sigma_{\Gamma} \), if \( \Gamma \vdash P \) and \( \Gamma \triangleright P \xrightarrow{\alpha_1} \sigma_1 \ldots \xrightarrow{\alpha_n} \sigma_n \) imply \( \sigma \prec \sigma_i \), for all \( i = 1, \ldots, n \).
Given a security level \( \sigma \in \Sigma \), we write \( \Gamma \vdash P \xrightarrow{\alpha} \sigma \Gamma' \vdash P' \) if whenever \( \Gamma \vdash P \xrightarrow{\sigma} \Gamma' \vdash P' \) then \( \sigma < \delta \). In this case we say that \( \Gamma \vdash P \) has performed a \( \sigma \)-high level action.

A process \( P \) in a type environment \( \Gamma \) satisfies the property \( \mathcal{NI}(\equiv_\sigma) \) if for every configuration \( \Gamma' \vdash P' \) reachable from \( \Gamma \vdash P \) and for every \( \sigma \)-high level source \( H \), a \( \sigma \)-level user cannot distinguish, in the sense of \( \equiv_\sigma \), \( \Gamma' \vdash P' \) from \( \Gamma' \vdash P' \mid H \).

The formal definition of \( \mathcal{NI}(\equiv_\sigma) \) is as follows.

**Definition 4 (\( \sigma \)-Noninterference).** Let \( \sigma \in \Sigma \), \( P \) be a process and \( \Gamma \) be a type environment such that \( \Gamma \vdash P \). The process \( P \) satisfies the \( \sigma \)-noninterference property in \( \Gamma \), written \( \Gamma \vdash P \in \mathcal{NI}(\equiv_\sigma) \), if for all \( \Gamma' \vdash P' \) such that \( \Gamma \vdash P \xrightarrow{\sim} \Gamma' \vdash P' \) and for all \( H \in \mathcal{H}_\sigma^\prec \), it holds \( \Gamma' \vdash P' \equiv_\sigma P' \mid H \).

**Example 2.** In the following examples, we assume just two security levels: \( H \) and \( L \) with \( L \preceq H \); let also \( h \) be a high level channel and \( \ell, \ell_1, \ell_2 \) be low level channels. Let \( \Gamma \) be the type environment \( h : H[], \; \ell : L[], \; \ell_1 : L[], \; \ell_2 : L[] \) and \( \sigma = L \).

Let us first consider the following simple insecure process: \( P_1 = (h(\ell) \cdot \ell()) \mid \overline{h()} \). To show that \( \Gamma \vdash P_1 \not\in \mathcal{NI}(\equiv_\sigma) \) it is sufficient to consider the configuration \( \Gamma' \vdash P'_1 \) with \( P'_1 = h(\ell) \cdot \ell() \) that is reachable from \( \Gamma \vdash P_1 \) after performing the output action \( \overline{h()} \). The process \( P'_1 \) is clearly insecure in the type environment \( \Gamma \) since the low level, observable, action \( \ell() \) directly depends on the high level input \( h() \). Indeed, by choosing \( H = \overline{h()} \) one can easily observe that \( \Gamma \vdash P'_1 \not\equiv_\sigma P'_1 \mid H \).

Let us consider a further classic example of insecure process, that is \( P_2 = h(x : T) \cdot \text{if } x = n \text{ then } \ell_1() \text{ else } \ell_2() \) in the type environment \( \Gamma' = h : H[T], \; \ell_i : L[], \; n : T \) (here the security level of \( n \) is irrelevant). To show that \( \Gamma' \vdash P_2 \not\in \mathcal{NI}(\equiv_\sigma) \) one can choose \( H = \overline{h(n)} \), where \( H \in \mathcal{H}_\sigma^\prec \) independently on the level of \( n \), and observe that \( \Gamma' \vdash P_2 \not\equiv_\sigma P_2 \mid H \). Intuitively, when \( n \) is a high level name, a low level observer may infer from \( P_2 \) the value of the high level variable \( x \), which is clearly unsound.

Finally, consider the process \( P_3 = P_2 \mid \overline{h(n)} \mid \overline{h(m)} \), where the variable \( x \) can be nondeterministically substituted either with \( n \) or \( m \). \( P_3 \) is still an insecure process since an external attack can destroy the nondeterminism causing an interference: for instance, if \( H = h(y).h(z).\overline{h(n)} \), then \( \Gamma' \vdash P_3 \not\equiv_\sigma P_3 \mid H \).

Building on the ideas developed in [1] for a class of persistent noninterference properties for CCS processes, we provide a characterization of \( \mathcal{NI}(\equiv_\sigma) \) in terms of an unwinding condition. Intuitively, the unwinding condition specifies local constraints on the typed actions of the system which imply the global security property. More precisely, our unwinding condition ensures that no \( \sigma \)-high action \( \alpha \) leading to a configuration \( C \) is observable by a \( \sigma \)-low user, as there always exists a configuration \( C' \), \( \sigma \)-equivalent to \( C \), that the system may reach without performing \( \alpha \).

**Definition 5 (\( \sigma \)-Unwinding Condition).** Let \( \sigma \in \Sigma \), \( P \) be a process and \( \Gamma \) be a type environment such that \( \Gamma \vdash P \). The process \( P \) satisfies the \( \sigma \)-unwinding condition in \( \Gamma \), written \( \Gamma \vdash P \in \mathcal{W}(\equiv_\sigma) \), if for all \( \Gamma' \vdash P_1 \) such that \( \Gamma \vdash P \xrightarrow{\sim} \Gamma' \vdash P_1 \)
- if \( \Gamma' \triangleright P_1 \xrightarrow{\alpha} \sigma \) \( \Gamma' \triangleright P_2 \) with \( \alpha \in \{ \pi(m), n(m) \} \), then \( \exists P_3 \) such that \( \Gamma' \triangleright P_1 \implies \Gamma' \triangleright P_3 \) and \( \Gamma' \vdash P_2 \cong_\sigma P_3 \);  
- if \( \Gamma' \triangleright P_1 \xrightarrow{\alpha} \Gamma', m:T \triangleright P_2 \) with \( \alpha \in \{ (\nu m:T)\pi(m), (\nu m:T)n(m) \} \), then \( \exists P_3 \) such that \( \Gamma' \triangleright P_1 \implies \Gamma' \triangleright P_3 \) and \( \Gamma' \vdash P_3 \cong_\sigma (\nu m:T)P_2 \).

This unwinding-based schema characterizes a notion of security with respect to all passive attacks which try to infer information about the classified behavior just by observing the \( \sigma \)-level behaviour of the system.

Both properties \( NI(\cong_\sigma) \) and \( W(\cong_\sigma) \) are persistent, as stated in the following proposition, whose proof is immediate.

**Proposition 5 (Persistence).** Let \( \sigma \in \Sigma, P \) be a process and \( \Gamma \) be a type environment such that \( \Gamma \vdash P \). For all \( \Gamma' \triangleright P' \) such that \( \Gamma \triangleright P \leadsto \Gamma' \triangleright P' \) it holds

- if \( \Gamma \triangleright P \in NI(\cong_\sigma) \) then \( \Gamma' \triangleright P' \in NI(\cong_\sigma) \).
- if \( \Gamma \triangleright P \in W(\cong_\sigma) \) then \( \Gamma' \triangleright P' \in W(\cong_\sigma) \).

The equivalence of properties \( NI(\cong_\sigma) \) and \( W(\cong_\sigma) \) is stated below.

**Theorem 2.** Let \( \sigma \in \Sigma, P \) be a process and \( \Gamma \) be a type environment such that \( \Gamma \vdash P, P \in NI(\cong_\sigma) \) if and only if \( P \in W(\cong_\sigma) \).

The characterization of \( \sigma \)-noninterfering processes based on unwinding provides a better understanding of the operational semantics of secure processes. Moreover, it allows one to define efficient proof techniques for \( \sigma \)-noninterference just by inspecting the typed LTS of processes. Notice that the \( \sigma \)-unwinding condition \( W(\cong_\sigma) \) is decidable in the case of finite state processes, i.e., processes whose typed LTS is finite. Moreover, by exploiting the following compositional results, the unwinding condition \( W(\cong_\sigma) \) can be used to define methods, e.g., a proof system, both to check the security of complex systems and to incrementally build processes which are secure by construction.

**Theorem 3 (Compositionality of \( W(\cong_\sigma) \)).** Let \( \sigma \in \Sigma, P \) and \( Q \) be two processes and \( \Gamma \) be a type environment such that \( \Gamma \vdash P, Q \). If \( \Gamma \triangleright P \in W(\cong_\sigma) \) and \( \Gamma \triangleright Q \in W(\cong_\sigma) \) then

- \( \Gamma, \Gamma' \triangleright \pi(b)P \in W(\cong_\sigma) \) where \( \Gamma, \Gamma' \vdash a : \delta[T] \), \( \Gamma, \Gamma' \triangleright b : T \) and \( \delta \leq \sigma \);  
- \( \Gamma, \Gamma' \triangleright a(x : T)P \in W(\cong_\sigma) \) where \( \Gamma, \Gamma' \vdash a : \delta[T] \) and \( \delta \leq \sigma \);  
- \( \Gamma, \Gamma' \triangleright \text{if } a = b \text{ then } P \text{ else } Q \in W(\cong_\sigma) \) where \( \Gamma, \Gamma' \vdash a : T \) and \( \Gamma, \Gamma' \triangleright b : T \);  
- \( \Gamma \triangleright P \mid Q \in W(\cong_\sigma) \);  
- \( \Gamma' \triangleright (\nu n : T)P \in W(\cong_\sigma) \) where \( \Gamma = \Gamma', n : T \);  
- \( \Gamma \triangleright \Pi P \in W(\cong_\sigma) \).

**Example 3.** Let \( P \) and \( Q \) be finite state processes and \( \Gamma \) be a type environment such that \( \Gamma \vdash P, Q \). Even if \( R = \square P \mid Q \) might be an infinite state process, we can easily check whether \( \Gamma \triangleright R \in NI(\cong_\sigma) \) just exploiting the decidability of \( \Gamma \triangleright P \in W(\cong_\sigma) \) and \( \Gamma \triangleright Q \in W(\cong_\sigma) \) and the compositionality of \( NI(\cong_\sigma) \) with respect to the parallel composition and replication operators.
4.1 Noninterference through a Partial Equivalence Relation

In [17, 16] the notion of noninterference for sequential and multithreaded programs is expressed in terms of a partial equivalence relation (per model) which captures the view of a $\sigma$-level observer. Intuitively, a configuration $C$, representing a program and the current state of the memory, is secure if $C \sim_{\sigma} C$ where $\sim_{\sigma}$ is a symmetric and transitive relation modeling the $\sigma$-level observation of program executions. The relation $\sim_{\sigma}$ is in general not reflexive, but it becomes reflexive on the set of secure configurations.

Below we show how this approach can be adapted to the $\pi$-calculus to characterize the class of $\sigma$-noninterfering processes. We first introduce the following notion of partial bisimilarity up to $\sigma$-high actions, $\approx_{\sigma}$. Intuitively, $\approx_{\sigma}$ requires that $\sigma$-high actions are simulated by internal transitions, while on the remaining actions it behaves as $\approx_{\sigma}$.

**Definition 6 (Partial Bisimilarity up to $\sigma$-high actions $\approx_{\sigma}$).** Let $\sigma \in \Sigma$. Partial bisimilarity up to $\sigma$-high actions is the largest symmetric relation $\approx_{\sigma}$ over configurations, such that whenever $\Gamma \vdash P \approx_{\sigma} Q$

- if $\Gamma \vdash P \xrightarrow{\alpha}_{\sigma} \Gamma' \vdash P'$, then there exists $Q'$ such that $\Gamma \vdash Q \xrightarrow{\alpha} \Gamma' \vdash Q'$ with $\Gamma' \vdash Q' \approx_{\sigma} P'$.
- if $\Gamma \vdash P \xrightarrow{\alpha}_{\sigma} \Gamma \vdash P'$ with $\alpha \in \{\pi(m), n(m)\}$, then there exists $Q'$ such that $\Gamma \vdash Q \xrightarrow{\alpha} \Gamma \vdash Q'$ with $\Gamma \vdash Q' \approx_{\sigma} P'$.
- if $\Gamma \vdash P \xrightarrow{\alpha}_{\sigma} \Gamma, m : T \vdash P'$ with $\alpha \in \{\nu m: T \pi(m), \nu m: T n(m)\}$, then there exists $Q'$ such that $\Gamma \vdash Q \xrightarrow{\alpha} \Gamma \vdash Q'$ with $\Gamma \vdash Q' \approx_{\sigma} (\nu m : T)P'$ and $\Gamma, m : T \vdash P' \approx_{\sigma} P'$.

The relation $\approx_{\sigma}$ is a partial equivalence relation, i.e., it is not reflexive. In fact, if we consider the process $P = h()().0$ and the type environment $\Gamma = h : \top[]$, $\ell : \bot[]$ we get $\Gamma \vdash P \not\approx_{\sigma} P$ when $\sigma = \bot$.

The next theorem states that $\approx_{\sigma}$ is reflexive on the set of well typed non-interfering processes. The proof exploits a sort of persistence property of $\approx_{\sigma}$, that is: if $\Gamma \vdash P \approx_{\sigma} P$, then for all $\Gamma' \vdash P'$ such that $\Gamma' \vdash P \sim \Gamma' \vdash P'$, it holds $\Gamma' \vdash P' \approx_{\sigma} P'$.

**Theorem 4.** Let $\sigma \in \Sigma$, $P$ be a process and $\Gamma$ be a type environment such that $\Gamma \vdash P$. $\Gamma \vdash P \in NI(\approx_{\sigma})$ if and only if $\Gamma \vdash P \approx_{\sigma} P$.

4.2 Noninterference through Name Restriction

In [5] the $P_{BNDC}$ property for CCS processes is characterized in terms of a single bisimulation-like equivalence check. We show that the same idea can be applied to the $\pi$-calculus. Let us first introduce the following definition.

**Definition 7.** Let $\sigma \in \Sigma$, $P$ be a process and $\Gamma$ be a type environment such that $\Gamma \vdash P$. We denote by $(\nu^{\sigma})P$ the process $(\nu m_1 : T_1)\ldots(\nu m_k : T_k)P$ where $m_1, \ldots, m_k$ are all the free names occurring in $P$ such that $\Gamma(m_i) = T_i$ and $A(T_i) > \sigma$. 

13
Definition 6 of partial bisimilarity up to \( \sigma \)-high actions can be modified as follows in order to obtain an equivalence relation.

**Definition 8 (Bisimilarity up to \( \sigma \)-high actions \( \approx_\sigma \)).** Let \( \sigma \in \Sigma \). Bisimilarity up to \( \sigma \)-high actions is the largest symmetric relation \( \approx_\sigma \) over configurations, such that whenever \( \Gamma \vdash P \approx_\sigma Q \)

- if \( \Gamma \triangleright P \xrightarrow{\alpha}_\sigma \Gamma' \triangleright P' \), then there exists \( Q' \) such that \( \Gamma \triangleright Q \xrightarrow{\alpha}_\sigma \Gamma' \triangleright Q' \) with \( \Gamma' \vdash Q' \approx_\sigma P' \).
- if \( \Gamma \triangleright P \xrightarrow{\alpha}_\sigma \Gamma \triangleright P' \) with \( \alpha \in \{ \nu(m), n(m) \} \), then there exists \( Q' \) such that either \( \Gamma \triangleright Q \xrightarrow{\alpha}_\sigma \Gamma \triangleright Q' \) with \( \Gamma \vdash Q' \approx_\sigma P' \) or \( \Gamma \triangleright Q \xrightarrow{\alpha}_\sigma \Gamma' \triangleright Q' \) with \( \Gamma' \vdash Q' \approx_\sigma P' \).
- if \( \Gamma \triangleright P \xrightarrow{\alpha}_\sigma \Gamma, m : T \triangleright P' \) with \( \alpha \in \{ (\nu m : T) \pi(m), (\nu m : T) n(m) \} \), then there exists \( Q' \) such that either \( \Gamma \triangleright Q \xrightarrow{\alpha}_\sigma \Gamma, m : T \triangleright Q' \) with \( \Gamma, m : T \vdash Q' \approx_\sigma P' \) or \( \Gamma \triangleright Q \xrightarrow{\alpha}_\sigma \Gamma' \triangleright Q' \) with \( \Gamma \vdash Q' \approx_\sigma (\nu m : T)P' \) and \( \Gamma, m : T \vdash P' \approx_\sigma (\nu \sigma)P' \).

We can now characterize \( \mathcal{NI}(\approx_\sigma) \) in terms of a single equivalence check between \( P \) and \( (\nu \sigma)P \) through \( \approx_\sigma \). The proof of the next theorem exploits the fact that if \( \Gamma \vdash P \approx_\sigma (\nu \sigma)P \), then for all \( \Gamma' \triangleright P' \) such that \( \Gamma \triangleright P \sim \Gamma' \triangleright P' \), it holds \( \Gamma' \triangleright P \approx_\sigma (\nu \sigma)P' \).

**Theorem 5.** Let \( \sigma \in \Sigma \), \( P \) be a program and \( \Gamma \) be a type environment such that \( \Gamma \vdash P \). \( \Gamma \triangleright P \in \mathcal{NI}(\approx_\sigma) \) if and only if \( \Gamma \vdash P \approx_\sigma (\nu \sigma)P \).

**Corollary 1.** Let \( \sigma \in \Sigma \), \( P \) be a process and \( \Gamma \) be a type environment such that \( \Gamma \vdash P \) and \( \forall n \in \text{fn}(P) \), \( \Lambda(n) \leq \sigma \) (i.e., \( P \) has no free \( \sigma \)-high level names). Then \( \Gamma \triangleright P \in \mathcal{NI}(\approx_\sigma) \).

**Proof.** The fact that \( P \) has no free \( \sigma \)-high level names implies that \( P \) is of the form \( (\nu \sigma)P \), hence \( P \approx_\sigma (\nu \sigma)P \) holds by reflexivity of \( \approx_\sigma \).

**Example 4.** Let \( P_1 = h(). \ell() \mid \ell(h()) \), \( P_2 = h(x:T). \text{if } x = n \text{ then } \ell(h()) \text{ else } \ell_2() \mid \ell(n) \mid \ell(m) \) and \( \Gamma \) and \( \Gamma' \) be the processes and the type environments of Example 2. We have seen that \( \Gamma \triangleright P_1 \not\in \mathcal{NI}(\approx_\sigma) \) and \( \Gamma' \triangleright P_3 \not\in \mathcal{NI}(\approx_\sigma) \). Now, by Corollary 1, we can immediately state that both \( \Gamma \triangleright (\nu h)P_1 \in \mathcal{NI}(\approx_\sigma) \) and \( \Gamma' \triangleright (\nu h)P_3 \in \mathcal{NI}(\approx_\sigma) \).

Notice that a process whose free names have a security level higher than \( \sigma \) is, in general, not secure. For instance, let \( \Gamma \) be the type environment \( h : \top[\cdot], \ell : \bot[\cdot] \) and \( P \) be the process \( h(x : \bot[\cdot]) \pi(). \) Assuming that \( \sigma < \top \), we have that the only free name \( h \) occurring in \( P \) has a security level higher than \( \sigma \). It is easy to see that \( \Gamma \triangleright P \not\in \mathcal{NI}(\approx_\sigma) \): in fact, by choosing \( H = \ell(h()) \), we have \( \Gamma \vdash P \not\approx_\sigma P \mid H \), that is \( P \) is insecure.

We conclude this section observing that, as in [4] for CCS, the definition of \( \sigma \)-noninterference can be also expressed in terms of bisimilarity on \( \top \)-actions over well-typed processes whose \( \sigma \)-high level names are restricted. This comes as a corollary of the following property.
Proposition 6. Let $\sigma \in \Sigma$, $P$ and $Q$ be two processes and $\Gamma$ be a type environment such that $\Gamma \vdash P, Q$. $\Gamma \vdash P \equiv_{\sigma} Q$ if and only if $\Gamma \vdash (\nu^\sigma)P \equiv_{\tau} (\nu^\sigma)Q$.

Corollary 2. Let $\sigma \in \Sigma$, $P$ be a process and $\Gamma$ be a type environment such that $\Gamma \vdash P$. $\Gamma \vdash P \in \mathcal{NI}(\equiv_{\sigma})$ if and only if for all $\Gamma' \triangleright P'$ such that $\Gamma \triangleright P \sim \Gamma' \triangleright P'$ and for all $H \in \mathcal{H}_{\sigma}'$, it holds $\Gamma' \vdash (\nu^\sigma)P' \equiv_{\tau} (\nu^\sigma)(P' \mid H)$.

5 Related Work

In this paper we develop a theory of noninterference for processes of the typed $\pi$-calculus. In the literature there are a number of works which study type-based techniques for noninterference. Hennessy and Riely [10, 8] consider a typed version of the asynchronous $\pi$-calculus where types associate read/write capabilities to channels as well as security clearances. They study noninterference properties based on $\textit{may}$ and $\textit{must}$ equivalences. Honda, Yoshida and Vasconcelos [11, 12] consider advanced type systems for processes of the linear/affine $\pi$-calculus where each action type is associated to a secrecy level. Their noninterference results are expressed in terms of typed $\textit{bisimulation}$ equivalences. In [14] Pottier develops a type theory which is, roughly, as expressive as the one of Hennessy and Riely [10] and proves a noninterference result based on $\textit{bisimulation}$ equivalence. Kobayashi in [13] proposes a refinement of a previous type system for deadlock/livelock-freedom and shows that well-typed processes enjoy a $\textit{bisimulation}$-based noninterference property.

In all these works types play an essential role in the proof of noninterference: the security of processes is ensured by the strong constraints imposed by the type systems. On the contrary, our approach relies on a much simpler typing discipline. Indeed, our type system does not deal with implicit information flow. Instead, we characterize noninterference in terms of the actions that may be performed by typed processes. In particular, we provide a number of sound and complete characterizations of $\sigma$-noninterference that can be efficiently used both to check the security of typed processes and to incrementally build systems which are secure by construction.

Another difference with respect to previous works is that they deal with open terms, while our theory applies to closed processes. However, the results presented in this paper scale to open terms by:

- introducing the open extension of $\equiv_{\sigma}$ as the type-indexed relation $\equiv_{\sigma}^o$ over terms such that $\Gamma \vdash T \equiv_{\sigma}^o U$ if and only if $\Gamma' \vdash T \rho \equiv_{\sigma}^o U \rho$ for all closing substitution $\rho$ which respects $\Gamma$ with $\Gamma'$, and

- saying that a term $T$ satisfies the $\sigma$-noninterference property in $\Gamma$, written $\Gamma \triangleright T \in \mathcal{NI}(\equiv_{\sigma}^o)$, if for all closing substitution $\rho$ which respects $\Gamma$ with $\Gamma'$, $\Gamma' \triangleright T \rho \in \mathcal{NI}(\equiv_{\sigma})$.

1. We say that $\rho = \{x_1 := m_1, \ldots, x_n := m_n\}$ is a substitution which respects $\Gamma$ with $\Gamma'$ if $\Gamma = \Delta, x_1 : T_1, \ldots, x_n : T_n$ and there exists $\Delta'$ such that $\Gamma' = \Delta, \Delta'$ and $\Gamma' \vdash m_i : T_i$ for $i = 1, \ldots, n$. 

15
In this way, we obtain that if $\Gamma \vdash T \in NT(\cong_o)$ then for all $H \in \mathcal{H}_T$, it holds $\Gamma \models T \cong_o T | H$.

Finally, we think that our characterizations could provide a basis for studying noninterference in more realistic scenarios, e.g., distributed systems admitting mechanisms for *downgrading* or *declassifying* information. We leave this topic for future work.

References

A  Proofs omitted from Section 3

In the following we write $\Gamma \vdash_\sigma P$ to state that $P$ is a process of level at most $\sigma$ in $\Gamma$, i.e., $\forall n \in \text{fn}(P)$, $A(\Gamma(n)) \leq \sigma$. Similarly, we write $\Gamma \vdash_\sigma C[\Gamma']$ to state that $C[\Gamma']$ is a $\sigma$-context.

Lemma 1. Let $\sigma \in \Sigma$, $\Gamma \vdash_\sigma R$ and $\Gamma \triangleright R \xrightarrow{\alpha} \Gamma' \triangleright R'$. Then $\Gamma' \vdash_\sigma R'$.

Proof. Immediate.

Proposition 7. Let $\sigma \in \Sigma$. $\approx_\sigma$ is a congruence with respect to $\sigma$-contexts.

Proof. The proof that $\approx_\sigma$ is an equivalence relation is standard. We prove that $\approx_\sigma$ is preserved by all $\sigma$-contexts. We consider all the constructs simultaneously: let $S$ be a binary relation such that

- $\approx_\sigma \subseteq S$
- $\Gamma \vdash P \triangleright Q$ implies $\Gamma, \Gamma' \vdash (P \triangleright R) \triangleright (Q \triangleright R)$ for all process $R$ such that $\Gamma, \Gamma' \vdash_\sigma R$.
- $\Gamma \vdash P \triangleright Q$ implies $\Gamma' \vdash (\nu n : T)P \triangleright (\nu n : T)Q$ where $\Gamma = \Gamma'$, $n : T$.

We show that $S$ is a bisimulation on $\sigma$-actions, hence $S \subseteq \approx_\sigma$. The proof is by induction on the formation of $S$.

- The case where $\Gamma \vdash P \triangleright Q$ since $\Gamma \vdash P \approx_\sigma Q$ is trivial.
- Let $\Gamma, \Gamma' \vdash (P \triangleright R) \triangleright (Q \triangleright R)$ with $\Gamma \vdash P \triangleright Q$ and $\Gamma, \Gamma' \vdash_\sigma R$. Assume $\Gamma, \Gamma' \triangleright P \triangleright R \xrightarrow{\alpha} \Gamma'' \triangleright P'$, this comes from one of following cases:
  - $\Gamma, \Gamma' \triangleright P \triangleright R \xrightarrow{\alpha} \Gamma'' \triangleright P''$, $\text{bn}(\alpha) \cap \text{fn}(R) = \emptyset$ and $P'' = P'' \triangleright R$. From $\Gamma \vdash P \triangleright Q$, by inductive hypothesis we have that $\exists Q'$ such that $\Gamma, \Gamma' \triangleright Q \xrightarrow{\delta} \Gamma'' \triangleright Q'$ and $\Gamma'' \vdash P'' \triangleright Q'$. Then by rule (PAR) of the LTS, $\Gamma, \Gamma' \triangleright Q \triangleright R \xrightarrow{\delta} \Gamma'' \triangleright Q' \triangleright R$. Now, from $\Gamma'' \vdash P'' \triangleright Q'$ by definition of $S$ we conclude $\Gamma'' \vdash (P'' \triangleright R) \triangleright (Q' \triangleright R)$ as desired.

Proposition 8. Let $\sigma \in \Sigma$. $\approx_\sigma$ is reduction closed.

Proof. Let $P, Q$ be processes such that $\Gamma \vdash P \approx_\sigma Q$ and $P \nrightarrow^* P'$. Then by rule (RED) of the LTS, $\Gamma \vdash P \nrightarrow^*_\sigma \Gamma \vdash P'$. By definition of $\approx_\sigma$, $\Gamma \vdash Q \Rightarrow \Gamma \vdash Q'$ and $\Gamma \vdash P' \approx_\sigma Q'$ as desired.
Proposition 9. Let $\sigma \in \Sigma$. $\sim_\sigma$ is $\sigma$-barb preserving.

Proof. Let $P, Q$ be processes such that $\Gamma \vdash P \approx_\sigma Q$ and $\Gamma \vdash P \downarrow_\sigma^n$, that is $P \xrightarrow{\pi(m)}^{n(m)} P'$. In this case, by Proposition 3 and the hypothesis $\Lambda(\Gamma(m)) \leq \sigma$, we have $\Gamma \triangleright P \xrightarrow{\pi(m)}^{n(m)} \Gamma \triangleright P'$. Now, by definition of $\approx_\sigma$ we also have $\Gamma \triangleright Q \xrightarrow{\pi(m)}^{n(m)} \Gamma \triangleright Q'$, then $\Gamma \vdash Q \downarrow_\sigma^n$ as desired.

Lemma 2. If $\Gamma, \omega ; \sigma[] \models (\bar{\omega}() \mid P) \cong_\sigma (\bar{\omega}() \mid Q)$ with $\omega$ fresh in $P, Q$, then $\Gamma \vdash P \cong_\sigma Q$.

Proof. It is sufficient to prove that the following relation

$$R = \{(\Delta \triangleright P, \Delta \triangleright Q) \mid \Gamma, \omega ; \sigma[] \models (\bar{\omega}() \mid P) \cong_\sigma (\bar{\omega}() \mid Q) \omega \notin \text{fn}(P) \cup \text{fn}(Q)\}$$

is reduction closed, $\sigma$-barb preserving and $\sigma$-contextual. See [9] for details.

Lemma 3. If $\Gamma, \omega ; \sigma[T] \models (\nu m ; T)(P \mid \bar{\omega}(m)) \cong_\sigma (\nu m ; T)(Q \mid \bar{\omega}(m))$ with $\omega$ fresh in $P, Q$, then $\Gamma, m ; T \vdash P \cong_\sigma Q$.

Proof. Let $R$ be the following relation:

$$R = \{(\Delta, m ; T \triangleright P, \Delta, m ; T \triangleright Q) \mid \Gamma, \omega ; \sigma[T] \models (\nu m ; T)(P \mid \bar{\omega}(m)) \cong_\sigma (\nu m ; T)(Q \mid \bar{\omega}(m)) \omega \notin \text{fn}(P) \cup \text{fn}(Q)\}$$

It is sufficient to prove that $R$ is reduction closed, $\sigma$-barb preserving and $\sigma$-contextual. See [9] for details.

Proposition 10. For any process $P, Q$, if $\Gamma \vdash P \cong_\sigma Q$ then $\Gamma \vdash P \approx_\sigma Q$.

Proof. Let $S$ be the following relation:

$$S = \{(\Delta \triangleright P, \Delta \triangleright Q) \mid \Gamma \vdash P \cong_\sigma Q\}.$$

We prove that $S$ is a bisimulation on $\sigma$ actions; then $\cong_\sigma \subseteq \approx_\sigma$ follows by the fact that $\approx_\sigma$ is the largest bisimulation on $\sigma$-actions. Assume $\Gamma \triangleright P \xrightarrow{\alpha} \Gamma' \triangleright P'$, we distinguish the following cases:

- $\Gamma \triangleright P \xrightarrow{n(m)} \Gamma \triangleright P'$ with $\Gamma \vdash n : \sigma_1[T], \Gamma \vdash m : T$ and $\sigma_1 \leq \sigma$. Now, let be

$$C_\omega[] = \pi(m) \bar{\omega}() \mid []$$

where $\omega$ is fresh and $\Gamma, \omega ; \sigma[] \models \sigma C_\omega[] \mid []$. Let $\Gamma'' = \Gamma, \omega ; \sigma[]$. Then $C_\omega[]$ is a $\sigma$-context and $\Gamma'' \xrightarrow{\sigma} \Gamma'' \triangleright \bar{\omega}() \mid P''$ with $\Gamma'' \vdash (\bar{\omega}() \mid P'') \downarrow_\sigma$. Now, from $\Gamma \vdash P \cong_\sigma Q$ and the fact that $\cong_\sigma$ is $\sigma$-contextual, we have that there exists $Q''$ such that $\Gamma'' \triangleright C_\omega[] \Rightarrow \Gamma'' \triangleright Q''$ with $\Gamma'' \vdash (\bar{\omega}() \mid P'') \cong_\sigma Q''$ and $\Gamma'' \vdash Q'' \downarrow_\sigma$. Then $Q'' = \bar{\omega}() \mid Q'$, and by Lemma 2, $\Gamma \vdash P' \cong_\sigma Q'$. Since $\Gamma'' \vdash C_\omega[] \downarrow_\sigma^m$, we have that $\Gamma \triangleright Q \xrightarrow{n(m)} \Gamma \triangleright Q'$ and we conclude $(\Gamma \triangleright P', \Gamma \triangleright Q') \in S$ as desired.
\[ \Gamma \vdash P \quad \xrightarrow{\tau} \quad \Gamma \vdash P'. \] From the hypothesis \( \Gamma \vdash P \simeq_{\sigma} Q \), we have \( \Gamma \vdash Q \quad \Rightarrow \quad \Gamma \vdash Q' \) with \( \Gamma \vdash P' \simeq_{\sigma} Q' \), and we conclude \( (\Gamma \vdash P', \Gamma \vdash Q') \in S \) as desired.

\[ \Gamma \vdash P \xrightarrow{(m)\sigma} \quad \Gamma \vdash P' \quad \text{with} \quad \Gamma \vdash n : \sigma_1[T], \quad \Gamma \vdash m : T \quad \text{and} \quad \sigma_1 \preceq \sigma. \] Now, let

\[ C_{\omega}[\cdot] = [\cdot] \mid n(x : T), \text{if } x = m \text{ then } \overline{\omega}(\cdot) \text{ else } \overline{\omega}(\cdot) \]

where \( \omega_1, \omega_2 \) are fresh and \( \Gamma, \omega_1: \sigma[T], \omega_2: \sigma[T] \vdash_{\sigma} C_{\omega}[\cdot] \). Let \( \Gamma' = \Gamma, \omega_1: \sigma[T], \omega_2: \sigma[T]. \) Then \( C_{\omega}[\cdot] \) is a \( \sigma \)-context and \( \Gamma' \vdash C_{\omega}[P] \quad \Rightarrow \quad \Gamma' \vdash \overline{\omega}(\cdot) \mid P' \) with \( \Gamma' \vdash P = (\overline{\omega}(\cdot) \mid P') \). Now, from \( \Gamma \vdash P \simeq_{\sigma} Q \) and the fact that \( \simeq_{\sigma} \) is \( \sigma \)-contextual, we have that there exists \( Q'' \) such that \( \Gamma' \vdash C_{\omega}[Q] \quad \Rightarrow \quad \Gamma' \vdash Q'' \) with \( \Gamma' \vdash \overline{\omega}(\cdot) \mid P' \simeq_{\sigma} Q'' \) and \( \Gamma' \vdash Q'' \downarrow^{\sigma_1}_{\omega_1} \). Then \( Q'' \simeq_{\sigma} Q \), and by Lemma 2, \( \Gamma \vdash P' \simeq_{\sigma} Q' \). Since \( \Gamma' \vdash C_{\omega}[Q] \downarrow^{\sigma_0}_{\omega_1} \), we have that \( \Gamma \vdash Q \quad \Rightarrow \quad \Gamma \vdash Q' \) and we conclude \( (\Gamma \vdash P', \Gamma \vdash Q') \in S \) as desired.

\[ \Gamma \vdash P \xrightarrow{(m'):\sigma} \quad \Gamma, m : T \vdash P' \quad \text{which comes from} \quad P = (\nu m : T)P_1, \quad \Gamma \vdash n : \sigma_1[T] \quad \text{with} \quad \sigma_1 \preceq \sigma, \quad \text{and} \quad \Gamma, m : T \vdash P_1 \xrightarrow{(m)\sigma} \quad \Gamma, m : T \vdash P'. \] Let \( \{p_1, \ldots, p_k\} = \{p \mid p \in \text{fn}(P) \cup \text{fn}(Q) \} \) and \( \Gamma \vdash p : T \), and let

\[ C_{\omega}[\cdot] = [\cdot] \mid n(x : T), \text{if } x = p_1 \text{ then } \overline{\omega}(\cdot) \text{ else } \overline{\omega}(\cdot) \]

\[ \vdots \]

\[ \text{if } x = p_k \text{ then } \overline{\omega}(\cdot) \text{ else } \overline{\omega}(x) \]

with \( \omega_1, \omega_2 \) fresh and \( \Gamma, \omega_1: \sigma[T], \omega_2: \sigma[T] \vdash_{\sigma} C_{\omega}[\cdot] \). Let \( \Gamma' = \Gamma, \omega_1: \sigma[T], \omega_2: \sigma[T]. \) Then \( C_{\omega}[\cdot] \) is a \( \sigma \)-context and \( \Gamma' \vdash C_{\omega}[P] \quad \Rightarrow \quad \Gamma' \vdash \overline{\omega}(\cdot) \mid P' \) with \( \Gamma' \vdash P = (\overline{\omega}(\cdot) \mid P') \). Now, from \( \Gamma \vdash P \simeq_{\sigma} Q \) and the fact that \( \simeq_{\sigma} \) is \( \sigma \)-contextual, we have that there exists \( Q'' \) such that \( \Gamma' \vdash C_{\omega}[Q] \quad \Rightarrow \quad \Gamma' \vdash Q'' \) with \( \Gamma' \vdash (\nu m : T)(\overline{\omega}(\cdot) \mid P') \simeq_{\sigma} Q'' \) and \( \Gamma' \vdash Q'' \downarrow^{\sigma_0}_{\omega_0} \). Since \( \Gamma' \vdash C_{\omega}[Q] \downarrow^{\sigma_0}_{\omega_1} \), we have that \( \Gamma \vdash Q \quad \Rightarrow \quad \Gamma \vdash (\nu m : T)Q \quad \text{with} \quad Q'' = (\nu m : T)(\overline{\omega}(\cdot) \mid P') \simeq_{\sigma} (\nu m : T)(\overline{\omega}(\cdot) \mid Q') \), by Lemma 3 we have \( \Gamma, m : T \vdash P' \simeq_{\sigma} Q' \) and we conclude \( (\Gamma, m : T \vdash P', \Gamma, m : T \vdash Q') \in S \) as desired.

\[ \Gamma \vdash P \xrightarrow{(m):\sigma} \quad \Gamma, m : T \vdash P' \quad \text{where} \quad \Gamma \vdash n : \sigma_1[T] \quad \text{and} \quad \sigma_1 \preceq \sigma. \] Now, let

\[ C_{\omega}[\cdot] = [\cdot] \mid (\nu m : T)(\overline{\omega}(\cdot) \mid P) \]

where \( \omega \) is fresh and \( \Gamma, \omega: \sigma[T] \vdash_{\sigma} C_{\omega}[\cdot] \). Let \( \Gamma' = \Gamma, \omega: \sigma[T]. \) Then \( C_{\omega}[\cdot] \) is a \( \sigma \)-context and \( \Gamma' \vdash C_{\omega}[P] \quad \Rightarrow \quad \Gamma' \vdash (\nu m : T)(\overline{\omega}(\cdot) \mid P') \) with \( \Gamma' \vdash P = (\nu m : T)(\overline{\omega}(\cdot) \mid P') \). Now, from \( \Gamma \vdash P \simeq_{\sigma} Q \) and the fact that \( \simeq_{\sigma} \) is \( \sigma \)-contextual, we have that there exists \( Q'' \) such that \( \Gamma'' \vdash C_{\omega}[Q] \quad \Rightarrow \quad \Gamma'' \vdash Q'' \) with \( \Gamma'' \vdash (\nu m : T)(\overline{\omega}(\cdot) \mid P') \simeq_{\sigma} Q'' \) and \( \Gamma'' \vdash Q'' \downarrow^{\sigma_0}_{\omega_0} \). Since \( \Gamma'' \vdash C_{\omega}[Q] \downarrow^{\sigma_0}_{\omega_0} \), we have that \( \Gamma \vdash Q \quad \Rightarrow \quad \Gamma \vdash (\nu m : T)(\overline{\omega}(\cdot) \mid Q) \), by Lemma 3 we have \( \Gamma, m : T \vdash P' \simeq_{\sigma} Q' \) and we conclude \( (\Gamma, m : T \vdash P', \Gamma, m : T \vdash Q') \in S \) as desired.
**Proof of Theorem 1** For any $P, Q$, $\Gamma \vdash P \cong_\sigma Q$ if and only if $\Gamma \vdash P \equiv_\sigma Q$.

Proof. The proof of $\equiv_\sigma \subseteq \cong_\sigma$ comes by Propositions 7, 8 and 9. The converse comes by Proposition 10.

**B Proofs omitted from Section 4**

**Lemma 4 (Weakening).** Let $P$ be a process, $\Gamma$ and $\Gamma'$ two type environments such that $\Gamma \vdash P$ and $\Gamma, \Gamma' \vdash \theta$. Then

- if $\Delta \triangleright P \in W(\cong_\sigma)$ then $\Gamma, \Gamma' \triangleright P \in W(\cong_\sigma)$
- if $\Delta \triangleright P_1 \equiv_\sigma P_2$ then $\Gamma, \Gamma' \triangleright P_1 \equiv_\sigma P_2$.

**Definition 9 (Bisimulation up to restriction and up to $\cong_\sigma$).** A symmetric relation $\mathcal{R}$ over configurations is a bisimulation up to restriction and up to $\cong_\sigma$ if $(\Gamma \triangleright P) \mathcal{R} (\Gamma \triangleright Q)$ implies: $\Gamma \triangleright P \xrightarrow{\alpha} \Gamma' \triangleright P'$, then there exists $Q'$ such that $\Gamma \triangleright Q \xrightarrow{\delta} \Gamma' \triangleright Q'$ and $\Gamma' \triangleright P' \equiv_\sigma (\nu n : T)P''$, $\Gamma' \triangleright Q' \equiv_\sigma (\nu n : T)Q''$, with $(\Gamma', n : T \triangleright P'') \mathcal{R} (\Gamma', n : T \triangleright Q'')$.

**Theorem 6.** If $\mathcal{R}$ is a bisimulation up to restriction and $\equiv_\sigma$, then $\mathcal{R} \subseteq \equiv_\sigma$.

Proof. We define the relation $\mathcal{S}$ as the smallest relation such that:

1. $\Gamma \vdash P \mathcal{R} Q$ implies $\Gamma \vdash P \cong_\sigma Q$.
2. $\Gamma \vdash P \equiv_\sigma R, \Gamma \vdash RS \ U$ and $\Gamma \vdash U \equiv_\sigma Q$ imply $\Gamma \vdash P \cong_\sigma Q$.
3. $\Gamma, n : T \vdash P \cong_\sigma Q$ implies $\Gamma \vdash (\nu n : T)P \cong_\sigma (\nu n : T)Q$.

We prove by induction on its definition that $\mathcal{S}$ is a bisimulation on $\sigma$-actions. This will assure the soundness of $\mathcal{R}$ since $\mathcal{R} \subseteq \mathcal{S} \subseteq \equiv_\sigma$.

- Let $\Gamma \vdash P \cong_\sigma Q$ because $\Gamma \vdash P \mathcal{R} Q$. Assume $\Gamma \triangleright P \xrightarrow{\alpha} \Gamma' \triangleright P'$, then from $\Gamma \vdash P \mathcal{R} Q$ we know that $\exists Q'$ such that $\Gamma \triangleright Q \xrightarrow{\delta} \Gamma' \triangleright Q'$ with $\Gamma' \triangleright P' \equiv_\sigma (\nu n : T)P''$, $\Gamma' \triangleright Q' \equiv_\sigma (\nu n : T)Q''$, and $\Gamma', n : T \triangleright P'' \mathcal{R} \mathcal{Q}''$. Now, by definition of $\mathcal{S}$ (1.) we have $\Gamma', n : T \triangleright P'' \cong_\sigma \mathcal{Q}''$, then by (2.) $\Gamma' \vdash (\nu n : T)P \cong_\sigma (\nu n : T)Q$.

- Let $\Gamma \vdash P \cong_\sigma Q$ because $\Gamma \vdash P \equiv_\sigma R$, $\Gamma \vdash RS \ U$ and $\Gamma \vdash U \equiv_\sigma Q$. Assume $\Gamma \triangleright P \xrightarrow{\alpha} \Gamma' \triangleright P'$, then from $\Gamma \vdash P \equiv_\sigma R$ we know that $\exists R'$ such that $\Gamma \triangleright R \xrightarrow{\delta} \Gamma' \triangleright R'$ with $\Gamma' \triangleright P' \equiv_\sigma R'$. Now, from $\Gamma \vdash RS \ U$, by induction we have that $\Gamma \triangleright U \xrightarrow{\alpha} \Gamma' \triangleright U'$ with $\Gamma' \vdash R' \mathcal{S} \ U'$, and from $\Gamma \vdash U \equiv_\sigma Q$ we have $\Gamma \triangleright Q \xrightarrow{\delta} \Gamma' \triangleright Q'$ with $\Gamma' \vdash U' \equiv_\sigma Q'$. Then, by definition of $\mathcal{S}$ (2.) we conclude $\Gamma' \vdash P' \cong_\sigma Q'$.

- Let $\Gamma \vdash (\nu n : T)P \cong_\sigma Q$ because $\Gamma, n : T \vdash P \cong_\sigma Q$. Assume $\Gamma \triangleright (\nu n : T)P \xrightarrow{\alpha} \Gamma' \triangleright P'$, we distinguish two subcases:
such that \( \Gamma \mathrel{\subseteq} \forall S \) we prove that \( \approx \).

First, let be \( \Gamma \vdash P \) this comes by (Par). Assume now that \( \Gamma \vdash P \) this comes by (Comm) \( \Gamma' \vdash P' \) with \( \alpha = \langle \nu m: T \rangle \langle n \rangle \), \( \Gamma' = \Gamma, n: T \). Now, from \( \Gamma, n: T \vdash P \) by induction we have that \( \Gamma \vdash (\nu m: T)Q \mathrel{\delta} \Gamma' \vdash (\nu m: T)Q_1 \) and by definition of \( S \) (3.) \( \Gamma' \vdash (\nu m: T)P_1S(\nu m: T)Q_1 \) as desired.

**Proof of Theorem 2** Let \( \sigma \in \Sigma \), \( P \) be a process and let \( \Gamma \) be a type environment such that \( \Gamma \vdash P \). Then \( \Gamma \vdash P \in \mathcal{N}T(\equiv \sigma) \) if and only if \( \Gamma \vdash P \in \mathcal{W}(\equiv \sigma) \).

Proof. \((\Leftarrow)\) Let be \( \Gamma \vdash P \in \mathcal{W}(\equiv \sigma) \), we prove that \( \forall \Gamma' \vdash P' \) s.t. \( \Gamma \vdash \Gamma' \vdash P' \) and \( \forall H \in \mathcal{H}_T \), it holds \( \Gamma' \vdash P' \equiv \sigma P' \ | \ H \). Let be

\[
S = \{ (\Gamma \vdash P, \Gamma \vdash P \ | \ H) \mid \Gamma \vdash P \in \mathcal{W}(\equiv \sigma) \text{ and } H \in \mathcal{H}_T \}
\]

we prove that \( S \) is a bisimulation on \( \sigma \)-actions up to restriction and up to \( \approx \). The thesis follows by Theorem 6, the fact that \( \approx \) is the largest bisimulation on \( \sigma \)-actions, and Theorem 1.

First, let be \( \Gamma \vdash P \mathrel{\alpha} \Gamma' \vdash P' \); hence \( \Gamma \vdash P \ | \ H \mathrel{\alpha} \Gamma' \vdash P' \ | \ H \) by rule (Par) of LTS. Since \( \Gamma \vdash P \mathrel{\sim} \Gamma' \vdash P' \), by Proposition 5 we have \( \Gamma' \vdash P' \in \mathcal{W}(\equiv \sigma) \), then \( \Gamma' \vdash P' \), \( \Gamma' \vdash P' \ | \ H \) \( \in S \) as desired.

Assume now that \( \Gamma \vdash P \ | \ H \mathrel{\alpha} \Gamma' \vdash Q \), we distinguish the following cases:

- this comes by (Par) from \( \Gamma \vdash H \mathrel{\alpha} \Gamma' \vdash H' \) and \( Q = P \ | \ H' \). In this case, since \( H \in \mathcal{H}_T \), we have that \( \alpha = \tau \), \( \Gamma' = \Gamma \) and \( H' \in \mathcal{H}_T \), hence \( \Gamma \vdash P \ | \ H' \in S \) as desired.

- this comes by (Par) from \( \Gamma \vdash P \mathrel{\alpha} \Gamma' \vdash P' \) and \( Q = P' \ | \ H \). Since \( \Gamma \vdash \Gamma' \vdash P' \), by Proposition 5 we have \( \Gamma' \vdash P' \in \mathcal{W}(\equiv \sigma) \), then \( \Gamma' \vdash P' \), \( \Gamma' \vdash P' \ | \ H \) \( \in S \) as desired.

- this comes by (Comm) from \( \Gamma \vdash P \mathrel{\langle m \rangle \delta} \Gamma \vdash P' \), \( \Gamma \vdash H \mathrel{\langle n \rangle \delta} \Gamma \vdash H' \) and \( Q = P' \ | \ H' \). Since \( H \in \mathcal{H}_T \), we have that \( \sigma < \delta \) and \( H \in \mathcal{H}_T \). From the hypothesis \( \Gamma \vdash P \in \mathcal{W}(\equiv \sigma) \) we have that there exists \( P'' \) s.t. \( \Gamma \vdash P \mathrel{\Rightarrow} \Gamma \vdash P'' \) and \( \Gamma \vdash P' \equiv \sigma P'' \), then also \( \Gamma \vdash P' \approx \sigma P'' \). Now, by Proposition 5 we have \( \Gamma \vdash P' \in \mathcal{W}(\equiv \sigma) \), then \( (\Gamma \vdash P', \Gamma \vdash P' \ | \ H') \in S \), hence \( (\Gamma \vdash P'') \approx \sigma \ (\Gamma \vdash P') S \ (\Gamma \vdash P' | \ H') \) as desired.

- this comes by (Comm) from \( \Gamma \vdash P \mathrel{\langle m \rangle \delta} \Gamma \vdash P' \), \( \Gamma \vdash H \mathrel{\langle n \rangle \delta} \Gamma \vdash H' \) and \( Q = P' \ | \ H' \). This case is analogous to the previous one.

- this comes by (Close) from \( P \mathrel{\langle \nu m: T \rangle \langle m \rangle \delta} P', H \mathrel{\langle n \rangle \delta} H' \) and \( Q = (\nu m: T)(P' | \ H') \). By Propositions 3 and 4, \( \Gamma \vdash P \mathrel{\langle \nu m: T \rangle \langle m \rangle \delta} P', m: T \vdash P' \),
Lemma 5. Let $\Gamma \vdash (\nu m:T)P \rightsquigarrow \Gamma' \vdash P'$, then one of the following two cases:

1. $P' = (\nu m:T)P''$ and $\Gamma, m:T \vdash P \rightsquigarrow \Gamma', m:T \vdash P''$
2. $\Gamma' = \Gamma''$, $m:T$ and $\Gamma, m:T \vdash P \rightsquigarrow \Gamma'', m:T \vdash P'\'$

Proof. By induction on the length of $\Gamma \vdash (\nu m:T)P \rightsquigarrow \Gamma' \vdash P'$. If the length is one, then $\Gamma \vdash (\nu m:T)P \xrightarrow{\alpha} \Gamma' \vdash P'$ must have been derived in one of the following two ways:

- by rule (Res) from $\Gamma, m:T \vdash P \xrightarrow{\alpha} \Gamma', m:T \vdash P''$ where $P' = (\nu m:T)P''$, which is case 1.
- by rule (Open) from $\Gamma, m:T \vdash P \xrightarrow{\pi(m)} \Gamma, m:T \vdash P'$ where $\Gamma' = \Gamma, m:T$ and $\alpha = (\nu m:T)\pi(m)$, which is case 2.

Let now consider the inductive case where $\Gamma \vdash (\nu m:T)P \rightsquigarrow \Gamma_1 \vdash P_1 \xrightarrow{\alpha} \Gamma' \vdash P'$. By induction we have two cases:

- $P_1 = (\nu m:T)P_2$ and $\Gamma, m:T \vdash P \rightsquigarrow \Gamma_1, m:T \vdash P_2$. We distinguish two subcases depending on the rule that gave $\Gamma_1 \vdash P_1 \xrightarrow{\alpha} \Gamma' \vdash P'$:
• it comes by \((\text{RES})\) from \(\Gamma_1, m : T \triangleright P_2 \xrightarrow{\alpha}_{\sigma} \Gamma', m : T \triangleright P''\) and \(P' = (\nu m : T)P''\). Hence \(\Gamma, m : T \triangleright P \Rightarrow \Gamma', m : T \triangleright P''\) which is case 1.

• it comes by \((\text{OPEN})\) from \(\Gamma_1, m : T \triangleright P_2 \xrightarrow{\pi(m)}_{\sigma} \Gamma_1, m : T \triangleright P'\) with \(\alpha = (\nu m : T)\pi(m)\) and \(\Gamma' = \Gamma_1, m : T\). Hence \(\Gamma, m : T \triangleright P \Rightarrow \Gamma_1, m : T \triangleright P'\) which is case 2.

- \(\Gamma_1 = \Gamma_2, m : T\) and \(\Gamma, m : T \triangleright P \Rightarrow \Gamma_2, m : T \triangleright P_1\). In this case we have \(\Gamma, m : T \triangleright P \Rightarrow \Gamma' \triangleright P'\) with \(\Gamma' = \Gamma''\) for some \(\Gamma''\), which is case 2.

Proposition 11. If \(\Gamma, m : T \triangleright P \in W(\equiv_{\sigma})\), then \(\Gamma \triangleright (\nu m : T)P \in W(\equiv_{\sigma})\).

Proof. Let \(\Gamma' \triangleright P'\) be a configuration such that \(\Gamma \triangleright (\nu m : T)P \Rightarrow \Gamma' \triangleright P'\), then we prove the following two items:

1. if \(\Gamma' \triangleright P'\) \begin{align*}
\xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright P_2 \quad &\text{with } \alpha \in \{\pi(q), p(q)\}, \quad \exists P_3 \text{ s.t. } \Gamma' \triangleright P' \Rightarrow \Gamma' \triangleright P_3 \text{ and } \Gamma' \triangleright P_3 \equiv_{\sigma} P_3. 
\end{align*}

2. if \(\Gamma' \triangleright P'\) \begin{align*}
\xrightarrow{\alpha}_{\sigma} \Gamma', q : T' \triangleright P_2 \quad &\text{with } \alpha \in \{(\nu q : T')\pi(q), (\nu q : T')p(q)\}, \quad \exists P_3 \text{ s.t. } \Gamma' \triangleright P' \Rightarrow \Gamma' \triangleright P_3 \text{ and } \Gamma' \triangleright P_3 \equiv_{\sigma} (\nu q : T')P_2. 
\end{align*}

By Lemma 5 we distinguish two cases:

- \(P' = (\nu m : T)P''\) and \(\Gamma, m : T \triangleright P \Rightarrow \Gamma', m : T \triangleright P''\). Let prove the two items in the definition of \(W(\equiv_{\sigma})\):

  1. if \(\Gamma' \triangleright P'\) \begin{align*}
\xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright P_2 \quad &\text{with } \alpha \in \{\pi(q), p(q)\}. \quad \text{Since } P' = (\nu m : T)P'', \text{ the action comes by } (\text{RES}) \text{ from } \Gamma', m : T \triangleright P'' \xrightarrow{\alpha}_{\sigma} \Gamma', m : T \triangleright P''\text{ where } P_2 = (\nu m : T)P''\text{ and } m \notin \text{fu}(\alpha) \cup \text{bn}(\alpha). \quad \text{From } \Gamma, m : T \triangleright P \Rightarrow \Gamma', m : T \triangleright P''\text{ and } \Gamma, m : T \triangleright P \in W(\equiv_{\sigma}), \text{ there exists } P^* \text{ such that } \Gamma', m : T \triangleright P'' \Rightarrow \Gamma', m : T \triangleright P^* \text{ with } \Gamma', m : T \triangleright P^* \equiv_{\sigma} P''\text{. Since } \equiv_{\sigma} \text{ is a congruence, we also have } \Gamma' \triangleright (\nu m : T)P^* \equiv_{\sigma} (\nu m : T)P''\text{, and we conclude observing that by } (\text{RES}) \text{ we have } \Gamma' \triangleright P' \Rightarrow \Gamma' \triangleright (\nu m : T)P''\text{.}

  2. if \(\Gamma' \triangleright P'\) \begin{align*}
\xrightarrow{\alpha}_{\sigma} \Gamma', q : T' \triangleright P_2 \quad &\text{with } \alpha \in \{(\nu q : T')\pi(q), (\nu q : T')p(q)\}. \quad \text{We distinguish two subcases:}
\end{align*}

    • \(q : T' \neq m : T\), then \(\Gamma' \triangleright P'\) \begin{align*}
\xrightarrow{\alpha}_{\sigma} \Gamma', q : T' \triangleright P_2 \quad &\text{comes by } (\text{RES}) \text{ from } \Gamma', m : T \triangleright P'' \xrightarrow{\alpha}_{\sigma} \Gamma', q : T' \triangleright P''\text{ with } P_2 = (\nu m : T)P''\text{. Now from } \Gamma, m : T \triangleright P \Rightarrow \Gamma', m : T \triangleright P'' \text{ and } \Gamma, m : T \triangleright P \in W(\equiv_{\sigma}) \text{ we have that there exists } P^* \text{ such that } \Gamma', m : T \triangleright P'' \Rightarrow \Gamma', m : T \triangleright P^* \text{ and } \Gamma', m : T \triangleright P^* \equiv_{\sigma} (\nu q : T')P''\text{. Then by rule } (\text{RES}) \text{ we also have } \Gamma' \triangleright (\nu m : T)P^* \Rightarrow \Gamma' \triangleright (\nu m : T)P'\text{ and } \Gamma' \triangleright (\nu m : T)P^* \equiv_{\sigma} (\nu m : T)(\nu q : T')P'' = (\nu q : T')P_2 \text{ since } \equiv_{\sigma} \text{ is a congruence.}
\end{align*}
Proposition 12. Let $P$ and $Q$ be two closed processes and $\Gamma$ be a type environment such that $\Gamma \vdash P, Q$. If $\Gamma \triangleright P \in W(\equiv_\sigma)$ and $\Gamma \triangleright Q \in W(\equiv_\sigma)$ then $\Gamma \triangleright P \mid Q \in W(\equiv_\sigma)$.

Proof. Consider the following relation $S$:

$$S = \{(\Gamma \triangleright P_1 \mid Q_1, \Gamma \triangleright P_2 \mid Q_2) \mid \Gamma \triangleright P_i, \Gamma \triangleright Q_i \in W(\equiv_\sigma) \text{ i = 1, 2}$$

$$\Gamma \vdash P_1 \equiv_\sigma P_2 \quad \Gamma \vdash Q_1 \equiv_\sigma Q_2 \}$$

it is sufficient to prove that $S$ is a partial bisimulation up to $\sigma$-high actions up to restriction and structural congruence $\equiv$. This implies $S \subseteq \equiv_\sigma$, then $\Gamma \vdash P \mid Q \equiv_\sigma P \mid Q$, and we conclude by Theorem 4 and Theorem 2. Let distinguish the following cases:

- $\Gamma \triangleright P_1 \mid Q_1 \xrightarrow{\alpha} \Gamma \triangleright R$. This must have been derived in one of the following cases:
  - $\Gamma \triangleright P_1 \mid Q_1 \xrightarrow{\alpha} \Gamma \triangleright P_1' \mid Q_1$ since $\Gamma \triangleright P_1 \xrightarrow{\alpha} \Gamma \triangleright P_1'$. From $\Gamma \vdash P_1 \equiv_\sigma P_2$, we have that $\Gamma \triangleright P_2 \xrightarrow{\alpha} \Gamma \triangleright P_2'$ with $\Gamma \triangleright P_1' \equiv_\sigma P_2'$. By rule (PAR) we have $\Gamma \triangleright P_2 \mid Q_2 \xrightarrow{\alpha} \Gamma \triangleright P_2' \mid Q_2$ and we easily conclude $(\Gamma \triangleright P_1' \mid Q_1, \Gamma \triangleright P_2' \mid Q_2) \in S$ observing that $\Gamma \triangleright P_1' \in W(\equiv_\sigma)$ by persistence of $W(\equiv_\sigma)$.
  - $\Gamma \triangleright P_1 \mid Q_1 \xrightarrow{\alpha} \Gamma \triangleright P_1' \mid Q_1$ since $\Gamma \triangleright Q_1 \xrightarrow{\alpha} \Gamma \triangleright Q_1'$. This case is similar to the previous one.
  - $\Gamma \triangleright P_1 \mid Q_1 \xrightarrow{\tau} \Gamma \triangleright (\nu m : T)(P_1' \mid Q_1')$ since $\Gamma \triangleright P_1 \xrightarrow{(\nu m : T)\pi(m)} \Gamma \triangleright P_1'$ and $\Gamma \triangleright Q_1 \xrightarrow{(\nu m : T)\pi(m)} \Gamma \triangleright Q_1'$. Note that the persistence of $W(\equiv_\sigma)$ implies $\Gamma \triangleright P_1' \in W(\equiv_\sigma)$ and $\Gamma \triangleright Q_1' \in W(\equiv_\sigma)$. From $\Gamma \vdash P_1 \equiv_\sigma P_2$ and $\Gamma \vdash Q_1 \equiv_\sigma Q_2$, we have that $\Gamma \triangleright P_2 \xrightarrow{(\nu m : T)\pi(m)} \Gamma \triangleright P_2'$ with $\Gamma \vdash P_1' \equiv_\sigma P_2'$ and $\Gamma \triangleright Q_2 \xrightarrow{(\nu m : T)\pi(m)} \Gamma \triangleright Q_2'$. We then have $\Gamma \triangleright P_2 \mid Q_2 \xrightarrow{(\nu m : T)\pi(m)} \Gamma \triangleright P_2' \mid Q_2'$. This case is also similar to the previous one.
  - $\Gamma \triangleright P_1 \mid Q_1 \xrightarrow{\tau} \Gamma \triangleright (\nu m : T)(P_1' \mid Q_1')$. Then we have $\Gamma \triangleright P_2 \xrightarrow{(\nu m : T)\pi(m)} \Gamma \triangleright P_2'$ with $\Gamma \vdash P_1' \equiv_\sigma P_2'$ and $\Gamma \vdash Q_2 \equiv_\sigma Q_2$, we have that $\Gamma \triangleright P_2 \xrightarrow{(\nu m : T)\pi(m)} \Gamma \triangleright P_2'$ with $\Gamma \vdash P_1' \equiv_\sigma P_2'$ and $\Gamma \vdash Q_2 \equiv_\sigma Q_2$, we have that $\Gamma \triangleright P_2 \xrightarrow{(\nu m : T)\pi(m)} \Gamma \triangleright P_2'$ with $\Gamma \vdash P_1' \equiv_\sigma P_2'$ and $\Gamma \vdash Q_2 \equiv_\sigma Q_2$. We then have $\Gamma \triangleright P_2 \mid Q_2 \xrightarrow{(\nu m : T)\pi(m)} \Gamma \triangleright P_2' \mid Q_2'$. This case is also similar to the previous one.

- \( \Gamma' = \Gamma'', \mathsf{m}:T \) and \( \Gamma, \mathsf{m}:T \triangleright P \rightarrow \Gamma' \triangleright P' \). In this case we conclude using the hypothesis \( \Gamma, \mathsf{m}:T \triangleright P \in W(\equiv_\sigma) \).
- $\Gamma \vdash P_1 \mid Q_1 \xrightarrow{\alpha} \Gamma' \vdash R$ with $\alpha \in \{n(m), \pi(m)\}$, then we distinguish two cases:
  - $\Gamma \vdash P_1 \xrightarrow{\alpha} \Gamma \vdash P'_1$ and $R = P'_1 \mid Q_1$. From $\Gamma \vdash P_1 \in \mathcal{W}(\equiv\sigma)$ we have that $\Gamma \vdash P'_1 \in \mathcal{W}(\equiv\sigma)$ and there exists $P''_1$ such that $\Gamma \vdash P''_1 \equiv_{\sigma} P'_1$. Now, from $\Gamma \vdash P_1 \equiv_{\sigma} P_2$, we have $\Gamma \vdash P_2 \implies \Gamma \vdash P''_1$ with $\Gamma \vdash P''_1 \equiv_{\sigma} P'_1$ and $\Gamma \vdash P''_1 \in \mathcal{W}(\equiv\sigma)$. By rule (PAR) we have $\Gamma \vdash P_2 \mid Q_2 \implies \Gamma \vdash P''_1 \mid Q_2$ and we conclude observing that $(\Gamma \vdash P'_1 \mid Q_1, \Gamma \vdash P'_2 \mid Q_2) \in \mathcal{S}$.
  - $\Gamma \vdash Q_1 \xrightarrow{\alpha} \Gamma' \vdash Q'_1$ and $R = P_1 \mid Q'_1$. This case is analogous to the previous one.

- $\Gamma \vdash P_1 \mid Q_1 \xrightarrow{\alpha} \Gamma' \vdash R$ with $\alpha \in \{(\nu m : T) n(m), (\nu m : T) \pi(m)\}$, then we distinguish two cases:
  - $\Gamma \vdash P_1 \xrightarrow{\alpha} \Gamma' \vdash P'_1$ and $R = P'_1 \mid Q_1$ with $\Gamma'$, $m : T$ and $m \notin \text{fn}(Q_1)$. From $\Gamma \vdash P_1 \in \mathcal{W}(\equiv\sigma)$ we have that $\Gamma \vdash P'_1 \in \mathcal{W}(\equiv\sigma)$ and there exists $P''_1$ such that $\Gamma \vdash P''_1 \equiv_{\sigma} (\nu m : T) P'_1$. Now, from $\Gamma \vdash P_1 \equiv_{\sigma} P_2$, we have $\Gamma \vdash P_2 \implies \Gamma \vdash P''_1$ with $\Gamma \vdash P''_1 \equiv_{\sigma} (\nu m : T) P'_1$ and $\Gamma \vdash P''_1 \in \mathcal{W}(\equiv\sigma)$. By rule (PAR) we have $\Gamma \vdash P_2 \mid Q_2 \implies \Gamma \vdash P''_1 \mid Q_2$. In order to conclude we need to show $(\Gamma \vdash (\nu m : T)(P'_1 \mid Q_1), \Gamma \vdash P''_1 \mid Q_2) \in \mathcal{S}$ and $(\Gamma, m ; T \triangleright P'_1 \mid Q_1, \Gamma, m ; T \triangleright P'_1 \mid Q_1) \in \mathcal{S}$. The second fact follows easily, and the first one follows from $\Gamma \vdash (\nu m : T)(P'_1 \mid Q_1) \equiv (\nu m : T) P'_1 \mid Q_1$ since $m \notin \text{fn}(Q_1)$.
  - $\Gamma \vdash Q_1 \xrightarrow{\alpha} \Gamma' \vdash Q'_1$ and $R = P_1 \mid Q'_1$. This case is analogous to the previous one.

- the symmetric cases, where $P_2 \mid Q_2$ moves are similar to the previous one.

Lemma 6. Let $P_i, P_2, Q_1, Q_2$ be processes and $\Gamma$ a type environment such that $\Gamma \vdash P_i, Q_i$ for $i = 1, 2$. Let be $\Gamma \vdash Q_1 \equiv_{\sigma} Q_2$, $\Gamma \vdash P_1 \equiv_{\sigma} P_2$, $\Gamma \triangleright P_i \in \mathcal{W}(\equiv\sigma)$ for $i = 1, 2$, then $\Gamma \vdash P_1 \mid Q_1 \equiv_{\sigma} P_2 \mid Q_2$.

Proof. Consider the following relation $\mathcal{S}$:

$$
\mathcal{S} = \{ (\Gamma \triangleright P_1 \mid Q_1, \Gamma \triangleright P_2 \mid Q_2) \mid \Gamma \vdash Q_1 \equiv_{\sigma} Q_2, \Gamma \vdash P_1 \equiv_{\sigma} P_2, \\
\Gamma \triangleright P_i \in \mathcal{W}(\equiv\sigma) \ i = 1, 2 \}
$$

it is sufficient to show that $\mathcal{S}$ is a bisimulation on $\sigma$-actions, which is not difficult.

Proposition 13. Let $P$ be a process and $\Gamma$ be a type environment such that $\Gamma \vdash P$. If $\Gamma \triangleright P \in \mathcal{W}(\equiv\sigma)$ then $\Gamma \triangleright P \in \mathcal{W}(\equiv\sigma)$.

Proof. Consider the following relation $\mathcal{S}$:

$$
\mathcal{S} = \{ (\Gamma \triangleright P \mid Q_1, \Gamma \triangleright P \mid Q_2) \mid \Gamma \triangleright P, \Gamma \triangleright Q_i \in \mathcal{W}(\equiv\sigma) \ i = 1, 2 \ \Gamma \vdash Q_1 \equiv_{\sigma} Q_2 \}
$$

it is sufficient to prove that $\mathcal{S}$ is a partial bisimulation up to $\sigma$-high actions up to restriction and structural congruence $\equiv$. This implies $\mathcal{S} \subseteq \approx_{\sigma}$, then $\Gamma \vdash P \approx_{\sigma} P'$, and we conclude by Theorem 4. The rest of the proof follows similarly to the proof of Proposition 12, using Lemma 6.
Theorem 7 (Compositionality of $\mathcal{W}(\approx_\sigma)$). Let $\sigma \in \Sigma$, $P$ and $Q$ be two processes and $\Gamma$ be a type environment such that $\Gamma \vdash P, Q$. If $\Gamma \vdash P \in \mathcal{W}(\approx_\sigma)$ and $\Gamma \vdash Q \in \mathcal{W}(\approx_\sigma)$ then

1. $\Gamma, \Gamma' \vdash \pi(b). P \in \mathcal{W}(\approx_\sigma)$ where $\Gamma, \Gamma' \vdash a : \delta[T]$ and $\delta \leq \sigma$;
2. $\Gamma, \Gamma' \vdash a(x : T). P \in \mathcal{W}(\approx_\sigma)$ where $\Gamma, \Gamma' \vdash a : \delta[T]$ and $\delta \leq \sigma$;
3. $\Gamma, \Gamma' \vdash a = b$ then $P$ else $Q \in \mathcal{W}(\approx_\sigma)$ where $\Gamma, \Gamma' \vdash a : T$ and $\Gamma, \Gamma' \vdash b : T$;
4. $\Gamma \vdash P \mid Q \in \mathcal{W}(\approx_\sigma)$;
5. $\Gamma \vdash (\nu m : T) P \in \mathcal{W}(\approx_\sigma)$ where $\Gamma = \Gamma', n : T$;
6. $\Gamma \vdash P \in \mathcal{W}(\approx_\sigma)$.

Proof. The cases (1), (2), (3) are immediate; the remaining cases comes by Proposition 12, Proposition 11 and Proposition 13.

B.2 Proofs about $\approx_\sigma$

Lemma 7 (Persistence of $\approx_\sigma$). If $\Gamma \vdash P \approx_\sigma P$, then for all $\Gamma' \vdash P'$ such that $\Gamma \vdash P \rightsquigarrow \Gamma' \vdash P'$, it holds $\Gamma' \vdash P' \approx_\sigma P'$.

Proof. By induction on the length of the derivation $\Gamma \vdash P \rightsquigarrow \Gamma' \vdash P'$. The base case, where the length is 0 is immediate. For the inductive case, assume $\Gamma \vdash P \rightsquigarrow \Gamma_1 \vdash P_1 \rightsquigarrow \Gamma' \vdash P'$; we distinguish the following cases:

- $\Gamma_1 \vdash P_1 \xrightarrow{\alpha} \Gamma' \vdash P'$. By induction we know $\Gamma_1 \vdash P_1 \approx_\sigma P_1$, then by definition of $\approx_\sigma$ we have $\Gamma_1 \vdash P_1 \longrightarrow \Gamma' \vdash P'$ and $\Gamma' \vdash P' \approx_\sigma P_2$. Then by symmetry of $\approx_\sigma$ we also have $\Gamma_1 \vdash P_2 \approx_\sigma P'$, and by transitivity $\Gamma_1 \vdash P' \approx_\sigma P'$.

- $\Gamma_1 \vdash P_1 \xrightarrow{\alpha} \Gamma' \vdash P'$ with $\alpha \in \{n(m), \pi(m)\}$ and $\Gamma' = \Gamma$. By induction we know $\Gamma_1 \vdash P_1 \approx_\sigma P_1$, then by definition of $\approx_\sigma$ we have $\Gamma_1 \vdash P_1 \longrightarrow \Gamma_1 \vdash P_2$ and $\Gamma_1 \vdash P' \approx_\sigma P_2$. Then by symmetry of $\approx_\sigma$ we also have $\Gamma_1 \vdash P_2 \approx_\sigma P'$, and by transitivity $\Gamma_1 \vdash P' \approx_\sigma P'$.

- $\Gamma_1 \vdash P_1 \xrightarrow{\alpha} \Gamma' \vdash P'$ with $\alpha \in \{\nu m : T, n(m), \nu m : T, \pi(m)\}$ and $\Gamma' = \Gamma, m : T$. By induction we know $\Gamma_1 \vdash P_1 \approx_\sigma P_1$, then by definition of $\approx_\sigma$ we have $\Gamma_1 \vdash P_1 \longrightarrow \Gamma_1 \vdash P_2$, $\Gamma_1 \vdash P_2 \approx_\sigma (\nu m : T) P'$ and $\Gamma' \vdash P' \approx_\sigma P'$ as desired.

Lemma 8. If $\Gamma \vdash P \approx_\sigma Q$ then $\Gamma \vdash P \approx_\sigma Q$.

Proof of Theorem 4 Let $\sigma \in \Sigma$, $P$ be a process and $\Gamma$ a type environment such that $\Gamma \vdash P$. Then $\Gamma \vdash P \in \mathcal{W}(\approx_\sigma)$ if and only if $\Gamma \vdash P \approx_\sigma P$.

Proof. By Theorem 2 it is sufficient to prove $\Gamma \vdash P \in \mathcal{W}(\approx_\sigma)$ iff $\Gamma \vdash P \approx_\sigma P$.

($\Rightarrow$) From $\Gamma \vdash P \approx_\sigma P$, by Lemma 7, we have that $\forall \Gamma' \vdash P'$ such that $\Gamma \vdash P \rightsquigarrow \Gamma' \vdash P', \Gamma' \vdash P' \approx_\sigma P'$. Let then be $\Gamma \vdash P \rightsquigarrow \Gamma' \vdash P'$, we distinguish two cases that correspond to the definition of $\Gamma \vdash P \in \mathcal{W}(\approx_\sigma)$:

- $\Gamma' \vdash P' \xrightarrow{\alpha} \Gamma' \vdash P_2$ with $\alpha \in \{\pi(m), n(m)\}$. Then by definition of $\approx_\sigma$ there exists $P_3$ such that $\Gamma' \vdash P' \rightsquigarrow \Gamma' \vdash P_3$ and $\Gamma' \vdash P_3 \approx_\sigma P_2$. By Lemma 8 and Theorem 1, we conclude $\Gamma' \vdash P_3 \approx_\sigma P_2$. 

27
\[ \Gamma \triangleright P' \xrightarrow{\alpha} \sigma \quad \Gamma', m : T \triangleright P_2 \text{ with } \alpha \in \{(\nu m : T \overline{\pi}(m)), (\nu m : T) n(m)\}. \]

Then by definition of \( \tilde{\approx}_\sigma \) there exists \( P_3 \) such that \( \Gamma \triangleright P' \implies \Gamma' \triangleright P_3 \) and \( \Gamma' \vdash P_3 \approx_{\sigma} (\nu m : T) P_2 \). By Lemma 8 and Theorem 1, we conclude \( \Gamma' \vdash P_3 \approx_{\sigma} (\nu m : T) P_2 \).

\( (\Rightarrow) \) Consider the following binary relation:

\[ S = \{(\Gamma \triangleright P, \Gamma \triangleright Q) \mid \Gamma \triangleright P \in \mathcal{W}(\approx_{\sigma}), \Gamma \triangleright Q \in \mathcal{W}(\approx_{\sigma}) \text{ and } \Gamma \vdash P \approx_{\sigma} Q \} \]

it is sufficient to prove that \( S \) is a partial bisimulation up to \( \sigma \)-high actions. Let distinguish three cases:

\(- \Gamma \triangleright P \xrightarrow{\alpha}_{\sigma} \quad \Gamma' \triangleright P'. \) From the hypothesis \( \Gamma \vdash P \approx_{\sigma} Q \) we have that there exists \( Q' \) such that \( \Gamma \triangleright Q \xrightarrow{\alpha}_{\sigma} \quad \Gamma' \triangleright Q' \) with \( \Gamma' \vdash P' \approx_{\sigma} Q' \). By Proposition 5 we have \( \Gamma' \vdash P' \in \mathcal{W}(\approx_{\sigma}) \) and \( \Gamma' \triangleright Q' \in \mathcal{W}(\approx_{\sigma}) \), hence, by definition of \( S \), \( (\Gamma' \triangleright P', \Gamma' \triangleright Q') \in S \) as desired.

\(- \Gamma \triangleright P \xrightarrow{\alpha}_{\sigma} \Gamma \triangleright P' \) with \( \alpha \in \{\overline{\pi}(m), n(m)\} \). From the hypothesis \( \Gamma \triangleright P \in \mathcal{W}(\approx_{\sigma}) \), we have that there exists \( P'' \) such that \( \Gamma \triangleright P \implies \Gamma \triangleright P'' \) and \( \Gamma \vdash P'' \approx_{\sigma} P'' \) by Theorem 1. Now, from \( \Gamma \vdash P \approx_{\sigma} Q \), we have that there exists \( Q' \) such that \( \Gamma \triangleright Q \implies \Gamma \triangleright Q' \) and \( \Gamma \vdash Q' \approx_{\sigma} P'' \), then also \( \Gamma \vdash P' \approx_{\sigma} Q' \). By Proposition 5 we have \( \Gamma \triangleright P' \in \mathcal{W}(\approx_{\sigma}) \) and \( \Gamma \triangleright Q' \in \mathcal{W}(\approx_{\sigma}) \), hence, by definition of \( S \), \( (\Gamma \triangleright P', \Gamma \triangleright Q') \in S \) as desired.

\(- \Gamma \triangleright P \xrightarrow{\alpha}_{\sigma} \Gamma, m : T \triangleright P' \) with \( \alpha \in \{(\nu m : T) \overline{\pi}(m), (\nu m : T) n(m)\} \). From the hypothesis \( \Gamma \triangleright P \in \mathcal{W}(\approx_{\sigma}) \), we have that there exists \( P'' \) such that \( \Gamma \triangleright P \implies \Gamma \triangleright P'' \) and \( \Gamma \vdash (\nu m : T) P'' \approx_{\sigma} P'' \) by Theorem 1. Now, from \( \Gamma \vdash P \approx_{\sigma} Q \), we have that there exists \( Q' \) such that \( \Gamma \triangleright Q \implies \Gamma \triangleright Q' \) and \( \Gamma \vdash Q' \approx_{\sigma} P'' \), then also \( \Gamma \vdash (\nu m : T) P'' \approx_{\sigma} Q' \). By Proposition 5 and Proposition 11, we have \( \Gamma \vdash (\nu m : T) P' \in \mathcal{W}(\approx_{\sigma}) \) and \( \Gamma \triangleright Q' \in \mathcal{W}(\approx_{\sigma}) \), hence, by definition of \( S \), \( (\Gamma \triangleright (\nu m : T) P', \Gamma \triangleright Q') \in S \).

To conclude we also need \( (\Gamma, m : T \triangleright P', \Gamma, m : T \triangleright P') \in S \), which comes from \( \Gamma, m : T \triangleright P' \in \mathcal{W}(\approx_{\sigma}) \) and \( \Gamma, m : T \vdash P' \approx_{\sigma} P' \).

B.3 Proofs about \( \tilde{\approx}_{\sigma} \)

Lemma 9. Let \( P \) be a process and \( \Gamma \) a type environment such that \( \Gamma \vdash P \).

1. \( \Gamma \triangleright (\nu^\sigma) P \xrightarrow{\alpha}_{\sigma} \), that is \( \Gamma \triangleright (\nu^\sigma) P \) can only perform \( \xrightarrow{\alpha}_{\sigma} \) transitions.
2. for all \( \Gamma' \triangleright Q \) such that \( \Gamma \triangleright (\nu^\sigma) P \rightsquigarrow \Gamma' \triangleright Q \), \( Q \) is of the form \( (\nu^\sigma) P' \) for some process \( P' \), and \( \Gamma' \triangleright (\nu^\sigma) P' \xrightarrow{\alpha}_{\sigma} \).
3. \( \Gamma \triangleright (\nu^\sigma) P \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright (\nu^\sigma) P' \) iff \( \Gamma \triangleright P \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright P' \).

Proof. The proof follows by induction on the length of the derivation \( \Gamma \triangleright (\nu^\sigma) P \rightsquigarrow \Gamma' \triangleright Q \) and the definition of the typed LTS given in Table 4.

Lemma 10. If \( \Gamma \vdash P \approx_{\sigma} (\nu^\sigma) Q \) then \( \Gamma \vdash (\nu^\sigma) P \approx_{\sigma} (\nu^\sigma) Q \).
Proof. Consider the following binary relation:

\[ S = \{ (\Gamma \triangleright (\nu^\sigma)P, \, \Gamma \triangleright (\nu^\sigma)Q) \mid \Gamma \models P \approx_{\sigma} (\nu^\sigma)Q \} \]

it is sufficient to prove that \( S \) is a bisimulation up to \( \sigma \)-high actions, that is \( S \subseteq \approx_{\sigma} \). This follows from the following cases:

- \( \Gamma \triangleright (\nu^\sigma)P \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright (\nu^\sigma)P' \). By Lemma 9, \( \Gamma \triangleright P \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright P' \). By the hypothesis that \( \Gamma \models P \approx_{\sigma} (\nu^\sigma)Q \), we have \( \Gamma \triangleright (\nu^\sigma)Q \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright (\nu^\sigma)Q' \) and \( \Gamma' \models P' \approx_{\sigma} (\nu^\sigma)Q' \). Hence, \( (\Gamma' \triangleright (\nu^\sigma)P', \, \Gamma' \triangleright (\nu^\sigma)Q') \in S \) as desired.
- \( \Gamma \triangleright (\nu^\sigma)Q \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright (\nu^\sigma)Q' \). By the hypothesis that \( \Gamma \models P \approx_{\sigma} (\nu^\sigma)Q \), we have \( \Gamma \triangleright P \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright P' \) and \( \Gamma \models P \approx_{\sigma} (\nu^\sigma)Q \). By Lemma 9, \( \Gamma \triangleright (\nu^\sigma)P \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright (\nu^\sigma)P' \) hence, \( (\Gamma' \triangleright (\nu^\sigma)P', \, \Gamma' \triangleright (\nu^\sigma)Q') \in S \) as desired.
- \( \Gamma \triangleright (\nu^\sigma)P, \, \Gamma \triangleright (\nu^\sigma)Q \) is vacuous by Lemma 9.

Lemma 11 (Persistence of \( \approx_{\sigma} \)). If \( \Gamma \models P \approx_{\sigma} (\nu^\sigma)P \), then for all \( \Gamma' \triangleright P' \) such that \( \Gamma \triangleright P \rightarrow \Gamma' \triangleright P' \), it holds \( \Gamma' \models P' \approx_{\sigma} (\nu^\sigma)P' \).

Proof. By induction on the length of the derivation \( \Gamma \triangleright P \rightarrow \Gamma' \triangleright P' \). The base case, where the length is 0 is immediate. For the inductive case, assume \( \Gamma \triangleright P \rightarrow \Gamma_1 \triangleright P_1 \xrightarrow{\alpha}_{\delta} \Gamma' \triangleright P' \); we distinguish the following cases:

- \( \Gamma_1 \triangleright P_1 \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright P' \). By induction we know \( \Gamma_1 \models P_1 \approx_{\sigma} (\nu^\sigma)P_1 \), then by definition of \( \approx_{\sigma} \) and Lemma 9, we have \( \Gamma_1 \triangleright (\nu^\sigma)P_1 \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright (\nu^\sigma)P_2 \) and \( \Gamma' \models P' \approx_{\sigma} (\nu^\sigma)P_2 \). By Lemma 10, \( \Gamma' \models (\nu^\sigma)P' \approx_{\sigma} (\nu^\sigma)P_2 \) and then, by transitivity of \( \approx_{\sigma} \), \( \Gamma' \models P' \approx_{\sigma} (\nu^\sigma)P' \) as desired.
- \( \Gamma_1 \triangleright P_1 \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright P' \) with \( \alpha \in \{ n(m) \}. \) By induction we know \( \Gamma_1 \models P_1 \approx_{\sigma} (\nu^\sigma)P_1 \), then by definition of \( \approx_{\sigma} \) and Lemma 9, we have \( \Gamma_1 \triangleright (\nu^\sigma)P_1 \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright (\nu^\sigma)P_2 \) and \( \Gamma' \models P' \approx_{\sigma} (\nu^\sigma)P_2 \). By Lemma 10, \( \Gamma' \models (\nu^\sigma)P' \approx_{\sigma} (\nu^\sigma)P_2 \) and then, by transitivity of \( \approx_{\sigma} \), \( \Gamma' \models P' \approx_{\sigma} (\nu^\sigma)P' \) as desired.
- \( \Gamma_1 \triangleright P_1 \xrightarrow{\alpha}_{\sigma} \Gamma' \triangleright P' \) with \( \alpha \in \{ (\nu m : T) n(m), (\nu m : T) \pi(m) \} \) and \( \Gamma' = \Gamma_1, m : T \). By induction we know \( \Gamma_1 \models P_1 \approx_{\sigma} (\nu^\sigma)P_1 \), then by definition of \( \approx_{\sigma} \) and Lemma 9, we have \( \Gamma_1 \triangleright (\nu^\sigma)P_1 \xrightarrow{\alpha}_{\sigma} \Gamma_1 \triangleright (\nu^\sigma)P_2 \) and \( \Gamma_1 \triangleright (\nu m : T) P' \approx_{\sigma} (\nu^\sigma)P_2 \) and \( \Gamma' \models P' \approx_{\sigma} (\nu^\sigma)P' \) as desired.

Lemma 12. Let \( P \) and \( Q \) be two processes and \( \Gamma \) a type environment such that \( \Gamma \models P, \, Q \). If \( \Gamma \models (\nu^\sigma)P \approx_{\sigma} (\nu^\sigma)Q \) then \( \Gamma \models P \approx_{\sigma} Q \).

Proof. Consider the following binary relation:

\[ S = \{ (\Gamma \triangleright P, \, \Gamma \triangleright Q) \mid \Gamma \models (\nu^\sigma)P \approx_{\sigma} (\nu^\sigma)Q \} \]

it is sufficient to prove that \( S \) is a bisimulation up to \( \sigma \)-actions, that is \( S \subseteq \approx_{\sigma} \). Let us distinguish the following cases:
– $\Gamma \triangleright P \xrightarrow{\alpha} \Gamma' \triangleright P'$. By Lemma 9, $\Gamma \triangleright (\nu^\sigma)P \xrightarrow{\alpha} \Gamma' \triangleright (\nu^\sigma)P'$. By the hypothesis that $\Gamma \triangleright (\nu^\sigma)P \equiv_\sigma (\nu^\sigma)Q$, we have $\Gamma \triangleright (\nu^\sigma)Q \xrightarrow{\sigma} \Gamma' \triangleright (\nu^\sigma)Q'$ and $\Gamma' \triangleright (\nu^\sigma)P' \equiv_\sigma (\nu^\sigma)Q'$. By Lemma 9, $\Gamma \triangleright Q \xrightarrow{\sigma} \Gamma' \triangleright Q'$ and then $(\Gamma' \triangleright P', \Gamma' \triangleright Q') \in S$ as desired.
– The symmetric case is analogous.

**Lemma 13.** Let $P$ and $Q$ be two processes and $\Gamma$ a type environment such that $\Gamma, m : T \vdash P, Q$. If $\Gamma, m : T \vdash (\nu^\sigma)P \equiv_\sigma Q$ then $\Gamma \vdash (\nu m : T)(\nu^\sigma)P \equiv_\sigma (\nu m : T)Q$.

**Proof.** Consider the following binary relation:

$S = \{(\Gamma \triangleright (\nu m : T)(\nu^\sigma)P), \Gamma \triangleright (\nu m : T)Q) \mid \Gamma, m : T \vdash (\nu^\sigma)P \equiv_\sigma Q\} \cup \approx_\sigma$

it is sufficient to prove that $S$ is a bisimulation up to $\sigma$-high actions and up to $\equiv$, where $\equiv$ is the standard structural congruence relation for the $\pi$-calculus (that we need here just to change the order of restricted names). Let us distinguish the following cases:

– $\Gamma \triangleright (\nu m : T)(\nu^\sigma)P \xrightarrow{\alpha} \Gamma' \triangleright P'$. This must have been derived in one of the following ways:

  • by (RES) from $\Gamma, m : T \triangleright (\nu^\sigma)P \xrightarrow{\alpha} P'$, $\Gamma, m : T \triangleright P'$ with $P' = (\nu m : T)P'$ and $m \notin \text{fn}(\alpha) \cup \text{bn}(\alpha)$. By Lemma 9 we know that $P' = (\nu^\sigma)P''$. By the hypothesis that $\Gamma, m : T \vdash (\nu^\sigma)P \equiv_\sigma Q$ there exists $Q'$ such that $\Gamma, m : T \vdash Q \xrightarrow{\sigma} \Gamma', m : T \vdash Q'$ and $\Gamma', m : T \vdash (\nu^\sigma)P' \equiv_\sigma Q'$. Hence, by (RES), $\Gamma \triangleright (\nu m : T)Q \xrightarrow{\sigma} \Gamma \triangleright (\nu m : T)Q'$ and $(\Gamma \triangleright (\nu^\sigma)P), \Gamma \triangleright (\nu m : T)Q' \in S$ as desired.

  • by (OPEN) and Lemma 9 from $\Gamma, m : T \triangleright (\nu^\sigma)P \xrightarrow{\pi(m)} \Gamma, m : T \triangleright (\nu^\sigma)P'$ and $\alpha = (\nu m : T)\pi(m)$. By the hypothesis that $\Gamma, m : T \vdash (\nu^\sigma)P \equiv_\sigma Q$ there exists $Q'$ such that $\Gamma, m : T \vdash Q \xrightarrow{\sigma} \Gamma, m : T \vdash Q'$ and $\Gamma, m : T \vdash (\nu^\sigma)P' \equiv_\sigma Q'$. Hence, $\Gamma \triangleright (\nu m : T)Q \xrightarrow{\sigma} \Gamma \triangleright (\nu m : T)Q'$ and $(\Gamma, m : T) \triangleright (\nu^\sigma)P', \Gamma, m : T \triangleright Q' \in S$ as desired.

– $\Gamma \triangleright (\nu m : T)(\nu^\sigma)P \xrightarrow{\alpha} \Gamma' \triangleright P'$. This case is vacuous by Lemma 9.

– $\Gamma \triangleright (\nu m : T)Q \xrightarrow{\alpha} \Gamma' \triangleright Q'$. This must have been derived in one of the following ways:

  • by (RES) from $\Gamma, m : T \triangleright Q \xrightarrow{\alpha} \Gamma', m : T \triangleright Q'$ with $Q' = (\nu m : T)Q'$ and $m \notin \text{fn}(\alpha) \cup \text{bn}(\alpha)$. By the hypothesis that $\Gamma, m : T \vdash (\nu^\sigma)P \equiv_\sigma Q$ and Lemma 9, there exists $P'$ such that $\Gamma, m : T \triangleright (\nu^\sigma)P \xrightarrow{\sigma} \Gamma', m : T \triangleright (\nu^\sigma)P'$ and $\Gamma', m : T \vdash (\nu^\sigma)P' \equiv_\sigma Q'$. Hence, $\Gamma \triangleright (\nu m : T)(\nu^\sigma)P \xrightarrow{\sigma} \Gamma' \triangleright (\nu m : T)(\nu^\sigma)P'$ and then $(\Gamma \triangleright (\nu^\sigma)P), \Gamma \triangleright (\nu m : T)Q' \in S$ as desired.

  • by (OPEN) from $\alpha = (\nu m : T)\pi(m)$ and $\Gamma, m : T \triangleright Q \xrightarrow{\pi(m)} \Gamma, m : T \triangleright Q'$. By the hypothesis that $\Gamma, m : T \vdash (\nu^\sigma)P \equiv_\sigma Q$ and Lemma 9, there exists $P'$ such that $\Gamma, m : T \triangleright (\nu^\sigma)P \xrightarrow{\pi(m)} \Gamma, m : T \triangleright (\nu^\sigma)P'$ and $\Gamma, m : T \vdash (\nu^\sigma)P' \equiv_\sigma Q'$. Hence, $\Gamma \triangleright (\nu m : T)(\nu^\sigma)P \xrightarrow{\pi(m)} \Gamma' \triangleright (\nu m : T)(\nu^\sigma)P'$ and then $(\Gamma, m : T) \triangleright (\nu^\sigma)P', \Gamma, m : T \triangleright Q' \in S$ as desired.
- $\Gamma \vdash (\nu \cdot T)Q \overset{\alpha}{\longrightarrow} \Gamma' \vdash Q''$. This must have been derived in one of the following ways:
  - $\alpha \in \{\pi(b), a(b), \tau\}$, then it must have been derived by (RES) from $\Gamma, m : T \vdash Q \overset{\alpha}{\longrightarrow} \Gamma, m : T \vdash Q'$ with $m \notin \text{fn}(a) \cup \text{bn}(a)$, $\Gamma' = \Gamma$ and $Q'' = (\nu \cdot T)Q'$. By the hypothesis that $\Gamma, m : T \vdash (\nu^\sigma)^P \equiv_\sigma Q$ and Lemma 9, there exists $P'$ such that $\Gamma, m : T \vdash (\nu^\sigma)^P \Longrightarrow \Gamma, m : T \vdash (\nu^\sigma)^P'$ and $\Gamma, m : T \vdash (\nu^\sigma)^P \equiv_\sigma Q'$. Hence $\Gamma \vdash (\nu \cdot T)(\nu^\sigma)^P \Longrightarrow \Gamma \vdash (\nu \cdot T)(\nu^\sigma)^P'$ and then $(\Gamma \vdash (\nu \cdot T)(\nu^\sigma)^P'), \Gamma \vdash (\nu \cdot T)Q') \in S$ as desired.
  - $\alpha \in \{\nu \cdot S \mid \pi(b), (\nu \cdot S) a(b)\}$ and one of the following two subcases:
    * $b \neq m$, then it must have been derived by (RES) from $\Gamma, m : T \vdash Q \overset{\pi}{\longrightarrow} \Gamma', m : T \vdash Q'$ with $\Gamma' = \Gamma, b : S, Q'' = (\nu \cdot T)Q'$. By the hypothesis that $\Gamma, m : T \vdash (\nu^\sigma)^P \equiv_\sigma Q$ and Lemma 9, there exists $P'$ such that $\Gamma, m : T \vdash (\nu^\sigma)^P \Longrightarrow \Gamma, m : T \vdash (\nu^\sigma)^P'$ and $\Gamma, m : T \vdash (\nu^\sigma)^P \equiv_\sigma (\nu \cdot S)Q'$. By the hypothesis that $\Gamma, m : T \vdash (\nu^\sigma)^P \Longrightarrow \Gamma, m : T \vdash (\nu^\sigma)^P'$ and then $(\Gamma \vdash (\nu \cdot T)(\nu^\sigma)^P'), \Gamma \vdash (\nu \cdot T)Q') \in S$ as desired.
    * $b = m$, then it must have been derived by (OPEN) from $\Gamma, m : T \vdash Q \overset{\pi}{\longrightarrow} \Gamma', m : T \vdash Q''$. By the hypothesis that $\Gamma, m : T \vdash (\nu^\sigma)^P \equiv_\sigma Q$ and Lemma 9, there exists $P'$ such that $\Gamma, m : T \vdash (\nu^\sigma)^P \Longrightarrow \Gamma, m : T \vdash (\nu^\sigma)^P'$ and $\Gamma, m : T \vdash (\nu^\sigma)^P \equiv_\sigma Q''$. By the hypothesis that $\Gamma, m : T \vdash (\nu^\sigma)^P \Longrightarrow \Gamma, m : T \vdash (\nu^\sigma)^P'$ and then $(\Gamma \vdash (\nu \cdot T)(\nu^\sigma)^P'), \Gamma \vdash (\nu \cdot T)Q') \in S$ as desired.

**Proof of Theorem 5** Let $\sigma \in \Sigma$, $P$ be a process and $\Gamma$ a type environment such that $\Gamma \vdash P$. Then $\Gamma \vdash P \in NT(\equiv_\sigma)$ if and only if $\Gamma \vdash P \equiv_\sigma (\nu^\sigma)^P$.

Proof. By Theorem 2 it is sufficient to show $\Gamma \vdash P \in W(\equiv_\sigma)$ iff $\Gamma \vdash P \equiv_\sigma (\nu^\sigma)^P$.

(\Rightarrow) From $\Gamma \vdash P \equiv_\sigma (\nu^\sigma)^P$, by Lemma 11, we have that $\forall \Gamma' \vdash P'$ such that $\Gamma' \vdash P' \vdash \Gamma' \vdash P'$, $\Gamma' \vdash P' \approx_\sigma (\nu^\sigma)^P$. Let then be $\Gamma, m : T \vdash P \in W(\equiv_\sigma)$.

- $\Gamma' \vdash P' \overset{\alpha}{\longrightarrow} \Gamma'' \vdash P_2$ with $\alpha \in \{\pi(m), n(m)\}$. Then by definition of $\equiv_\sigma$ and Lemma 9 there exists $P_3$ such that $\Gamma'' \vdash (\nu^\sigma)^P_3 \Longrightarrow \Gamma' \vdash (\nu^\sigma)^P_3$ and $\Gamma'' \vdash (\nu^\sigma)^P_3 \equiv_\sigma P_2$. Hence, $\Gamma' \vdash P' \Longrightarrow \Gamma' \vdash P_3$. By persistence of $\equiv_\sigma$, $\Gamma'' \vdash (\nu^\sigma)^P_3 \equiv_\sigma P_2$ and then, by transitivity of $\equiv_\sigma$, $\Gamma' \vdash (\nu^\sigma)^P_2 \approx_\sigma (\nu^\sigma)^P_3$. By Lemma 12, $\Gamma' \vdash P_3 \approx_\sigma P_2$, i.e., the thesis.

- $\Gamma' \vdash P' \overset{\pi}{\longrightarrow} \Gamma', m : T \vdash P_2$ with $\alpha \in \{\nu \cdot T \pi(m), (\nu \cdot T) n(m)\}$. Then by definition of $\equiv_\sigma$ and Lemma 9 there exists $P_3$ such that $\Gamma' \vdash (\nu^\sigma)^P_3 \Longrightarrow \Gamma' \vdash (\nu^\sigma)^P_3$ and $\Gamma' \vdash (\nu^\sigma)^P_3 \equiv_\sigma P_2$. Hence, $\Gamma' \vdash P' \Longrightarrow \Gamma' \vdash P_3$. By persistence of $\equiv_\sigma$, $\Gamma', m : T \vdash (\nu^\sigma)^P_2 \approx_\sigma P_2$ and, by Lemma 13 and the fact that $\equiv \approx \equiv_\sigma$, $\Gamma' \vdash (\nu^\sigma)^P_2 \approx_\sigma (\nu \cdot T)^P_2$. By transitivity of $\equiv_\sigma$,
\( \Gamma' \vdash (\nu^\sigma)(\nu m:T)P_2 \approx_\sigma (\nu^\sigma)P_3 \). By Lemma 12, \( \Gamma' \vdash P_3 \approx_\sigma (\nu m:T)P_2 \), i.e., the thesis.

\((\Rightarrow)\) Consider the following binary relation:

\[ S = \{ (\Gamma \triangleright (\nu^\sigma)P, \Gamma \triangleright Q) \mid \Gamma \triangleright Q \in \mathcal{W}(\equiv_\sigma) \text{ and } \Gamma \vdash P \approx_\sigma Q \} \]

it is sufficient to prove that \( S \) is a bisimulation up to \( \sigma \)-high actions, that is \( S \subseteq \approx_\sigma \). Let us distinguish the following cases:

- \( \Gamma \triangleright (\nu^\sigma)P \xrightarrow[\sigma]{\alpha} \Gamma' \triangleright P'' \). By Lemma 9 \( P'' = (\nu^\sigma)P' \) and \( \Gamma \triangleright P \xrightarrow[\sigma]{\alpha} \Gamma' \triangleright P' \). From the hypothesis \( \Gamma \vdash P \approx_\sigma Q \) we have that there exists \( Q' \) such that \( \Gamma \triangleright Q \xrightarrow[\sigma]{\alpha} \Gamma' \triangleright Q' \) with \( \Gamma' \vdash P' \approx_\sigma Q' \). By Proposition 5 we have \( \Gamma' \triangleright Q' \in \mathcal{W}(\equiv_\sigma) \), hence, by definition of \( S \), \( (\Gamma' \triangleright (\nu^\sigma)P', \Gamma' \triangleright Q') \in S \) as desired.

- \( \Gamma \triangleright (\nu^\sigma)P \xrightarrow[\sigma]{\alpha} \Gamma' \triangleright P' \) is a vacuous case by Lemma 9.

- \( \Gamma \triangleright Q \xrightarrow[\sigma]{\alpha} \Gamma' \triangleright Q' \). From the hypothesis \( \Gamma \vdash P \approx_\sigma Q \) we have that there exists \( P' \) such that \( \Gamma \triangleright P \xrightarrow[\sigma]{\alpha} \Gamma' \triangleright P' \) with \( \Gamma' \vdash P' \approx_\sigma Q' \). Then, since \( \alpha \) is an action of level not greater than \( \sigma \), by rule (RES), \( \Gamma \triangleright (\nu^\sigma)P \xrightarrow[\sigma]{\alpha} \Gamma' \triangleright (\nu^\sigma)P' \). By Proposition 5 we have \( \Gamma' \triangleright Q' \in \mathcal{W}(\equiv_\sigma) \), hence, by definition of \( S \), \( (\Gamma' \triangleright (\nu^\sigma)P', \Gamma' \triangleright Q') \in S \) as desired.

- \( \Gamma \triangleright Q \xrightarrow[\sigma]{\alpha} \Gamma \triangleright Q' \) with \( \alpha \in \{ \pi_m, n(m) \} \). From the hypothesis \( \Gamma \triangleright Q \in \mathcal{W}(\equiv_\sigma) \), we have that there exists \( Q'' \) such that \( \Gamma \triangleright Q \Longrightarrow \Gamma \triangleright Q'' \) and \( \Gamma \vdash Q' \approx_\sigma Q'' \), hence \( \Gamma \vdash Q' \approx_\sigma Q'' \) by Theorem 1. Now, from \( \Gamma \vdash P \approx_\sigma Q' \), we have that there exists \( P' \) such that \( \Gamma \triangleright P \Longrightarrow \Gamma \triangleright P' \) and \( \Gamma \vdash P' \approx_\sigma Q'' \), then also \( \Gamma \vdash P' \approx_\sigma Q' \). Hence, by rule (RES) also \( \Gamma \triangleright (\nu^\sigma)P \Longrightarrow \Gamma \triangleright (\nu^\sigma)P' \). By Proposition 5 we have \( \Gamma \triangleright Q' \in \mathcal{W}(\equiv_\sigma) \), hence, by definition of \( S \), \( (\Gamma \triangleright (\nu^\sigma)P', \Gamma \triangleright Q') \in S \) as desired.

- \( \Gamma \triangleright Q \xrightarrow[\sigma]{\alpha} \Gamma, m:T \triangleright Q' \) with \( \alpha \in \{ (\nu m:T)\pi(n), (\nu m:T)n(m) \} \). From the hypothesis \( \Gamma \triangleright Q \in \mathcal{W}(\equiv_\sigma) \), we have that there exists \( Q'' \) such that \( \Gamma \triangleright Q \Longrightarrow \Gamma \triangleright Q'' \) and \( \Gamma \vdash (\nu m:T)Q' \approx_\sigma Q'' \), hence \( \Gamma \vdash (\nu m:T)Q' \approx_\sigma Q'' \) by Theorem 1. Now, from \( \Gamma \vdash P \approx_\sigma Q' \), we have that there exists \( P' \) such that \( \Gamma \triangleright P \Longrightarrow \Gamma \triangleright P' \) and \( \Gamma \vdash P' \approx_\sigma Q'' \), then also \( \Gamma \vdash (\nu m:T)Q' \approx_\sigma P' \). Hence, by rule (RES) also \( \Gamma \triangleright (\nu^\sigma)P \Longrightarrow \Gamma \triangleright (\nu^\sigma)P' \). By Proposition 5 and Proposition 11, we have \( \Gamma \triangleright (\nu m:T)Q' \in \mathcal{W}(\equiv_\sigma) \), hence, by definition of \( S \), \( (\Gamma \triangleright (\nu^\sigma)P', \Gamma \triangleright (\nu m:T)Q') \in S \). To conclude we also need \( (\Gamma, m:T \triangleright (\nu^\sigma)Q', \Gamma, m:T \triangleright Q') \in S \), which comes from \( \Gamma, m:T \triangleright Q' \in \mathcal{W}(\equiv_\sigma) \) and \( \Gamma, m:T \triangleright Q' \approx_\sigma Q' \).

**Lemma 14.** Let \( P, Q \) be processes such that \( \Gamma \vdash P, Q \). If \( \Gamma \vdash (\nu^\sigma)P \approx_\sigma (\nu^\sigma)Q \) then \( \Gamma \vdash P \approx_\sigma Q \).

**Proof.** The proof is similar to that of Lemma 12, and relies on Lemma 9.

**Proof of Proposition 6** Let \( P, Q \) be processes such that \( \Gamma \vdash P, Q \). \( \Gamma \vdash P \approx_\sigma Q \) if and only if \( \Gamma \vdash (\nu^\sigma)P \approx_\top (\nu^\sigma)Q \).
Proof. ($\Rightarrow$) By Proposition 7, from $\Gamma \vdash P \approx_\sigma Q$ we have $\Gamma \vdash (\nu^\sigma)P \approx_\sigma (\nu^\sigma)Q$, and we conclude observing that, by Lemma 9, $(\nu^\sigma)P,(\nu^\sigma)Q$ do never perform actions of level higher than $\sigma$.

($\Leftarrow$) From $\Gamma \vdash (\nu^\sigma)P \approx_\top (\nu^\sigma)Q$ we have $\Gamma \vdash (\nu^\sigma)P \approx_\sigma (\nu^\sigma)Q$ since $\approx_\top \subseteq \approx_\sigma$, and we conclude by Lemma 14.