Regular Expression Types for Strings in a Text Processing Language: Proofs of the Theorems (Draft)

Naoshi Tabuchi Eijiro Sumii Akinori Yonezawa
{tabee,sumii,yonezawa}@yl.is.s.u-tokyo.ac.jp
Department of Computer Science,
Graduate School of Information Science and Technology,
University of Tokyo

February 10, 2003

Abstract

We prove the theorems and lemmas stated in [2]. The most significant results among all are: (1) termination and determinacy of pattern matching, which guarantees that the pattern matching algorithm always yields just one result (thus, a program does never falls into an infinite loop during pattern matching), and (2) soundness and completeness of the typing rules for pattern matching, which guarantee that our type system infers all and only possible strings bound to the variables in a pattern.

1 Preliminaries

We repeat the formal definition of λ^{re} [2] in Figure 1–5, so that readers can refer to it when necessary. Throughout this paper, we assume that all patterns are *standardized* [1] beforehand. Standardization, in a word, "pushes away" the empty string ε outside of a pattern P. That is, if ε matches P, the standardized form of P becomes $\varepsilon \mid P^-$ where $novar(P^-) \equiv novar(P) \cap \bar{\varepsilon}$. For example, $\mathtt{a}^*$ is standardized to $\varepsilon \mid \mathtt{aa}^*$ and $(\mathtt{a}^*)^*$ to $\varepsilon \mid \mathtt{aa}^*((\mathtt{aa}^*)^*)$.

2 Determinacy and Termination of Pattern Matching

2.1 Overview of the Proof

Proving the termination property of pattern matching is not as simple as it may seem. A naïve approach such as induction on the structure of patterns does not work, because patterns are not structurally reduced in (M-Rep) and (M-Seq-Rep). Instead of proving the property directly, we introduce another matching algorithm that calculates *all possible matchings* along with the first/longest one. This idea owes to [3], with a minor change in that we require the pattern to be standardized in order to ensure termination. The new algorithm intuitively works as follows: it receives a set of pairs (s, θ) of a string and a substitution as the input, and returns another set of pairs (s', θ'). Here, each output string s' is constructed from some input string s by removing a prefix that matches the pattern (thus, $s = s''s'$ for some s'' that matches the pattern). θ' in the output is similarly constructed from some θ in the input by adding new substitutions for the variables that appear in the pattern.

$$
\begin{array}{llcll}
M & \text{(term)} & ::= & x & \text{(variable)} \\
 & & \mid & \texttt{fix}(f, x, M) & \text{(recursive function)} \\
 & & \mid & M_1 M_2 & \text{(function application)} \\
 & & \mid & s & \text{(constant string)} \\
 & & \mid & M_1 \mathbin{\hat{}} M_2 & \text{(string concatenation)} \\
 & & \mid & \texttt{match}\ M\ \texttt{with}\ P_1 \Rightarrow M_1 \mid \cdots \mid P_n \Rightarrow M_n & \text{(pattern matching)} \\
 & & \mid & \texttt{print}\ M & \text{(standard output)} \\
P & \text{(pattern)} & ::= & s & \text{(constant string)} \\
 & & \mid & P_1 \mid P_2 & \text{(choice)} \\
 & & \mid & x\ \texttt{as}\ P & \text{(variable binding)} \\
 & & \mid & P^* & \text{(repetition)} \\
 & & \mid & P_1 P_2 & \text{(sequence)}
\end{array}
$$

Figure 1: Syntax

$$
\begin{array}{rlcl}
v & \text{(value)} & ::= & \texttt{fix}(f, x, M) \\
 & & \mid & s \\
C[\,] & \text{(evaluation context)} & ::= & [\,]M \\
 & & \mid & v[\,] \\
 & & \mid & [\,] \mathbin{\hat{}} M \\
 & & \mid & v \mathbin{\hat{}} [\,] \\
 & & \mid & \texttt{match}\ [\,]\ \texttt{with}\ P_1 \Rightarrow M_1 \mid \cdots \mid P_n \Rightarrow M_n \\
 & & \mid & \texttt{print}\ [\,]
\end{array}
$$

$$
\frac{}{\texttt{fix}(f, x, M)v \xrightarrow{\varepsilon} [v/x][\texttt{fix}(f, x, M)/f]M}\text{(R-App)} \qquad \frac{}{s_1 \mathbin{\hat{}} s_2 \xrightarrow{\varepsilon} s_1 + s_2}\text{(R-Cat)}
$$

$$
\frac{}{\texttt{print}\ s \xrightarrow{s} \varepsilon}\text{(R-Print)} \qquad \frac{\forall i < m.\, s \rhd P_i \Rightarrow \bot \quad s \rhd P_m \Rightarrow \theta \quad \theta \neq \bot}{\texttt{match}\ s\ \texttt{with}\ P_1 \Rightarrow M_1 \mid \cdots \mid P_n \Rightarrow M_n \xrightarrow{\varepsilon} \theta M_m}\text{(R-Match)}
$$

$$
\frac{M_1 \xrightarrow{s} M_2}{C[M_1] \xrightarrow{s} C[M_2]}\text{(R-Ctx)}
$$

Figure 2: Operational Semantics of Terms

$$\frac{}{s \rhd s \Rightarrow \emptyset}\text{(M-Const-Succ)} \qquad \frac{s \neq s'}{s \rhd s' \Rightarrow \bot}\text{(M-Const-Fail)}$$

$$\frac{s \rhd P_1 \Rightarrow \theta \quad \theta \neq \bot}{s \rhd P_1 \mid P_2 \Rightarrow \theta}\text{(M-Choice-Fst)} \qquad \frac{s \rhd P_1 \Rightarrow \bot \quad s \rhd P_2 \Rightarrow \theta}{s \rhd P_1 \mid P_2 \Rightarrow \theta}\text{(M-Choice-Snd)}$$

$$\frac{s \rhd P \Rightarrow \theta}{s \rhd x \texttt{ as } P \Rightarrow \theta \uplus \{x \mapsto s\}}\text{(M-Bind)} \qquad \frac{s \rhd PP^* \mid \varepsilon \Rightarrow \theta}{s \rhd P^* \Rightarrow \theta}\text{(M-Rep)}$$

$$\frac{s_2^{-1}s_1 \rhd P \Rightarrow \theta}{s_1 \rhd s_2P \Rightarrow \theta}\text{(M-Seq-Const-Succ)} \qquad \frac{s_2^{-1}s_1 \text{ not exist}}{s_1 \rhd s_2P \Rightarrow \bot}\text{(M-Seq-Const-Fail)}$$

$$\frac{s \rhd P_1P_3 \mid P_2P_3 \Rightarrow \theta}{s \rhd (P_1 \mid P_2)P_3 \Rightarrow \theta}\text{(M-Seq-Choice)}$$

$$\frac{y \notin var(P_1P_2) \quad s_1 \rhd P_1(y \texttt{ as } P_2) \Rightarrow \theta \uplus \{y \mapsto s_2\}}{s_1 \rhd (x \texttt{ as } P_1)P_2 \Rightarrow \theta \uplus \{x \mapsto s_1s_2^{-1}\}}\text{(M-Seq-Bind)}$$

$$\frac{s \rhd (P_1P_1^* \mid \varepsilon)P_2 \Rightarrow \theta}{s \rhd P_1^*P_2 \Rightarrow \theta}\text{(M-Seq-Rep)} \qquad \frac{s \rhd P_1(P_2P_3) \Rightarrow \theta}{s \rhd (P_1P_2)P_3 \Rightarrow \theta}\text{(M-Seq-Seq)}$$

Figure 3: Operational Semantics of Patterns

τ	(type)	::=	$\tau_1 \xrightarrow{T} \tau_2$			
		$\mid$	T			
T	(string type)	::=	s	$[\![s]\!]$	=	$\{s\}$
		$\mid$	$T_1 \mid T_2$	$[\![T_1 \mid T_2]\!]$	=	$[\![T_1]\!] \cup [\![T_2]\!]$
		$\mid$	T^*	$[\![T^*]\!]$	=	$\{s_1 + \cdots + s_n \mid s_i \in [\![T]\!] \ (i = 1, \ldots, n)\}$
		$\mid$	T_1T_2	$[\![T_1T_2]\!]$	=	$\{s_1 + s_2 \mid s_1 \in [\![T_1]\!] \wedge s_2 \in [\![T_2]\!]\}$
		$\mid$	$T_1 \cap T_2$	$[\![T_1 \cap T_2]\!]$	=	$[\![T_1]\!] \cap [\![T_2]\!]$
		$\mid$	$T_1 \cap \overline{T_2}$	$[\![T_1 \cap \overline{T_2}]\!]$	=	$[\![T_1]\!] \setminus [\![T_2]\!]$
		$\mid$	$T_1^{-1}T_2$	$[\![T_1^{-1}T_2]\!]$	=	$\{s' \mid s \in [\![T_1]\!] \wedge s + s' \in [\![T_2]\!]\}$
		$\mid$	$T_1T_2^{-1}$	$[\![T_1T_2^{-1}]\!]$	=	$\{s \mid s + s' \in [\![T_1]\!] \wedge s' \in [\![T_2]\!]\}$

Figure 4: Syntax and Semantics of Types

$$\frac{[\![T_1]\!] \subseteq [\![T_2]\!]}{T_1 \leq T_2}\text{(S-Str)} \qquad \frac{\tau_1' \leq \tau_1 \quad \tau_2 \leq \tau_2' \quad T_1 \leq T_2}{\tau_1 \xrightarrow{T_1} \tau_2 \leq \tau_1' \xrightarrow{T_2} \tau_2'}\text{(S-Fun)}$$

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau, \varepsilon}\text{(T-Var)} \qquad \frac{\begin{array}{c}\Gamma, f : \tau_1 \xrightarrow{T} \tau_2, x : \tau_1 \vdash M : \tau_2', T' \\ \tau_2' \leq \tau_2 \quad T' \leq T\end{array}}{\Gamma \vdash \mathtt{fix}(f, x, M) : \tau_1 \xrightarrow{T} \tau_2, \varepsilon}\text{(T-Fix)}$$

$$\frac{\begin{array}{c}\Gamma \vdash M_1 : \tau_2 \xrightarrow{T} \tau_1, T_1 \\ \Gamma \vdash M_2 : \tau_2', T_2 \quad \tau_2' \leq \tau_2\end{array}}{\Gamma \vdash M_1 M_2 : \tau_1, T_1 T_2 T}\text{(T-App)} \qquad \frac{}{\Gamma \vdash s : s, \varepsilon}\text{(T-Const)}$$

$$\frac{\Gamma \vdash M_1 : T_1, T_1' \quad \Gamma \vdash M_2 : T_2, T_2'}{\Gamma \vdash M_1 \mathbin{\hat{}} M_2 : T_1 T_2, T_1' T_2'}\text{(T-Cat)} \qquad \frac{\Gamma \vdash M : T, T'}{\Gamma \vdash \mathtt{print}\ M : \varepsilon, T'T}\text{(T-Print)}$$

$$\frac{\begin{array}{c}\Gamma \vdash M : T_1, T' \\ T_m \rightsquigarrow P_m \Rightarrow \Gamma_m \quad \text{warn of redundancy if } T_m \cap novar(P_m) \leq \emptyset \\ \Gamma, \Gamma_m \vdash M_m : \tau_m, T_m' \quad T_{m+1} = T_m \cap \overline{novar(P_m)} \quad (m = 1, \ldots, n) \\ T_{n+1} \leq \emptyset\end{array}}{\Gamma \vdash \mathtt{match}\ M\ \mathtt{with}\ P_1 \Rightarrow M_1 \mid \cdots \mid P_n \Rightarrow M_n : \tau_1 \mid \cdots \mid \tau_m, T'(T_1' \mid \cdots \mid T_m')}\text{(T-Match)}$$

$$\frac{([\![T]\!], P) \in \Pi \quad \forall x \in dom(\Gamma) = var(P).\, \Gamma(x) = \emptyset}{\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma}\text{(P-Mem)} \qquad \frac{}{\Pi \vdash T \rightsquigarrow s \Rightarrow \emptyset}\text{(P-Const)}$$

$$\frac{\begin{array}{c}\Pi \vdash T \rightsquigarrow P_1 \Rightarrow \Gamma_1 \quad \Pi \vdash T \cap \overline{novar(P_1)} \rightsquigarrow P_2 \Rightarrow \Gamma_2 \\ \forall x \in dom(\Gamma) = dom(\Gamma_1) = dom(\Gamma_2).\, \Gamma(x) = \Gamma_1(x) \mid \Gamma_2(x)\end{array}}{\Pi \vdash T \rightsquigarrow P_1 \mid P_2 \Rightarrow \Gamma}\text{(P-Chioce)}$$

$$\frac{\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma}{\Pi \vdash T \rightsquigarrow x\ \mathtt{as}\ P \Rightarrow \Gamma \uplus \{x : \Gamma(P) \cap T\}}\text{(P-Bind)} \qquad \frac{\Pi \uplus \{([\![T]\!], P^*)\} \vdash T \rightsquigarrow PP^* \mid \varepsilon \Rightarrow \Gamma}{\Pi \vdash T \rightsquigarrow P^* \Rightarrow \Gamma}\text{(P-Rep)}$$

$$\frac{\Pi \vdash s^{-1}T \rightsquigarrow P \Rightarrow \Gamma}{\Pi \vdash T \rightsquigarrow sP \Rightarrow \Gamma}\text{(P-Seq-Const)} \qquad \frac{\Pi \vdash T \rightsquigarrow P_1 P_3 \mid P_2 P_3 \Rightarrow \Gamma}{\Pi \vdash T \rightsquigarrow (P_1 \mid P_2) P_3 \Rightarrow \Gamma}\text{(P-Seq-Choice)}$$

$$\frac{\forall ([\![T]\!], P) \in \Pi.\, y \notin var(P) \quad y \notin var(P_1 P_2) \quad \Pi \vdash T_1 \rightsquigarrow P_1(y\ \mathtt{as}\ P_2) \Rightarrow \Gamma \uplus \{y : T_2\}}{\Pi \vdash T_1 \rightsquigarrow (x\ \mathtt{as}\ P_1) P_2 \Rightarrow \Gamma \uplus \{x : \Gamma(P_1) \cap T_1 T_2^{-1}\}}\text{(P-Seq-Bind)}$$

$$\frac{\Pi \uplus ([\![T]\!], P_1^* P_2) \vdash T \rightsquigarrow (P_1 P_1^* \mid \varepsilon) P_2 \Rightarrow \Gamma}{\Pi \vdash T \rightsquigarrow P_1^* P_2 \Rightarrow \Gamma}\text{(P-Seq-Rep)} \qquad \frac{\Pi \vdash T \rightsquigarrow P_1(P_2 P_3) \Rightarrow \Gamma}{\Pi \vdash T \rightsquigarrow (P_1 P_2) P_3 \Rightarrow \Gamma}\text{(P-Seq-Seq)}$$

Figure 5: Typing Rules

$$\frac{}{S \rhd s \Rightarrow \{(s^{-1}s', \theta) \mid (s', \theta) \in S\}}\text{(SM-Const)}$$

$$\frac{S \rhd P_1 \Rightarrow S_1 \quad S \rhd P_2 \Rightarrow S_2}{S \rhd P_1 \mid P_2 \Rightarrow S_1 \cup S_2}\text{(SM-Choice)} \qquad \frac{\begin{array}{c} S \rhd P_1 \Rightarrow S_1 \\ S_1 \rhd P_2 \Rightarrow S_2 \end{array}}{S \rhd P_1 P_2 \Rightarrow S_2}\text{(SM-Seq)}$$

$$\frac{\begin{array}{c} S'' = \{(s', \theta' \uplus \{x \mapsto ss'^{-1}\}) \mid \\ (s, \theta) \in S \wedge \{(s, \theta)\} \rhd P \Rightarrow S' \wedge (s', \theta') \in S'\} \end{array}}{S \rhd x \texttt{ as } P \Rightarrow S''}\text{(SM-Bind)}$$

$$\frac{S \rhd P \Rightarrow \emptyset}{S \rhd P^* \Rightarrow S}\text{(SM-Rep-NoMore)} \qquad \frac{\begin{array}{c} S \rhd P \Rightarrow S_1 \quad S_1 \neq \emptyset \\ S_1 \rhd P^* \Rightarrow S_2 \end{array}}{S \rhd P^* \Rightarrow S \cup S_2}\text{(SM-Rep-More)}$$

Figure 6: Set-Based Pattern Matching

The proof proceeds as follows: we first prove the termination (and determinacy) property of our new set-based matching, and then, show that the new matching algorithm ($S \rhd P \Rightarrow S'$) is actually compatible with the original one ($s \rhd P \Rightarrow \theta$). Therefore, the desired result immediately follows.

2.2 Definitions

Definition 1 (Set-based Pattern Matching). The set based pattern matching $S \rhd P \Rightarrow S'$, where S and S' are sets of pairs (s, θ) of a string and a substitution, is defined as the least relation derivable by the rules in Figure 6.

Definition 2 (Weight of Patterns). The "weight" of a pattern P, written $||P||$, is defined inductively on the structure of P as follows:

$$\begin{array}{rcl} ||s|| & = & 1 \\ ||P_1 P_2|| & = & 2 \times ||P_1|| + ||P_2|| \\ ||P_1 \mid P_2|| & = & ||P_1|| + ||P_2|| + 1 \\ ||x \texttt{ as } P|| & = & 1 + ||P|| \\ ||P^*|| & = & 1 + ||P|| \end{array}$$

Definition 3. We write $len(S)$ for the maximum length of strings in S. That is, $len(S) = \max_{(s,\theta) \in S} |s|$

Definition 4. We write $S{\downarrow_s}$ for the set $\{\theta \mid (s, \theta) \in S\}$ of substitutions paired with s in S.

2.3 Proofs

Lemma 5. If $S \rhd P \Rightarrow S'$ and $S = S_1 \cup S_2$, then $S_1 \rhd P \Rightarrow S'_1$, $S_2 \rhd P \Rightarrow S'_2$ with $S' = S'_1 \cup S'_2$. Conversely, if $S_1 \rhd P \Rightarrow S'_1$, $S_2 \rhd P \Rightarrow S'_2$ and $S' = S'_1 \cup S'_2$, then $S_1 \cup S_2 \rhd P \Rightarrow S'$.

Proof. Straightforward induction on the derivation of $S \rhd P \Rightarrow S'$. □

Lemma 6. If $\{(s,\theta)\} \rhd P \Rightarrow S$, then $ss'^{-1} \in [\![novar(P)]\!]$ for each $(s',\theta') \in S$.

Proof. Straightforward induction on the derivation of $\{(s,\theta)\} \rhd P \Rightarrow S$. We use lemma 5 if $S_1 \rhd P \Rightarrow S_2$ and $|S_1| \geq 2$ in applying the induction hypothesis. □

Lemma 7. If $s_1 \rhd P_1$ and $s_2 \rhd P_2$, then $s_1 s_2 \rhd P_1 P_2$. Conversely, if $s \rhd P_1 P_2$, then $s_1 \rhd P_1$ and $s_2 \rhd P_2$ for some s_1, s_2 such that $s = s_1 s_2$.

Proof. Induction on the lexicographic order of the pair $(||P_1||, |s_1 s_2|)$.

Case $P_1 = s$:
$s_1 \rhd s$ if and only if $s_1 = s$. By the definition of (M-Seq-Const), $ss_2 \rhd sP_2$ if and only if $s_2 \rhd P_2$.

Case $P_1 = P'^*$:
Suppose $s_2 \rhd P_2$. If $s_1 = \varepsilon$, then the statement trivially holds. Otherwise, by the induction hypothesis, if $s_1 \rhd P' P'^*$, then $s_{11} \rhd P'$ and $s_{12} \rhd P'^*$ with $s_{11} s_{12} = s_1$. Since we assumed patterns being standardized, $s_{11} \neq \varepsilon$ thus $|s_{12} s_2| < |s_1 s_2|$. Therefore, $s_{12} s_2 \rhd P'^* P_2$ and $s_{11} s_{12} s_2 \rhd P' P'^* P_2$ by the induction hypothesis. As the conclusion, $s_1 s_2 \rhd P'^* P_2$ by the definition of (M-Seq-Rep).

Other Cases:
Similar.

□

Lemma 8 (Determinacy and Termination of Set-based Matching). For arbitrary S and P, there exists a unique S' such that $S \rhd P \Rightarrow S'$ (under the condition that P is standardized).

Proof. The proof proceeds by the induction on the lexicographic order of the pair $(||P||, len(S))$. We only show the proof for $P = P_0^*$. Other cases are not difficult. Note that $\varepsilon \notin [\![novar(P_0)]\!]$ since we assumed P being standardized without loss of generality.

By the induction hypothesis, $S \rhd P_0 \Rightarrow S'$ for a unique S'. Suppose $len(S) = 0$ and thus $\forall (s,\theta) \in S.\, s = \varepsilon$. Then, from lemma 5, 6 and $\varepsilon \notin [\![novar(P_0)]\!]$, $S' = \emptyset$ follows. We can thus derive $S \rhd P_0^* \Rightarrow S$ by (SM-Rep-NoMore). If $len(S) = n > 0$ and $S' \neq \emptyset$, then $len(S') < n$ since P_0^* is, again, standardized. Therefore, by lemma 5 and 6, each $(s',\theta') \in S'$ satisfies $s'' s' = s$ for some $(s,\theta) \in S$ and $s'' \in [\![novar(P_0)]\!]$. Then, by the induction hypothesis, there exists a unique S'' such that $S' \rhd P_0^* \Rightarrow S''$, with which we can derive $S \rhd P_0^* \Rightarrow S \cup S''$ by (SM-Rep-More).

□

Lemma 9. If $\{(s,\emptyset)\} \rhd P \Rightarrow S$ and $S\downarrow_\varepsilon \neq \emptyset$, then there exists a unique $\theta \in S\downarrow_\varepsilon$ such that $s \rhd P \Rightarrow \theta$. Otherwise, $s \rhd P \Rightarrow \bot$.

Proof. We prove the following stronger statement. If $S \rhd P \Rightarrow S'$ and $S'\downarrow_\varepsilon \neq \emptyset$, then for some $(s,\theta) \in S$ and $\theta' \in S'\downarrow_\varepsilon$, $s \rhd P \Rightarrow \theta''$ $(\neq \bot)$ with $\theta' = \theta \uplus \theta''$. Furthermore, θ' is unique with respect to θ. If $S'\downarrow_\varepsilon = \emptyset$, $s \rhd P \Rightarrow \bot$ for all $(s,\theta) \in S$. The proof proceeds by induction on the lexicographic order of the pair $(||P||, len(S))$. We only show the cases of $P = P_1 P_2$ and perform further case analysis on the structure of P_1. Other cases are easier.

Case $P_1 = s_1$:
Suppose $S \rhd P_1 \Rightarrow S_1$ and $S_1 \rhd P_2 \Rightarrow S_2$. By the definition of (SM-Const), $S_1 = \{(s_1{}^{-1}s, \theta) \mid (s,\theta) \in S \wedge s_1{}^{-1}s \text{ exists}\}$. By the induction hypothesis, if $S_2 \neq \emptyset$, then there exists a unique

$\theta' \in S_2{\downarrow_\varepsilon}$ such that $s \rhd P_2 \Rightarrow \theta''$ $(\neq \bot)$ and $\theta' = \theta \uplus \theta''$ for some $(s, \theta) \in S_1$. Otherwise, $s \rhd P_2 \Rightarrow \bot$ for all $(s, \theta) \in S_1$.

Case $P_1 = P_{11} \mid P_{12}$:

By the definition of (SM-Choice), $S_1 = S_{11} \cup S_{12}$ where $S \rhd P_{1i} \Rightarrow S_{1i}$ $(i = 1, 2)$. Therefore by lemma 5, $S_2 = S_{21} \cup S_{22}$ where $S_{1i} \rhd P_2 \Rightarrow S_{2i}$ $(i = 1, 2)$. By the induction hypothesis, $s \rhd P_{1i}P_2 \Rightarrow \theta''$ $(\neq \bot)$ for some $(s, \theta) \in S$ if and only if $S_{2i}{\downarrow_\varepsilon} \neq \emptyset$, with a unique $\theta' \in S_{2i}{\downarrow_\varepsilon}$ such that $\theta' = \theta \uplus \theta''$ $(i = 1, 2)$. Thus, $s \rhd (P_{11}P_2) \mid (P_{12}P_2)$ for some $(s, \theta) \in S$ if and only if $S_2{\downarrow_\varepsilon} \neq \emptyset$. The statement of the lemma follows from the definition of (M-Seq-Choice).

Case $P_1 = P_{11}P_{12}$:

By the definition of $||P||$, $||P_{11}(P_{12}P_2)|| < ||(P_{11}P_{12})P_2||$. Therefore by the induction hypothesis, if $S \rhd P_{11} \Rightarrow S_{11}$, $S_{11} \rhd P_{12}P_2 \Rightarrow S'_2$ and $S'_2{\downarrow_\varepsilon} \neq \emptyset$ $(= \emptyset)$, then $s \rhd P_{11}(P_{12}P_2) \Rightarrow \theta''$ $(\neq \bot)$ $(s \rhd P_{11}(P_{12}P_2) \Rightarrow \bot)$ for some $(s, \theta) \in S$, with a unique $\theta' \in S'_2{\downarrow_\varepsilon}$ such that $\theta' = \theta \uplus \theta''$, respectively. By lemma 8 and by the definition of (SM-Seq), $S'_2 = S_2$.

Case $P_1 = x$ `as` P':

Note that $||(x \texttt{ as } P')P_2|| > ||P'(y \texttt{ as } P_2)||$ by the definition of $||P||$. Suppose $S \rhd P' \Rightarrow S'_1$ and $S'_1 \rhd y \texttt{ as } P_2 \Rightarrow S'_2$ where $y \notin var(P_1P_2)$. Then, by the induction hypothesis, if $S'_2{\downarrow_\varepsilon} \neq \emptyset$, there exist some $(s, \theta) \in S$ and $\theta_2 \in S'_2{\downarrow_\varepsilon}$ such that $s \rhd P_1(y \texttt{ as } P_2) \Rightarrow \theta_3 (\neq \bot)$ and $\theta_2 = \theta \uplus \theta_3$, in addition, θ_2 is unique with respect to (s, θ). By the definition of (SM-Bind), $\theta_2(y) = s_1$ for some $(s_1, \theta_4) \in S'_1$ such that s_1 is a suffix of s. Therefore, by the definition of (M-Seq-Bind), $s \rhd (x \texttt{ as } P')P_2 \Rightarrow \theta'_3 \uplus \{x \mapsto s{s_1}^{-1}\}$ where $\theta'_3\{y \mapsto s'\} = \theta_3$ for some s'.

Case $P_1 = P'^*$:

If $S \rhd P' \Rightarrow \emptyset$ and thus $S \rhd P'^* \Rightarrow S$, then the statement immediately follows from the induction hypothesis. If $S \rhd P' \Rightarrow S_1 \neq \emptyset$, then $s \rhd P'$ for some $s \in S$ and since we assumed P' being standardized, $len(S_1) < len(S)$. Therefore by the induction hypothesis, if $S_1 \rhd P'^*P_2 \Rightarrow S_2$ and $S_2{\downarrow_\varepsilon} \neq \emptyset$, then $s_1 \rhd P'^*P_2 \Rightarrow \theta_3$ for some $(s_1, \theta) \in S_1$ and $\theta_2 \in S_2{\downarrow_\varepsilon}$. Note that $var(P') = \emptyset$ since we allow substitutions to be extended only conservatively. Therefore s_1 is a suffix of some $(s, \theta) \in S$ and thus $s{s_1}^{-1} \rhd P' \Rightarrow \emptyset$ by the induction hypothesis. As the conclusion, the statement follows from lemma 7 and from the definition of (M-Seq-Rep).

□

Lemma 10 (Determinacy and Termination of Pattern-matching). If $s \rhd P \Rightarrow \theta$ and $s \rhd P \Rightarrow \theta'$, then $\theta = \theta'$. $s \rhd P$ if and only if $s \in [\![novar(P)]\!]$. Conversely, $s \rhd P \Rightarrow \bot$ if and only if $s \notin [\![novar(P)]\!]$.

Proof. Immediately follows from lemma 8 and 9. □

Lemma 11. If $s \rhd P$ then $s \in [\![novar(P)]\!]$. if $s \rhd P \Rightarrow \bot$ then $s \notin [\![novar(P)]\!]$.

Proof. Straightforward induction on the derivation of $s \rhd P \Rightarrow \theta$. □

Lemma 12. $s \rhd P \iff s \in [\![novar(P)]\!]$.

Proof. Immediately follows from lemma 10 and 11. □

3 Soundness and Completeness of Type Inference for Pattern Matching

3.1 Overview of the Proof

Since the proof of soundness and completeness property is rather tricky, we first explain briefly how it proceeds. Then, why cannot we prove it in a straightforward manner? The essential reason lies in the existence of Π of memoization set. Remember, what we want to show is that "If $\emptyset \vdash T \rightsquigarrow P \Rightarrow \Gamma$, then each $\Gamma(x)$ of inferred type of x exactly agrees with the set $\{\theta(x) \mid s \in [\![T]\!] \wedge s \rhd P \Rightarrow \theta \wedge \theta \neq \bot\}$ of strings which can actually be bound to x." Notice that we cannot naïvely proceed by induction on the derivation of $\emptyset \vdash T \rightsquigarrow P \Rightarrow \Gamma$, because Π may become non-empty at (P-Rep) etc. where the induction hypothesis cannot be applied. Neither can we simply strengthen the statement to, say, "If $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma$ then ..." because Π may contain any "garbage" which interfere with the derivation. Below is an example of the case in which such garbage breaks the property (against the $\supseteq$ part):

$$\cfrac{\cfrac{([\![\texttt{ab}]\!], \texttt{a}(y \texttt{ as } \texttt{b})) \in \{([\![\texttt{ab}]\!], \texttt{a}(y \texttt{ as } \texttt{b}))\}}{\{([\![\texttt{ab}]\!], \texttt{a}(y \texttt{ as } \texttt{b}))\} \vdash \texttt{ab} \rightsquigarrow \texttt{a}(y \texttt{ as } \texttt{b}) \Rightarrow \{y : \emptyset\}}\text{(P-Mem)}}{\{([\![\texttt{ab}]\!], \texttt{a}(y \texttt{ as } \texttt{b}))\} \vdash \texttt{ab} \rightsquigarrow (x \texttt{ as } \texttt{a})\texttt{b} \Rightarrow \{x : \emptyset\}}\text{(P-Seq-Bind)}$$

Here, the inferred type of variable x is $\emptyset$, but an instant observation tells us it should be $\texttt{a}$ (or supertype of it, if we only care about the $\supseteq$ part). Taking these facts into account, we must ensure that Π does not contain such harmful garbage at each step of derivation. To do so, some extra definitions are required including new derivation rules and predicates on Π. They are introduced informally in following paragraphs.

First of all, the typing relation $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma$ is extended to $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$ to record the set of pairs $([\![T]\!], P)$ of a type and a pattern which are *actually used* during the derivation. The whole revised rules are defined in Figure 7. Our new definition has two major changes compared to the old one: (1) the new rule returns Π' of a set of pairs $([\![T]\!], P)$ which were used by (P-Mem) on the top of derivation and (2) repetition and choice patterns are directly expanded as shown in (RP-Rep), (RP-Seq-Choice) and (RP-Seq-Rep), for brevity of the proof. It is not difficult to observe the new rules and the old ones are compatible to each other.

Secondly, two predicates on Π are defined which states Π has desirable properties: (1) the well-formedness of Π (with respect to a type T and a pattern P) on the bottom of derivation, written $\Pi \overset{P}{\bowtie} T$, states that Π contains nothing harmful for T and P of current interest in derivation of $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$, and (2) the well-consumedness of used pairs Π (with respect to a type T) on the top of derivation, written $\Pi \smile T$ and used in the form of $\Pi \cap \Pi' \smile T$, states that elements in Π were used only in a way that does not destroy our soundness/completeness property.

Then, the proof proceeds as follows. (1) We show that if $\Pi \overset{P}{\bowtie} T$ holds at the derivation of $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$, then the same property holds at the next step of derivation (i.e. if Π seems to contain nothing injurious at the starting point, none of its elements suddenly become injurious in remaining derivation[1]), and (2) if $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$ and $\Pi \overset{P}{\bowtie} T$, then $\Pi \cap \Pi' \smile T$ holds (i.e. a derivation started with a well-formed memoization set would use only desirable elements in it). To tie the $\Pi \overset{P}{\bowtie} T$ (well-formedness) and $\Pi \cap \Pi' \smile T$ (well-consumedness), we employ another auxiliary relation $\Pi \hookrightarrow \Pi' \vdash T \rightsquigarrow P$ as in Figure 8.

[1]In fact, the definition of $\Pi \overset{P}{\bowtie} T$ were chosen to be invariant, in an *ad hoc* manner

Finally, we prove the main statement: if $\Pi \vdash T \leadsto P \Rightarrow \Gamma; \Pi'$ and $\Pi \overset{P}{\bowtie} T$, then soundness and completeness are satisfied (in a slightly strengthened form taking elements in Π' into account).

3.2 Definitions

Definition 13 (Expansion Relation between Patterns). The "expansion relation" between P and P', written $P \overset{\flat}{\sim} P'$, is defined as follows:

$$
\begin{array}{ll}
 & P \overset{\flat}{\sim} P' \\
\iff & \\
 & P' = P_1^* \wedge P = P_1' P' \wedge \varepsilon \not\leq \mathit{novar}(P_1') \leq \mathit{novar}(P_1) \text{ for some } P_1, P_1' \text{ or} \\
 & P' = P_1^* P_2 \wedge P = P_1' P_1^* P_2 \wedge \varepsilon \not\leq \mathit{novar}(P_1') \leq \mathit{novar}(P_1) \text{ for some } P_1, P_2 \text{ and } P_1'
\end{array}
$$

This relation roughly states "P' has a repetition pattern P_1^* in its head and P is obtained from P' by expanding that head-position repetition into $P_1 P_1^*$." Note that P' needs to be standardized for $P \overset{\flat}{\sim} P'$ to hold for some P.

Definition 14. We define a partial order $\sqsubseteq$ between types as follows:

$$T \sqsubseteq T' \iff \exists T''. T \leq T''^{-1} T'$$

We write $T \sqsubset T'$ when $T \sqsubseteq T'$ and $T \not\equiv T'$

This relation states that T is semantically compatible to T' with some prefix removed. Examples are:

$$
\begin{array}{ll}
\mathtt{ab} \sqsubseteq \mathtt{aab} & \text{since } \mathtt{ab} \equiv \mathtt{a}^{-1}\mathtt{aab} \\
\mathtt{a}^* \sqsubseteq \mathtt{a}^* & \text{since } \mathtt{a}^* \equiv \mathtt{a}^{-1}\mathtt{a}^*, (\mathtt{aa})^{-1}\mathtt{a}^*, \ldots \text{ etc.} \\
\mathtt{ab} \not\sqsubseteq \mathtt{abb} & \text{since there exists no } T \text{ such that } \mathtt{ab} \equiv \mathtt{T}^{-1}\mathtt{abb}
\end{array}
$$

Remark 15.

$$
\begin{array}{c}
T \sqsubseteq T' \Rightarrow T''^{-1} T \sqsubseteq T' \\
T \leq T' \wedge T' \sqsubseteq T'' \Rightarrow T \sqsubseteq T''
\end{array}
$$

.

Definition 16 (Range of x in P with respect to T). We write $R(x, T, P)$ for the set $\{\theta(x) \mid s \in [\![T]\!] \wedge s \rhd P \Rightarrow \theta \wedge \theta \neq \bot\}$ of strings which can be bound to x ($x \in \mathit{var}(P)$) when a string of type T matches P.

$$R(x, T, P) = \{\theta(x) \mid s \in [\![T]\!] \wedge s \rhd P \Rightarrow \theta \wedge \theta \neq \bot\}$$

Definition 17 (Range of x in Π with respect to T). We write $R^{mem}(x, T, \mathit{\Pi})$ for the union of sets of $R(x, T, P)$ where the pair $([\![T]\!], P)$ is included in Π.

$$R^{mem}(x, T, \mathit{\Pi}) = \bigcup_{([\![T]\!], P) \in \Pi} (R(x, T, P))$$

Definition 18. We write $\Pi(T)$ for a type which satisfy the following equation:

$$[\![\Pi(T)]\!] = \bigcup_{([\![T]\!], P) \in \Pi} ([\![\mathit{novar}(P)]\!])$$

$$\frac{}{\Pi \vdash T \rightsquigarrow s \Rightarrow \emptyset; \emptyset}\text{(RP-Const)}$$

$$\frac{\begin{array}{c}\Pi \vdash T \rightsquigarrow P_1 \Rightarrow \Gamma_1; \Pi_1 \quad \Pi \vdash T \cap \overline{novar(P_1)} \rightsquigarrow P_2 \Rightarrow \Gamma_2; \Pi_2 \\ \forall x \in dom(\Gamma) = dom(\Gamma_1) = dom(\Gamma_2).\ \Gamma(x) = \Gamma_1(x) \mid \Gamma_2(x)\end{array}}{\Pi \vdash T \rightsquigarrow P_1 \mid P_2 \Rightarrow \Gamma; \Pi_1 \cup \Pi_2}\text{(RP-Choice)}$$

$$\frac{\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'}{\Pi \vdash T \rightsquigarrow x \texttt{ as } P \Rightarrow \Gamma, x : \Gamma(P) \cap T; \Pi'}\text{(RP-Bind)}$$

$$\frac{\begin{array}{c}\Pi \uplus \{(\llbracket T \rrbracket, P^*)\} \vdash T \rightsquigarrow PP^* \Rightarrow \emptyset; \Pi_1 \\ \Pi \vdash T \cap \overline{PP^*} \rightsquigarrow \varepsilon \Rightarrow \emptyset; \Pi_2 \quad var(P) = \emptyset\end{array}}{\Pi \vdash T \rightsquigarrow P^* \Rightarrow \emptyset; \Pi_1 \cup \Pi_2}\text{(RP-Rep)}$$

$$\frac{\Pi \vdash s^{-1}T \rightsquigarrow P \Rightarrow \Gamma; \Pi'}{\Pi \vdash T \rightsquigarrow sP \Rightarrow \Gamma; \Pi'}\text{(RP-Seq-Const)}$$

$$\frac{\begin{array}{c}\Pi \vdash T \rightsquigarrow P_1P_2 \Rightarrow \Gamma_1; \Pi_1 \\ \Pi \vdash T \cap \overline{novar(P_1P_2)} \rightsquigarrow P_1P_3 \Rightarrow \Gamma_2; \Pi_2 \\ \forall x \in dom(\Gamma) = dom(\Gamma_1) = dom(\Gamma_2).\ \Gamma(x) = \Gamma_1(x) \mid \Gamma_2(x)\end{array}}{\Pi \vdash T \rightsquigarrow (P_1 \mid P_2)P_3 \Rightarrow \Gamma; \Pi'}\text{(RP-Seq-Choice)}$$

$$\frac{\begin{array}{c}\forall(\llbracket T \rrbracket, P) \in \Pi.\ y \notin var(P) \quad y \notin var(P_1P_2) \\ \Pi \vdash T \rightsquigarrow P_1(y \texttt{ as } P_2) \Rightarrow \Gamma, y : T'; \Pi'\end{array}}{\Pi \vdash T \rightsquigarrow (x \texttt{ as } P_1)P_2 \Rightarrow \Gamma, x : \Gamma(P_1) \cap (TT'^{-1}); \Pi'}\text{(RP-Seq-Bind)}$$

$$\frac{\begin{array}{c}\Pi \uplus (\llbracket T \rrbracket, P_1^*P_2) \vdash T \rightsquigarrow (P_1P_1^*)P_2 \Rightarrow \Gamma_1; \Pi_1 \\ \Pi \vdash T \cap \overline{novar(P_1P_1^*P_2)} \rightsquigarrow P_2 \Rightarrow \Gamma_2; \Pi_2 \\ \forall x \in dom(\Gamma) = dom(\Gamma_1) = dom(\Gamma_2).\ \Gamma(x) = \Gamma_1(x) \mid \Gamma_2(x)\end{array}}{\Pi \vdash T \rightsquigarrow P_1^*P_2 \Rightarrow \Gamma; \Pi_1 \cup \Pi_2}\text{(RP-Seq-Rep)}$$

$$\frac{\Pi \vdash T \rightsquigarrow P_1(P_2P_3) \Rightarrow \Gamma; \Pi'}{\Pi \vdash T \rightsquigarrow (P_1P_2)P_3 \Rightarrow \Gamma; \Pi'}\text{(RP-Seq-Seq)}$$

$$\frac{(\llbracket T \rrbracket, P) \in \Pi \quad \forall x \in dom(\Gamma) = var(P).\ \Gamma(x) = \emptyset}{\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \{(\llbracket T \rrbracket, P)\}}\text{(RP-Mem)}$$

Figure 7: Revised Typing Rules for Patterns

Remark 19. If $\Pi(T) = \emptyset$, then $R^{mem}(x, T, \mathit{\Pi}) = \emptyset$.

Definition 20 (Well-formedness of Π). We call a Π is *well-formed* with respect to T and P, written $\Pi \overset{P}{\bowtie} T$, if and only if

$$\begin{array}{ll} \forall([\![T']\!], P') \in \Pi. & \\ \quad 1.\ T \sqsubseteq T' & \text{and} \\ \quad 2.\ T \sqsubset T' \Rightarrow ||P|| \leq ||P'|| & \text{and} \\ \quad 3.\ T \equiv T' \Rightarrow ||P|| < ||P'|| \vee P \overset{\flat}{\sim} P') & \text{and} \\ \quad 4.\ (\Pi \setminus \{([\![T']\!], P')\})(T') = \emptyset & \end{array}$$

Intuitively, well-formed sets are those obtained during a derivation that started with $\Pi = \emptyset$. The well-formedness is not always a necessary condition of soundness and completeness. For instance, the following derivation $\Pi \vdash \mathtt{a} \rightsquigarrow x \ \mathtt{as}\ \mathtt{a}^* \Rightarrow \{x : \mathtt{a}\}$ is still sound and complete with, say, $\Pi = \{([\![\mathtt{a}]\!], \mathtt{c}), ([\![\mathtt{a}]\!], \mathtt{d})\}$. Nevertheless, we neglect such overgeneralization for simplicity.

Definition 21 (Well-consumedness of Π). We call a Π is *well-consumed* with respect to T, written $\Pi \smile T$, if and only if

$$\begin{array}{ll} \text{if } ([\![T]\!], P') \in \Pi \text{ then} & \\ \quad 1.\ (\Pi \setminus \{([\![T]\!], P')\})(T) = \emptyset & \text{and} \\ \quad 2.\ \text{if } P' = P_1P_2 \text{ then } \mathit{novar}(P_1)^{-1}T \equiv T \wedge \mathit{var}(P_1) = \emptyset & \end{array}$$

Definition 22 (Valid Connection of Π, Π'). We call Π and Π' being *validly connected* with respect to the derivation $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$, written $\Pi \hookrightarrow \Pi' \vdash T \rightsquigarrow P$, if and only if $\Pi \hookrightarrow \Pi' \vdash T \rightsquigarrow P$ is derivable by the rules in Figure 8.

3.3 Proofs

Lemma 23. For arbitrary Γ and P, $\Gamma(P) \leq \mathit{novar}(P)$ holds.

Proof. Straightforward induction on the structure of P. □

Lemma 24. For each variable $x \in \mathit{var}(P_2)$ of a pattern P_1P_2, $R(x, T, P_1P_2) \subseteq R(x, \mathit{novar}(P_1)^{-1}T, P_2)$.

Proof. Suppose $s \in R(x, T, P_1P_2)$. Then, by definition,

$$s \in \{\theta(x) \mid s' \in [\![T]\!] \wedge s' \rhd P_1P_2 \Rightarrow \theta(\neq \bot)\}$$

By lemma 12 and definition of semantics of types, this is equivalent to

$$\begin{array}{rl} s \in \{\theta(x) \mid & s_1s_2 \in [\![T]\!] \wedge s_1 \in [\![\mathit{novar}(P_1)]\!] \\ & \wedge s_2 \rhd P_2 \Rightarrow \theta'(\neq \bot) \\ & \wedge s_1s_2 \rhd P_1P_2 \Rightarrow \theta(\neq \bot)\} \end{array}$$

.

Then, we can show that there always exist appropriate s_1 and s_2 such that $s_1s_2 = s \wedge s_1 \rhd P_1 \wedge s_2 \rhd P_2 \Rightarrow \theta'(\neq \bot) \wedge \forall x \in \mathit{var}(P_2).(\theta'(x) = \theta(x))$, by straightforward induction on the derivation of $s \rhd P_1P_2 \Rightarrow \theta$.

□

$$\frac{}{\Pi \hookrightarrow \emptyset \vdash T \leadsto s}\text{(VC-Const)} \qquad \frac{(\llbracket T \rrbracket, P) \in \Pi \quad \{(\llbracket T \rrbracket, P)\} \smile T}{\Pi \hookrightarrow \{(\llbracket T \rrbracket, P)\} \vdash T \leadsto P}\text{(VC-Mem)}$$

$$\frac{\Pi \hookrightarrow \Pi_1 \vdash T \leadsto P_1 \quad \Pi \hookrightarrow \Pi_2 \vdash T \cap \overline{novar(P_1)} \leadsto P_2}{\Pi \hookrightarrow \Pi_1 \cup \Pi_2 \vdash T \leadsto P_1 \mid P_2}\text{(VC-Choice)}$$

$$\frac{\Pi \hookrightarrow \Pi' \vdash T \leadsto P}{\Pi \hookrightarrow \Pi' \vdash T \leadsto x \texttt{ as } P}\text{(VC-Bind)}$$

$$\frac{\Pi \uplus \{(\llbracket T \rrbracket, P^*)\} \hookrightarrow \Pi' \vdash T \leadsto PP^* \mid \varepsilon}{\Pi \hookrightarrow \Pi' \vdash T \leadsto P^*}\text{(VC-Rep)}$$

$$\frac{\Pi \hookrightarrow \Pi' \vdash s^{-1}T \leadsto P}{\Pi \hookrightarrow \Pi' \vdash T \leadsto sP}\text{(VC-Seq-Const)}$$

$$\frac{\Pi \hookrightarrow \Pi' \vdash T \leadsto P_1(P_2P_3)}{\Pi \hookrightarrow \Pi' \vdash T \leadsto (P_1P_2)P_3}\text{(VC-Seq-Seq)}$$

$$\frac{\Pi \hookrightarrow \Pi_1 \vdash T \leadsto P_1P_2 \quad \Pi \hookrightarrow \Pi_2 \vdash T \cap \overline{novar(P_1P_2)} \leadsto P_1P_3}{\Pi \hookrightarrow \Pi' \vdash T \leadsto (P_1 \mid P_2)P_3}\text{(VC-Seq-Choice)}$$

$$\frac{\begin{array}{c}\Pi \hookrightarrow \Pi' \vdash T \leadsto P_1(y \texttt{ as } P_2) \\ \forall (\llbracket T \rrbracket, P) \in \Pi.\, y \notin var(P) \quad y \notin var(P_1P_2)\end{array}}{\Pi \hookrightarrow \Pi' \vdash T \leadsto (x \texttt{ as } P_1)P_2}\text{(VC-Seq-Bind)}$$

$$\frac{\begin{array}{c}\Pi \uplus \{(\llbracket T \rrbracket, P_1^*P_2)\} \hookrightarrow \Pi_1 \vdash T \leadsto P_1P_1^*P_2 \\ \Pi \hookrightarrow \Pi_2 \vdash T \cap \overline{novar(P_1P_1^*P_2)} \leadsto P_2\end{array}}{\Pi \hookrightarrow \Pi_1 \cup \Pi_2 \vdash T \leadsto P_1^*P_2}\text{(VC-Seq-Rep)}$$

Figure 8: Derivation Rules of Valid-Connection Relation

Lemma 25. For arbitrary T, P, if $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$ and $\Pi \overset{P}{\bowtie} T$ hold, then $\Pi \hookrightarrow \Pi' \vdash T \rightsquigarrow P$.

Proof. Induction on the derivation of $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$. We only prove the following cases: (RP-Mem), (RP-Const), (RP-Bind), (RP-Seq-Rep) and (RP-Seq-Const). The proof proceed by the case analysis on the final rule in the derivation.

Case (RP-Mem): Vacuously satisfied since $\Pi \overset{P}{\bowtie} T$ does not hold.

Case (RP-Const): Trivial. $\Pi \hookrightarrow \emptyset \vdash T \rightsquigarrow s$ always holds for arbitrary Π, T and s.

Case (RP-Bind): Suppose $\Pi \overset{x \texttt{ as } P}{\bowtie} T$. Then, $\Pi(T) = \emptyset$, because by the definition of $\overset{\flat}{\sim}$, there exists no P' that satisfy $(x \texttt{ as } P) \overset{\flat}{\sim} P'$. Since $||x \texttt{ as } P|| > ||P||$, $\Pi \overset{P}{\bowtie} T$ also holds. Therefore the statement follows from the induction hypothesis.

Case (RP-Seq-Rep):

By the definition of $\overset{\flat}{\sim}$, there exists no P such that $P_1^* P_2 \overset{\flat}{\sim} P$, thus $\Pi(T) = \emptyset$. Therefore, if $T \cap \overline{novar(P_1 P_1^* P_2)} \equiv T$, then $\Pi \overset{P_2}{\bowtie} T \cap \overline{novar(P_1 P_1^* P_2)}$ also holds. If $T \cap \overline{novar(P_1 P_1^* P_2)} \not\equiv T$, then, by the definition of $\Pi \overset{P}{\bowtie} T$ and $\sqsubseteq$, we can conclude $T \cap \overline{novar(P_1 P_1^* P_2)} \sqsubset T'$ holds for all$([\![T']\!], P') \in \Pi$. Thus, $\Pi \overset{P_2}{\bowtie} T \cap \overline{novar(P_1 P_1^* P_2)}$ also holds in this case.

We assumed patterns being standardized, which implies $\varepsilon \not\leq novar(P_1)$, thus $\Pi \uplus \{([\![T]\!], P_1^* P_2)\} \overset{P_1 P_1^* P_2}{\bowtie} T$ obviously holds by definition.

Therefore, we can apply the induction hypothesis on both branches of the derivation.

Case (RP-Seq-Const):

Suppose $\Pi \overset{sP}{\bowtie} T$ and $([\![T]\!], P') \in \Pi$ for some P'. Then, P' satisfies $sP \overset{\flat}{\sim} P'$, which implies there exist some P_1' and P_2' such that $s \leq novar(P_2')$ and $P = P' = (P_2')^* P_1'$. There are two possibilities. (1) If $s^{-1}T \not\equiv T$, then $\Pi \overset{P}{\bowtie} s^{-1}T$ holds because $s^{-1}T \sqsubset T'$ and $||P|| \leq ||P'||$ hold for each $([\![T']\!], P') \in \Pi$. Therefore, the statement follows from the induction hypothesis. (2) If $s^{-1}T \equiv T$, then $(s^*)^{-1}T \equiv T$ since $T \equiv \varepsilon^{-1}T \equiv s^{-1}T \equiv (ss)^{-1}T \equiv \ldots$, which implies $\Pi \smile T$ and $([\![T]\!], P') \in \Pi$. Therefore, the derivation proceeds by using (VC-Mem).

The other cases are similar. □

Corollary 26. If $\emptyset \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$, then $\emptyset \hookrightarrow \Pi' \vdash T \rightsquigarrow P$.

Lemma 27. For arbitrary Π, Π', T and P, if $\Pi \overset{P}{\bowtie} T$ and $\Pi \hookrightarrow \Pi' \vdash T \rightsquigarrow P$, then $\Pi \cap \Pi' \smile T$.

Proof. Straightforward induction on the derivation of $\Pi \hookrightarrow \Pi' \vdash T \rightsquigarrow P$. □

Corollary 28. For arbitrary T and P, if $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$ and $\Pi \overset{P}{\bowtie} T$, then $\Pi \cap \Pi' \smile T$.

Lemma 29. For arbitrary T and $s \in [\![T]\!]$, if $\emptyset \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$, then $s \rhd P \iff s \in [\![\Gamma(P)]\!]$.

Proof. We prove the following stronger statement: for arbitrary T and $s \in [\![T]\!]$, if $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$ and $\Pi \overset{P}{\bowtie} T$ hold, then $s \rhd P \iff s \in [\![\Gamma(P) \mid ((\Pi \cap \Pi')(T) \cap novar(P))]\!]$

The proof proceeds by induction on the derivation of $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$. We perform case analysis on the final rule in the derivation of $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$.

Case (RP-Mem):
Immediately follows from lemma 12, since $s \rhd P \iff s \in novar(P) = \{([\![T]\!], P)\}(T) \cap novar(P)$.

Case (RP-Const):
Case (RP-Rep):
Immediately follows from lemma 12 and $\Gamma = \emptyset$.

Case (RP-Seq-Choice):
If $\Pi \overset{(P_1|P_2)P_3}{\bowtie} T$, then $\Pi \overset{(P_1P_3)|(P_2P_3)}{\bowtie} T$ also holds. Therefore the statement follows from the induction hypothesis since $\Gamma((P_1 \mid P_2)P_3) \equiv \Gamma((P_1P_3) \mid (P_2P_3))$.

Case (RP-Seq-Const):
Case (RP-Seq-Seq):
Similar to (RP-Seq-Choice).

Case (RP-Bind):
If $\Pi \overset{P'}{\bowtie} T$, then $\Pi \overset{P'}{\bowtie} T$ also holds. By the definition of (RP-Bind), $\Pi \vdash T \rightsquigarrow P' \Rightarrow \Gamma'; \Pi'$ where $\Gamma = \Gamma', x : \Gamma'(P') \cap T$. By the induction hypothesis, if $s \in [\![T]\!]$, then $s \rhd P' \iff s \in [\![\Gamma'(P') \mid ((\Pi \cap \Pi')(T) \cap novar(P'))]\!] \iff s \in [\![T \cap \Gamma'(P') \mid ((\Pi \cap \Pi')(T) \cap novar(P'))]\!]$. Finally, it is trivial that $s \rhd P' \iff s \rhd x \texttt{ as } P'$.

Case (RP-Seq-Bind):
Suppose $\Pi \overset{(x \texttt{ as } P_1)P_2}{\bowtie} T$. Then, $\Pi \overset{P_1(y \texttt{ as } P_2)}{\bowtie} T$ also holds. Therefore $\Pi \hookrightarrow \Pi' \vdash T \rightsquigarrow P_1(y \texttt{ as } P_2)$ by lemma 25. By the definition of (RP-Seq-Bind), $\Pi \vdash T \rightsquigarrow P_1(y \texttt{ as } P_2) \Rightarrow \Gamma', y : T'; \Pi'$ where $\Gamma = \Gamma', x : \Gamma(P_1) \cap (TT'^{-1})$.

We can show that $T' \leq \Gamma'(P_2)$ holds by straightforward induction on the derivation of $\Pi \vdash T \rightsquigarrow P_1(y \texttt{ as } P_2) \Rightarrow \Gamma'; \Pi'$, therefore,

$$
\begin{array}{rrl}
 & & s \rhd P_1(y \texttt{ as } P_2) \Rightarrow \theta \\
 & & \text{by the induction hypothesis} \\
\iff & s \in & [\![\Gamma'(P_1)(T' \cap \Gamma'(P_2)) \mid ((\Pi \cap \Pi')(T) \cap novar(P_1P_2))]\!] \\
\iff & s \in & [\![(\Gamma(P_1)'T' \cap \Gamma(P_1)'\Gamma'(P_2)) \mid ((\Pi \cap \Pi')(T) \cap novar(P_1P_2))]\!] \\
\iff & s \in & [\![(\Gamma(P_1)'T' \cap \Gamma(P_1)'\Gamma'(P_2)) \mid ((\Pi \cap \Pi')(T) \cap novar(P_1P_2))]\!] \\
\Rightarrow & s \in & [\![\Gamma(P_1)'\Gamma'(P_2) \mid ((\Pi \cap \Pi')(T) \cap novar(P_1P_2))]\!] \\
 & & \text{because } s \in [\![T]\!] \subseteq [\![TT'^{-1}T']\!] \subseteq [\![TT'^{-1}\Gamma'(P_2)]\!] \\
\Rightarrow & s \in & [\![TT'^{-1}\Gamma'(P_2) \cap \Gamma(P_1)'\Gamma'(P_2) \mid ((\Pi \cap \Pi')(T) \cap novar(P_1P_2))]\!] \\
\iff & s \in & [\![(\Gamma(P_1)' \cap TT'^{-1})\Gamma'(P_2) \mid ((\Pi \cap \Pi')(T) \cap novar(P_1P_2))]\!] \\
\iff & s \in & [\![(\Gamma(P_1) \cap TT'^{-1})\Gamma(P_2) \mid ((\Pi \cap \Pi')(T) \cap novar(P_1P_2))]\!] \\
 & = & [\![\Gamma((x \texttt{ as } P_1)P_2)) \mid ((\Pi \cap \Pi')(T) \cap novar(P_1P_2))]\!]
\end{array}
$$

Conversely, if $s \in [\![\Gamma((x \texttt{ as } P_1)P_2)) \mid ((\Pi \cap \Pi')(T) \cap novar(P_1P_2))]\!]$, then $s \in novar((x \texttt{ as } P_1)P_2) \iff s \rhd (x \texttt{ as } P_1)P_2$ by lemma 12 and 23.

Case (RP-Choice):

By the definition of (RP-Choice), $\Pi \vdash T \rightsquigarrow P_1 \Rightarrow \Gamma_1; \Pi_1$ and $\Pi \vdash T \cap \overline{novar(P_1)} \rightsquigarrow P_2 \Rightarrow \Gamma_2; \Pi_2$ where $\Gamma(x) = \Gamma_1(x) \mid \Gamma_2(x)$ $(\forall x \in dom(\Gamma))$, $\Pi' = \Pi_1 \cup \Pi_2$.

We have two possibilities in deriving $s \rhd P_1 \mid P_2$. (1) $s \rhd P_1$: in this case, we can conclude, from the induction hypothesis, that $s \rhd P_1 \iff s \in [\![\Gamma_1(P_1) \mid ((\Pi \cap \Pi_1)(T) \cap novar(P_1))]\!]$. (2) $s \rhd P_1 \Rightarrow \bot$ and $s \rhd P_2 \Rightarrow \theta$: in this case, $s \notin [\![novar P_1]\!]$ by lemma 12, thus $s \in [\![T \cap \overline{novar(P_1)}]\!]$. Therefore, by the induction hypothesis, $s \rhd P_2 \Rightarrow \theta \iff s \in [\![\Gamma_2(P_2) \mid ((\Pi \cap \Pi_2)(T) \cap novar(P_2))]\!]$.

Consequently,

$$
\begin{aligned}
& s \rhd P_1 \mid P_2 \Rightarrow \theta \\
\iff & s \in [\![\Gamma_1(P_1) \mid \Gamma_2(P_2) \mid ((\Pi \cap \Pi_1)(T) \cap novar(P_1)) \mid ((\Pi \cap \Pi_2)(T) \cap novar(P_2))]\!] \\
\iff & s \in [\![\Gamma_1(P_1) \mid \Gamma_2(P_2) \mid ((\Pi \cap (\Pi_1 \cup \Pi_2))(T) \cap novar(P_1)) \mid ((\Pi \cap (\Pi_1 \cup \Pi_2))(T) \cap novar(P_2))]\!] \\
\iff & s \in [\![\Gamma_1(P_1) \mid \Gamma_2(P_2) \mid ((\Pi \cap (\Pi_1 \cup \Pi_2))(T) \cap (novar(P_1) \mid novar(P_2)))]\!] \\
\iff & s \in [\![\Gamma_1(P_1) \mid \Gamma_2(P_2) \mid ((\Pi \cap \Pi')(T) \cap (novar(P_1 \mid P_2)))]\!] \\
\Rightarrow & s \in [\![\Gamma_1(P_1) \mid \Gamma_1(P_2) \mid \Gamma_2(P_1) \mid \Gamma_2(P_2) \mid ((\Pi \cap \Pi')(T) \cap (novar(P_1 \mid P_2)))]\!] \\
\iff & s \in [\![\Gamma(P_1) \mid \Gamma(P_2) \mid ((\Pi \cap \Pi')(T) \cap (novar(P_1 \mid P_2)))]\!]
\end{aligned}
$$

Conversely, by lemma 12 and 23, if $s \in [\![\Gamma_1(P_2) \mid \Gamma_2(P_1) \mid ((\Pi \cap \Pi')(T) \cap (novar(P_1 \mid P_2)))]\!]$ then $s \in [\![novar(P_2) \mid novar(P_1)]\!] \iff s \rhd P_1 \mid P_2 \Rightarrow \theta (\neq \bot)$.

Case (RP-Seq-Rep):

By the definition of (RP-Seq-Rep), $\Pi \uplus ([\![T]\!], P_1^* P_2) \vdash T \rightsquigarrow (P_1 P_1^*) P_2 \Rightarrow \Gamma_1; \Pi_1$, $\Pi \vdash T \cap \overline{novar(P_1 P_1^*)} \rightsquigarrow P_2 \Rightarrow \Gamma_2; \Pi_2$ where $\Pi' = \Pi_1 \cup \Pi_2$, $\Gamma(P_1^* P_2) = \Gamma_1(P_1^* P_2) \mid \Gamma_2(P_1^* P_2)$.

There are two possibilities with respect to Π_1. (1) $([\![T]\!], P_1^* P_2) \notin \Pi_1$: in this case, similar discussion to the case of (RP-Choice) holds. (2) $([\![T]\!], P_1^* P_2) \in \Pi_1$: by lemma 27, $(\Pi \uplus ([\![T]\!], P_1^* P_2)) \cap \Pi_1 \smile T$ holds. Thus, by the definition of $\Pi \smile T$, (i) $((\Pi \uplus \{([\![T]\!], P_1^* P_2)\}) \cap \Pi_1)(T) = novar(P_1^* P_2)$ and (ii) $(\Pi \cap \Pi_1)(T) = \emptyset$ hold.

Suppose $s \rhd P_1^* P_2$. Then, if $s \rhd P_1 P_1^* P_2$ then

$$
\begin{aligned}
& s \rhd P_1 P_1^* P_2 \\
\iff\ s \in\ & [\![\Gamma_1(P_1 P_1^* P_2) \mid (((\Pi \uplus \{([\![T]\!], P_1^* P_2)\}) \cap \Pi_1)(T) \cap novar(P_1 P_1^* P_2))]\!] \\
=\ & [\![\Gamma_1(P_1 P_1^* P_2) \mid (novar(P_1^* P_2) \cap novar(P_1 P_1^* P_2))]\!]
\end{aligned}
$$

by the induction hypothesis and (i), (ii).

If $s \rhd P_1 P_1^* P_2 \Rightarrow \bot$ and $s \rhd P_2$, then by lemma 12, $s \notin [\![novar P_1 P_1^* P_2]\!]$. Therefore $s \in [\![T \cap \overline{novar P_1 P_1^* P_2}]\!]$. Thus, by the induction hypothesis, $s \rhd P_2 \Rightarrow \theta \iff s \in [\![\Gamma_2(P_2) \mid ((\Pi \cap \Pi_2)(T) \cap novar(P_2))]\!]$.

Consequently,

$$
\begin{aligned}
& s \rhd P_1^* P_2 \Rightarrow \theta \\
\iff & s \in [\![\Gamma_1(P_1 P_1^* P_2) \mid \Gamma_2(P_2) \mid (novar(P_1^* P_2)) \cap (novar(P_1 P_1^* P_2)) \mid ((\Pi \cap \Pi_2)(T) \cap novar(P_2))]\!] \\
& \text{because } (novar(P_1^* P_2)) \cap (novar(P_1 P_1^* P_2)) \leq novar(P_1^* P_2) \\
\Rightarrow & s \in [\![\Gamma_1(P_1 P_1^* P_2) \mid \Gamma_2(P_2) \mid novar(P_1^* P_2) \mid ((\Pi \cap \Pi_2)(T) \cap novar(P_2))]\!] \\
& \text{since } s \rhd P_1^* P_2,\ s \in [\![novar(P_1^* P_2)]\!] \text{ by lemma 12} \\
\iff & s \in [\![\Gamma_1(P_1 P_1^* P_2) \mid \Gamma_2(P_2) \mid ((\Pi \cap \Pi_2)(T) \cap novar(P_2))]\!]
\end{aligned}
$$

Conversely, if $s \in [\![\Gamma_1(P_1 P_1^* P_2) \mid \Gamma_2(P_2) \mid ((\Pi \cap \Pi_2)(T) \cap novar(P_2))]\!]$, then by lemma 12 and 23, $s \in [\![novar(P_1^* P_2)]\!]$. Therefore $s \rhd P_1^* P_2$.

□

Proof. (Soundness and Completeness of Pattern-matching)

We prove the following stronger statement: if $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$ and $\Pi \overset{P}{\bowtie} T$, then $dom(\Gamma) = var(P)$ and for each $x \in dom(\Gamma)$, $[\![\Gamma(x)]\!] = R(x, T, P) \setminus R^{mem}(x, T, \mathit{\Pi} \cap \mathit{\Pi}')$.

The proof proceeds by induction on the derivation of $\Pi \vdash T \rightsquigarrow P \Rightarrow \Gamma; \Pi'$. We proceed by case analysis on the final rule in the derivation.

Case (RP-Mem):
Immediately follows from the definition of (RP-Mem): $R(x, T, P) \setminus R^{mem}(x, T, \mathit{\Pi} \cap \{([\![T]\!], P)\}) = R(x, T, P) \setminus R(x, T, P) = \emptyset = \Gamma(x)$.

Case (RP-Const):
Case (RP-Rep):
Trivial since $dom(\Gamma) = \emptyset$.

Case (RP-Seq-Choice):
Suppose $\Pi \overset{(P_1|P_2)P_3}{\bowtie} T$. Then $\Pi \overset{P_1P_3|P_2P_3}{\bowtie} T$ also holds. By the induction hypothesis, $\Gamma(x) = R(x, T, P_1P_3 \mid P_2P_3) \setminus R^{mem}(x, T, \mathit{\Pi} \cap \mathit{\Pi}')$. It is obvious that $R(x, T, (P_1 \mid P_2)P_3) = R(x, T, P_1P_3 \mid P_2P_3)$.

Case (RP-Seq-Const):
Case (RP-Seq-Seq):
Similar to (RP-Seq-Choice).

Case $\Pi \vdash T \rightsquigarrow x \texttt{ as } P' \Rightarrow \Gamma; \Pi'$:
By the definition of $\Pi \overset{x \texttt{ as } P'}{\bowtie} T$, $(\Pi \cap \Pi')(T) = \emptyset$ hence $R^{mem}(x, T, x \texttt{ as } P') = \emptyset$.
By the definition of (RP-Seq-Choice), $\Pi \vdash T \rightsquigarrow P' \Rightarrow \Gamma'; \Pi'$ where $\Gamma = \Gamma' \uplus \{x : \Gamma'(P') \cap T\}$. Therefore $\Gamma(x) = \Gamma'(P') \cap T \leq novar(P') \cap T$. On the other hand, $R(x, T, x \texttt{ as } P') = \{s \mid s \in [\![T]\!] \wedge s \triangleright P'\} = [\![novar(P') \cap T]\!]$. Thus $[\![\Gamma(x)]\!] \subseteq R(x, T, x \texttt{ as } P')$. Conversely, if $s \in [\![novar(P') \cap T]\!]$, then by lemma 29, $s \in \Gamma(P')$. Therefore $\Gamma(x) = R(x, T, x \texttt{ as } P')$.
For other $y \in var(P')$, the statement immediately follows from the induction hypothesis.

Case (RP-Seq-Bind):
By the definition of $\Pi \overset{(x \texttt{ as } P_1)P_2}{\bowtie} T$, $(\Pi \cap \Pi')(T) = \emptyset$. As a result, $R^{mem}(x, T, \mathit{\Pi} \cap \mathit{\Pi}') = \emptyset$.
Suppose $\Pi \vdash T \rightsquigarrow P_1(y \texttt{ as } P_2) \Rightarrow \Gamma'; \Pi'$. Then, by the induction hypothesis, $[\![\Gamma'(y)]\!] = R(y, T, P_1(y \texttt{ as } P_2)) \setminus R^{mem}(y, T, \mathit{\Pi} \cap \mathit{\Pi}')$. Meanwhile, since y does not appear in P_1P_2 and in

Π, $R^{mem}(y, T, \mathit{\Pi} \cap \mathit{\Pi}') = \emptyset$. Consequently,

$$\begin{array}{rcl} R(x, T, (x \texttt{ as } P_1)P_2) & = & \{\theta(x) \mid s \in T \wedge s \rhd (x \texttt{ as } P_1)P_2 \Rightarrow \theta(\neq \bot)\} \\ & & \text{by the definition of (M-Seq-Bind)} \\ & = & \{t \mid tt' \in [\![T]\!] \wedge t' \in R(y, T, P_1(y \texttt{ as } P_2)) \wedge tt' \rhd P_1(y \texttt{ as } P_2)\} \\ & & R(y, T, P_1(y \texttt{ as } P_2)) = [\![\Gamma'(y)]\!] \text{ by the above discussion} \\ & = & \{t \mid tt' \in [\![T]\!] \wedge t' \in [\![\Gamma'(y)]\!] \wedge tt' \rhd P_1(y \texttt{ as } P_2)\} \\ & & \text{by lemma 7} \\ & = & \{t \mid tt' \in [\![T]\!] \wedge t' \in [\![\Gamma'(y)]\!] \wedge t \rhd P_1 \wedge t' \rhd y \texttt{ as } P_2\} \\ & & \text{by lemma 29} \\ & = & \{t \mid tt' \in [\![T]\!] \wedge t' \in [\![\Gamma'(y)]\!] \wedge t \in [\![\Gamma'(P_1)]\!]\} \\ & = & \{t \mid t \in [\![T\Gamma'(y)^{-1}]\!] \wedge t \in [\![\Gamma(P_1)]\!]\} \\ & = & [\![\Gamma'(P_1) \cap T\Gamma'(y)^{-1}]\!] \\ & = & [\![\Gamma(x)]\!] \end{array}$$

Case (RP-Choice):

By the definition of (RP-Choice), $\Pi \vdash T \rightsquigarrow P_1 \Rightarrow \Gamma_1; \Pi_1$ and $\Pi \vdash T \cap \overline{novar(P_1)} \rightsquigarrow P_2 \Rightarrow \Gamma_2; \Pi_2$ where $\Gamma(x) = \Gamma_1(x) \mid \Gamma_2(x)$ $(\forall x \in dom(\Gamma))$, $\Pi' = \Pi_1 \cup \Pi_2$. Therefore,

$$\begin{array}{rcl} R(x, T, P_1 \mid P_2) & = & \{\theta(x) \mid s \in [\![T]\!] \wedge s \rhd P_1 \mid P_2 \Rightarrow \theta(\neq \bot)\} \\ & = & \{\theta(x) \mid s \in [\![T]\!] \wedge (s \rhd P_1 \Rightarrow \theta(\neq \bot) \vee (s \rhd P_1 \Rightarrow \bot \wedge s \rhd P_2 \Rightarrow \theta(\neq \bot))\} \\ & & \text{by lemma 12} \\ & = & \{\theta(x) \mid s \in [\![T]\!] \wedge (s \rhd P_1 \Rightarrow \theta(\neq \bot) \vee (s \notin [\![novar(P_1)]\!] \wedge s \rhd P_2 \Rightarrow \theta(\neq \bot))\} \\ & = & \{\theta(x) \mid (s \in [\![T]\!] \wedge s \rhd P_1 \Rightarrow \theta(\neq \bot)) \vee (s \in [\![T \cap \overline{novar(P_1)}]\!] \wedge s \rhd P_2 \Rightarrow \theta(\neq \bot))\} \\ & = & R(x, T, P_1) \cup R(x, T \cap \overline{novar(P_1)}, P_2) \end{array}$$

On the other hand,

$$\begin{array}{rcl} [\![\Gamma(x)]\!] & = & [\![\Gamma_1(x) \mid \Gamma_2(x)]\!] \\ & & \text{by the induction hypothesis} \\ & = & (R(x, T, P_1) \setminus R^{mem}(x, T, \mathit{\Pi} \cap \mathit{\Pi}_1)) \cup (R(x, T \cap \overline{novar(P_1)}, P_2) \setminus R^{mem}(x, T, \mathit{\Pi} \cap \mathit{\Pi}_2)) \\ & = & (R(x, T, P_1) \cup R(x, T \cap \overline{novar(P_1)}, P_2)) \setminus (R^{mem}(x, T, \mathit{\Pi} \cap \mathit{\Pi}_1) \cup R^{mem}(x, T, \mathit{\Pi} \cap \mathit{\Pi}_2)) \\ & = & (R(x, T, P_1) \cup R(x, T \cap \overline{novar(P_1)}, P_2)) \setminus R^{mem}(x, T, \mathit{\Pi} \cap (\mathit{\Pi}_1 \cup \mathit{\Pi}_2)) \\ & = & R(x, T, P_1 \mid P_2) \setminus R^{mem}(x, T, \mathit{\Pi} \cap \mathit{\Pi}') \end{array}$$

Case (RP-Seq-Rep):

By the definition of (RP-Seq-Rep), $\Pi \uplus \{([\![T]\!], P_1^* P_2)\} \vdash T \rightsquigarrow (P_1 P_1^*)P_2 \Rightarrow \Gamma_1; \Pi_1$ and $\Pi \vdash T \cap \overline{novar(P_1 P_1^*)} \rightsquigarrow P_2 \Rightarrow \Gamma_2; \Pi_2$ where $\Pi' = \Pi_1 \cup \Pi_2$.

There are two possibilities with respect to Π_1. (1) $([\![T]\!], P_1^* P_2) \notin \Pi_1$: in this case, similar discussion to the case of (RP-Choice) holds. (2) $([\![T]\!], P_1^* P_2) \in \Pi_1$: in this case, $(\Pi \uplus ([\![T]\!], P_1^* P_2)) \cap \Pi_1 \smile T$ by lemma 27, therefore, by definition, (i) $R^{mem}(x, T, \mathit{\Pi} \cap \mathit{\Pi}_1) = \emptyset$ and (ii) $novar(P_1^*)^{-1} T \equiv T$.

Since $R^{mem}(x, T, \mathit{\Pi} \uplus ([\![T]\!], P_1^* P_2) \cap \mathit{\Pi}_1) \supseteq R(x, T, P_1^* P_2)$, we can conclude that, by the induction hypothesis,

$$[\![\Gamma_1(x)]\!] = R(x, T, P_1 P_1^* P_2) \setminus R^{mem}(x, T, \mathit{\Pi} \uplus ([\![T]\!], P_1^* P_2) \cap \mathit{\Pi}_1) = \emptyset$$

holds for Γ_1. Thus $\Gamma(x) = \Gamma_2(x)$. Furthermore, $R(x, T, P_1^* P_2) \setminus R^{mem}(x, T, \Pi \cap (\Pi_1 \cup \Pi_2)) = R(x, T, P_1^* P_2) \setminus R^{mem}(x, T, \Pi \cap \Pi_2)$ by (i).

Suppose $s \rhd P_2 \Rightarrow \theta(\neq \bot)$ and $s \rhd P_1^* P_2 \Rightarrow \theta'(\neq \bot)$. Since we assumed patterns being standardized, which implies $\varepsilon \notin novar(P_1)$, (a) $s \rhd P_1 P_1^* P_2 \Rightarrow \bot$ and (b) $\theta'(x) = \theta(x)$ are concluded by the definition of (M-Seq-Rep), (M-Choice) etc.

By (b), if $s \rhd P_2 \Rightarrow \theta$, then $s \rhd P_1^* P_2 \Rightarrow \theta$. Thus $R(x, T, P_2) \subseteq R(x, T, P_1^* P_2)$. Meanwhile, $R(x, T, P_1^* P_2) \subseteq R(x, novar(P_1^*)^{-1} T, P_2)$ by lemma 24. Therefore $R(x, T, P_1^* P_2) = R(x, T, P_2)$ since (ii) suggests $novar(P_1^*)^{-1} T \equiv T$.

By (a) and lemma 12,

$$\begin{aligned} R(x, T, P_2) &= \{\theta(x) \mid s \in [\![T]\!] \wedge s \rhd P_2 \Rightarrow \theta(\neq \bot)\} \\ &= \{\theta(x) \mid s \in [\![T]\!] \wedge s \notin [\![novar(P_1 P_1^* P_2)]\!] \wedge s \rhd P_2 \Rightarrow \theta(\neq \bot)\} \\ &= R(x, T \cap \overline{novar(P_1 P_1^* P_2)}, P_2) \end{aligned}$$

holds. From the above-mentioned discussion,

$$R(x, T, P_1^* P_2) \setminus R^{mem}(x, T, \Pi \cap \Pi_2) = R(x, T \cap \overline{novar(P_1 P_1^* P_2)}, P_2) \setminus R^{mem}(x, T, \Pi \cap \Pi_2)$$

is concluded, which exactly agrees with $[\![\Gamma_2(x)]\!]$ (therefore $[\![\Gamma(x)]\!]$) by the induction hypothesis.

□

References

[1] Robert Harper. Proof-directed debugging. *Journal of Functional Programming*, 9(4):463–469, 1999.

[2] Naoshi Tabuchi, Eijiro Sumii, and Akinori Yonezawa. Regular expression types for strings in a text processing language (extended abstract). In *Workshop on Types in Programming*, 2002. To Appear.

[3] Atsushi Yamamoto. Regular expressions and pattern matching algorithms as functions. http://aglaia.c.u-tokyo.ac.jp/~yamamoto/regex_function/, 2001. In Japanese.