Lezione 32 MSC
Petri Nets Semantics of Some Process Algebras

Roberto Gorrieri

Giving net semantics to a process algebra

- **Operational**: adaptation of the SOS technique we used for giving LTS semantics
- **Denotational**: novel approach based on the syntax (i.e., places are syntactic objects)
- The two semantics are coherent: in many cases they coincide
- Of the many process algebras in the hierarchy, we choose a simple case: **BPP**

BPP: syntax

$$
\begin{aligned}
s &::= \mathbf{0} \mid \mu.p \mid s+s \\
q &::= s \mid C & \textit{sequential processes} \\
p &::= q \mid p|p & \textit{parallel processes}
\end{aligned}
$$

- A constant C is equipped with a defining equation C = r, **with r in syntactic category s**.
- Constants are guarded by construction.
- **Finite calculus**: finitely many constants and finitely many actions can be used.
- **Asynchronous parallel composition** (no communication is allowed).

Sequential subterms

- For any p, the set of its sequential subterms *sub*(*p*) is defined by means of the auxiliary function *sub*(*p*,I), whose second parameter I is a set of already known constants, initially empty.
- **Theorem 5.9: For any p, sub(p) is finite**

$$sub(\mathbf{0},I) = \{\mathbf{0}\}$$

$$sub(\mu.p,I) = \{\mu.p\} \cup sub(p,I)$$

$$sub(p_1+p_2,I) = \{p_1+p_2\} \cup sub(p_1,I) \cup sub(p_2,I)$$

$$sub(C,I) = \begin{cases} \emptyset & C \in I, \\ \{C\} \cup sub(p, I \cup \{C\}) & C \notin I \wedge C \doteq p \end{cases}$$

$$sub(p_1 \mid p_2,I) = sub(p_1,I) \cup sub(p_2,I)$$

Places and decomposition function

- The set of all the places is S_{BPP} = sequential BPP processes, except **0**.
- The decomposition function dec

dec: BPP processes → finite multisets over S_{BPP}

$$dec(\mathbf{0}) = \emptyset \qquad dec(\mu.p) = \{\mu.p\}$$

$$dec(p+p') = \{p+p'\} \qquad dec(C) = \{C\}$$

$$dec(p\,|\,p') = dec(p) \oplus dec(p')$$

Lemma 5.2. *For any BPP process p,* $\bigcup_{s \in dom(dec(p))} sub(s) \subseteq sub(p)$.

Examples

$$dec(a.\mathbf{0} \,|\, (b.\mathbf{0} \,|\, a.\mathbf{0})) = dec(a.\mathbf{0}) \oplus dec(b.\mathbf{0} \,|\, a.\mathbf{0}) = a.\mathbf{0} \oplus dec(b.\mathbf{0}) \oplus dec(a.\mathbf{0}) =$$
$$= \{a.\mathbf{0}, b.\mathbf{0}, a.\mathbf{0}\} = 2 \cdot a.\mathbf{0} \oplus b.\mathbf{0}.$$

Example 5.12. Consider the semi-counter $SC \doteq inc.(SC \,|\, dec.\mathbf{0})$, discussed in Example 5.9.[4] Then, the decomposition $dec(SC)$ is $\{SC\}$, $dec(inc.(SC \,|\, dec.\mathbf{0})) = \{inc.(SC \,|\, dec.\mathbf{0})\}$ and $dec(SC \,|\, dec.\mathbf{0}) = \{SC, dec.\mathbf{0}\}$. □

- Function dec is not injective: for instance,

  dec(p|q) = dec(q|p)   dec(p|0) = dec(p)

  dec(p|(q|r)) = dec((p|q)|r)
- Function dec is surjective: for any finite multiset of places m we can find a BPP process p such that dec(p) = m.

Net transitions

- T_{BPP} is the set of all transitions that can be derived by the axiom and rules below

$$\text{(pref)}\ \frac{}{\{\mu.p\} \xrightarrow{\mu} dec(p)} \qquad \text{(cons)}\ \frac{dec(p) \xrightarrow{\mu} m}{\{C\} \xrightarrow{\mu} m}\ C \doteq p$$

$$(\text{sum}_1)\ \frac{dec(p) \xrightarrow{\mu} m}{\{p+q\} \xrightarrow{\mu} m} \qquad (\text{sum}_2)\ \frac{dec(q) \xrightarrow{\mu} m}{\{p+q\} \xrightarrow{\mu} m}$$

Proposition 5.14. *For any $t \in T_{BPP}$, we have that $|{}^\bullet t| = 1$*

Proof. By induction on the proof of t, according to the rules

Proposition 5.15. *For any $t \in T_{BPP}$ of the form $t = (\{p\}, \mu, m)$, we have that for any $s \in dom(m)$, $s \in sub(p)$.*

The whole net for BPP and Net(p)

- The whole net for BPP is N_{BPP} = (S_{BPP},Act,T_{BPP})

Definition 5.4. Let p be a BPP process. The P/T system statically associated with p is $Net(p) = (S_p, A_p, T_p, m_0)$, where $m_0 = dec(p)$ and

$$\begin{aligned} S_p &= [\![dom(m_0)\rangle \quad \text{computed in } N_{BPP}, \\ T_p &= \{t \in T_{BPP} \mid S_p[\![t\rangle\}, \\ A_p &= \{\mu \in Act \mid \exists t \in T_p, \mu = l(t))\}. \end{aligned}$$

□

Proposition 5.16. *For any $p \in \mathscr{P}_{BPP}$, $Net(p)$ is a statically reduced P/T net.*[5] □

Proposition 5.17. *If $dec(p) = dec(q)$, then $Net(p) = Net(q)$.* □

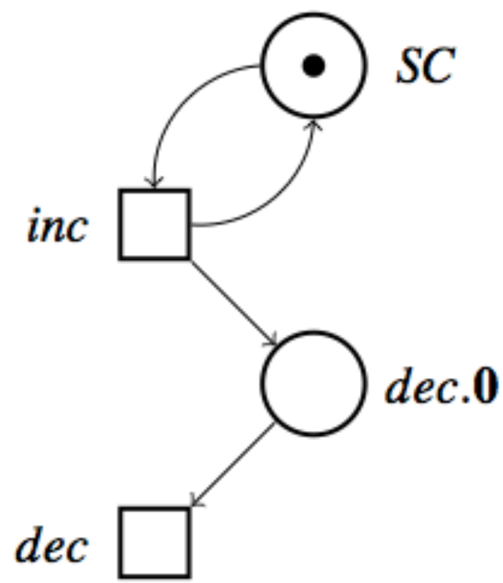
Proposition 5.18. *For any $p_1, p_2 \in \mathscr{P}_{BPP}$, if $Net(p_i) = (S_i, A_i, t_i, m_i)$ for $i = 1, 2$, then $Net(p_1 \mid p_2) = (S_1 \cup S_2, A_1 \cup A_2, T_1 \cup T_2, m_1 \oplus m_2)$.* □

---

[5] As for BPP nets, the notion of statically reachable subnet and dynamically reachable subnet coincide, $Net(p)$ is also dynamically reduced.

Example: semi-counter

Example 5.13. Consider again the semi-counter $SC \doteq inc.(SC \,|\, dec.\mathbf{0})$. Then, by axiom (pref), the net transition $\{inc.(SC \,|\, dec.\mathbf{0})\} \xrightarrow{inc} \{SC, dec.\mathbf{0}\}$ is derivable. By using this net transition as the premise for rule (cons), $\{SC\} \xrightarrow{inc} \{SC, dec.\mathbf{0}\}$ is also derivable. Finally, the net transition $\{dec.\mathbf{0}\} \xrightarrow{dec} \emptyset$ is also derivable by axiom (pref). The net for the semi-counter SC is outlined in Figure 5.8. Note that this net is unbounded, because there is no upper limit to the number of tokens that can be accumulated in place $dec.\mathbf{0}$. □



A BBP process generates a BPP net

Theorem 5.10. (Finite number of places) *For any BPP process p, the set S_p of its places statically reachable from $dom(dec(p))$ is finite.*

Proof. By induction on the static reachability relation $\Longrightarrow^*$ (Definition 3.9), we can prove that any place s that is statically reachable from $dom(dec(p))$ is a subterm of p. Since $sub(p)$ is finite by Theorem 5.9, the thesis follows trivially.

Theorem 5.11. (Finite P/T Petri net) *For any BPP process p, the P/T net reachable from p, $Net(p) = (S_p, A_p, T_p, dec(p))$, is finite.*

Proof. By Theorem 5.10, the set S_p of the places reachable from p is finite. By Proposition 5.14 the pre-set of any derivable transition is a singleton. By Lemma 4.2, the set T^q of transitions with pre-set $\{q\}$ is finite, for any $q \in S_p$. Hence, also $T_p = \bigcup_{q \in S_p} T^q$ must be finite, being a finite union of finite sets. □

Corollary 5.3. (BPP nets) *For any BPP process p, $Net(p)$ is a BPP net.*

Proof. By Theorem 5.11, $Net(p)$ is a finite P/T net. By Proposition 5.14, for any $t \in T_p$, we have $|{}^\bullet t| = 1$, i.e., $Net(p)$ is a BPP net. □

Representability Theorem (1)

- In the previous slide we have proved that only BBP nets can be the semantics of a BPP process
- Now we show that each BPP net is the semantics of some BPP process, up to net isomorphism: this is the representability theorem (below described for FNM/P/T nets)

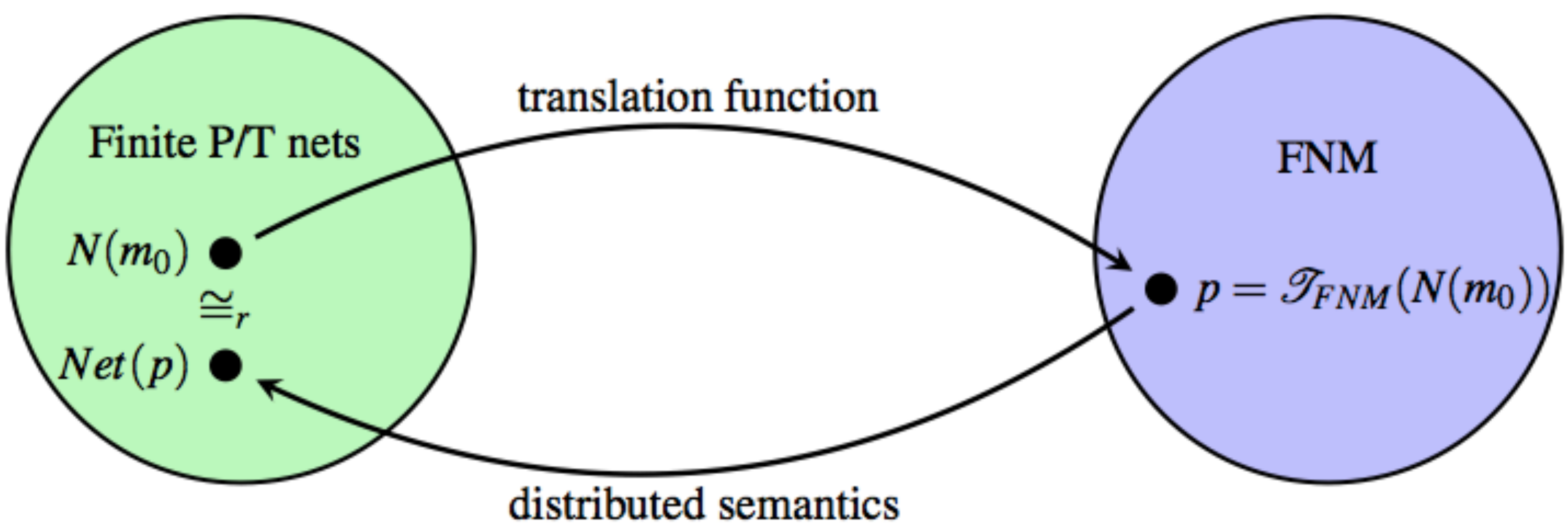


Fig. 1.8 Graphical description of the representability theorem for finite P/T nets and FNM

Representability Theorem (2)

Definition 5.5. (Translating BPP nets into BPP terms) Let $N(m_0) = (S, A, T, m_0)$ — with $S = \{s_1, \ldots, s_n\}$, $A \subseteq Act$, $T = \{t_1, \ldots, t_k\}$, and $l(t_j) = \mu_j$ — be a BPP net. Function $\mathscr{T}_{BPP}(-)$, from BPP nets to BPP processes, is defined as

$$\mathscr{T}_{BPP}(N(m_0)) = \underbrace{C_1|\cdots|C_1}_{m_0(s_1)}|\cdots|\underbrace{C_n|\cdots|C_n}_{m_0(s_n)}$$

where each C_i is equipped with a defining equation $C_i \doteq c_i^1 + \cdots + c_i^k$ (with $C_i \doteq \mathbf{0}$ if $k = 0$), and each summand c_i^j, for $j = 1, \ldots, k$, is equal to

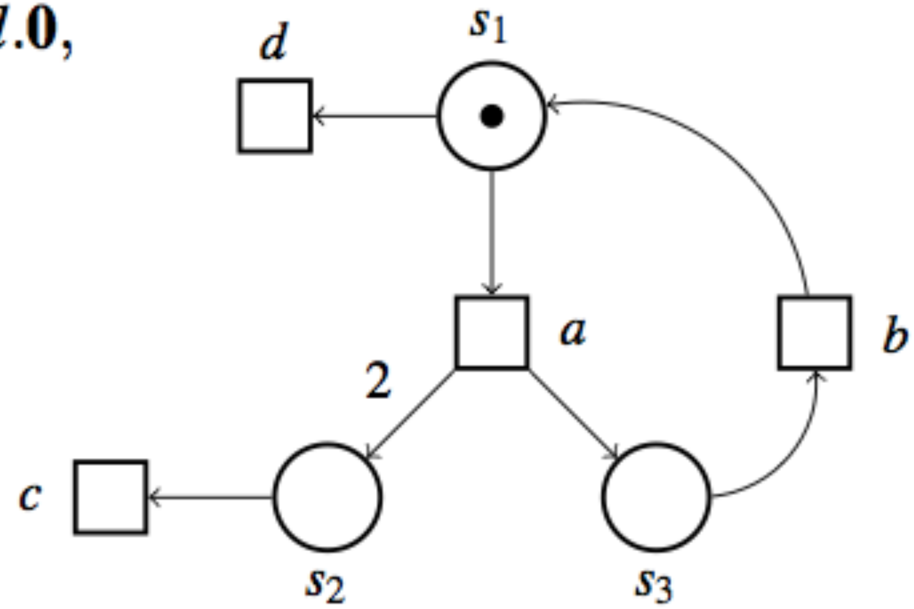
- $\mathbf{0}$, if $s_i \notin {}^\bullet t_j$;
- $\mu_j.\Pi_j$, if ${}^\bullet t_j = \{s_i\}$, where process Π_j is $\underbrace{C_1|\cdots|C_1}_{t_j^\bullet(s_1)}|\cdots|\underbrace{C_n|\cdots|C_n}_{t_j^\bullet(s_n)}$, meaning that

  $\Pi_j = \mathbf{0}$ if $t_j^\bullet = \emptyset$. □

Example

- A constant C_i for each place s_i
- $\mathscr{T}_{BPP}(N, s_1) = C_1$

$$C_1 \doteq a.(C_2 \,|\, C_2 \,|\, C_3) + \mathbf{0} + \mathbf{0} + d.\mathbf{0},$$
$$C_2 \doteq \mathbf{0} + \mathbf{0} + c.\mathbf{0} + \mathbf{0},$$
$$C_3 \doteq \mathbf{0} + b.C_1 + \mathbf{0} + \mathbf{0}.$$



Representability Theorem (3)

Theorem 5.13. (Representability theorem 3) *Let $N(m_0) = (S, A, T, m_0)$ be a (statically) reduced BPP net such that $A \subseteq Act$, and let $p = \mathscr{T}_{BPP}(N(m_0))$. Then, $Net(p)$ is isomorphic to $N(m_0)$.*

If $N(m_0)$ is not reduced, then Net(p) is isomorphic only to $Net_s(N(m_0))$.
For instance, this net with four places and two transitions

$$N(\{s_1\}) = (\{s_1, s_2, s_3, s_4\}, \{a\}, \{(\{s_1\}, a, \{s_2\}), (\{s_3\}, a, \{s_4\})\}, \{s_1\})$$

$$\begin{array}{llll} C_1 \doteq a.C_2 + \mathbf{0}, & & C_2 \doteq \mathbf{0} + \mathbf{0}, \\ C_3 \doteq \mathbf{0} + a.C_4, & & C_4 \doteq \mathbf{0} + \mathbf{0}, \end{array}$$

P = $\mathscr{T}_{BPP}$ (N, s1) = C1

Net(p) = $(\{C_1, C_2\}, \{a\}, \{(\{C_1\}, a, \{C_2\})\}, \{C_1\})$

which has two places and one transition only, i.e., it is isomorphic to the subnet of *N*({*s*1}) reachable from the initial marking {*s*1}.

Denotational Net Semantics

- Syntax driven definition: the places of the constructed net are syntactic objects, i.e., BPP sequential process terms (except **0**).
- Parametrized on a set of of constants that has already been found while scanning *p*; such a set is initially empty and it is used to avoid looping on recursive constants.

Definitions (1)

$$
\begin{array}{rcll}
[\![\mathbf{0}]\!]_I & = & (\emptyset,\emptyset,\emptyset,\theta) & \\
[\![\mu.p]\!]_I & = & (S,A,T,\{\mu.p\}) & \text{given } [\![p]\!]_I = (S',A',T',dec(p)) \text{ and where} \\
& & & S = \{\mu.p\}\cup S',\ A = \{\mu\}\cup A',\ T = \{(\{\mu.p\},\mu,dec(p))\}\cup T' \\
[\![p_1+p_2]\!]_I & = & (S,A,T,\{p_1+p_2\}) & \text{given } [\![p_i]\!]_I = (S_i,A_i,T_i,dec(p_i)) \text{ for } i=1,2, \text{ and where} \\
& & & S = \{p_1+p_2\}\cup S'_1\cup S'_2, \text{ with, for } i=1,2, \\
& & & S'_i = \begin{cases} S_i & \exists t\in T_i \text{ such that } t^\bullet(p_i)>0 \\ S_i\setminus\{p_i\} & \text{otherwise} \end{cases} \\
& & & A = A_1\cup A_2,\ T = T'\cup T'_1\cup T'_2, \text{ with, for } i=1,2, \\
& & & T'_i = \begin{cases} T_i & \exists t\in T_i\,.\,t^\bullet(p_i)>0 \\ T_i\setminus\{t\in T_i \mid {}^\bullet t(p_i)>0\} & \text{otherwise} \end{cases} \\
& & & T' = \{(\{p_1+p_2\},\mu,m) \mid (\{p_i\},\mu,m)\in T_i, i=1,2\}
\end{array}
$$

Definitions (2)

$$[\![C]\!]_I = (\{C\}, \emptyset, \emptyset, \{C\}) \qquad \text{if } C \in I$$

$$[\![C]\!]_I = (S, A, T, \{C\}) \qquad \text{if } C \notin I, \text{ given } C \doteq p \text{ and } [\![p]\!]_{I \cup \{C\}} = (S', A', T', dec(p))$$

$$A = A', S = \{C\} \cup S'', \text{ where}$$

$$S'' = \begin{cases} S' & \exists t \in T' . t^\bullet(p) > 0 \\ S' \setminus \{p\} & \text{otherwise} \end{cases}$$

$$T = \{(\{C\}, \mu, m) \mid (\{p\}, \mu, m) \in T'\} \cup T'' \text{ where}$$

$$T'' = \begin{cases} T' & \exists t \in T' . t^\bullet(p) > 0 \\ T' \setminus \{t \in T' \mid {}^\bullet t(p) > 0\} & \text{otherwise} \end{cases}$$

$$[\![p_1 \mid p_2]\!]_I = (S, A, T, m_0) \qquad \text{given } [\![p_i]\!]_I = (S_i, A_i, T_i, m_i) \text{ for } i = 1, 2, \text{ and where}$$

$$S = S_1 \cup S_2,\ A = A_1 \cup A_2,\ T = T_1 \cup T_2,\ m_0 = m_1 \oplus m_2$$

Example (1)

Example 4.6. Consider constant $B \doteq b.A$, where $A \doteq a.b.A$. By using the definitions in Table 4.5, $[\![A]\!]_{\{A,B\}} = (\{A\}, \emptyset, \emptyset, \{A\})$. Then, by action prefixing, $[\![b.A]\!]_{\{A,B\}} = (\{b.A, A\}, \{b\}, \{(\{b.A\}, b, \{A\})\}, \{b.A\})$. Again, by action prefixing, $[\![a.b.A]\!]_{\{A,B\}} = (\{a.b.A, b.A, A\}, \{a, b\}, \{(\{a.b.A\}, a, \{b.A\}), (\{b.A\}, b, \{A\})\}, \{a.b.A\})$. Now, the rule for constants ensures that

$$[\![A]\!]_{\{B\}} = (\{b.A, A\}, \{a, b\}, \{(\{A\}, a, \{b.A\}), (\{b.A\}, b, \{A\})\}, \{A\}).$$

Note that place $a.b.A$ has been removed, as no transition in $[\![a.b.A]\!]_{\{A,B\}}$ reaches that place. This net is depicted in Figure 4.2(a). By action prefixing,

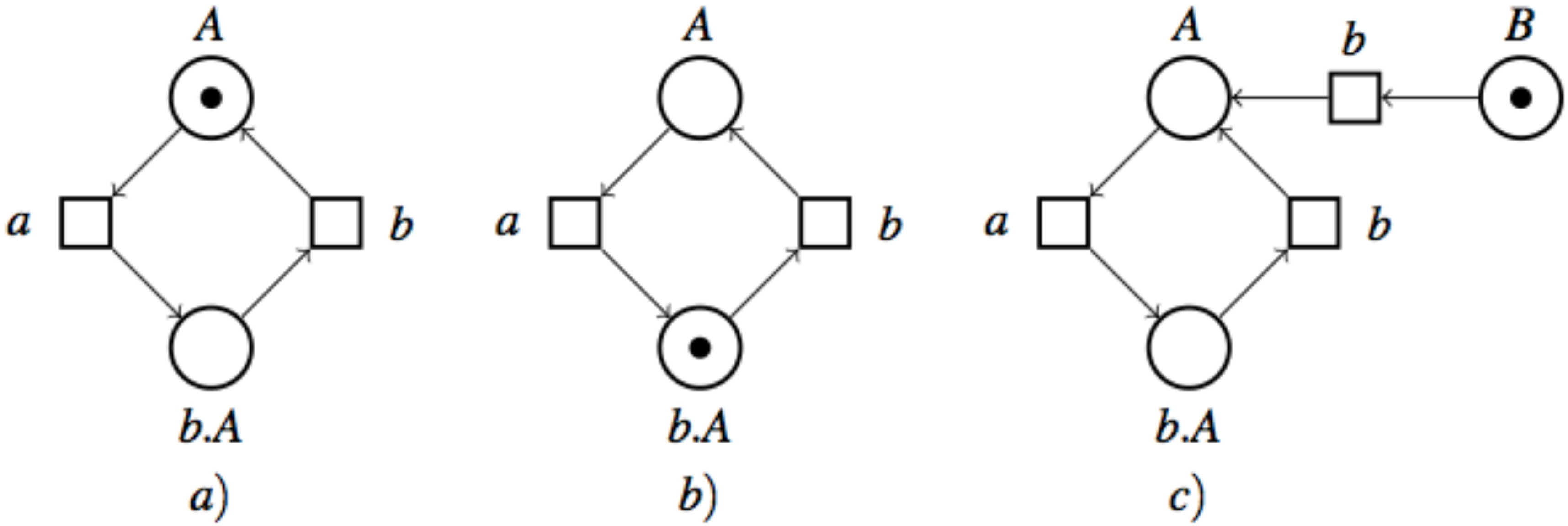
$$[\![b.A]\!]_{\{B\}} = (\{b.A, A\}, \{a, b\}, \{(\{A\}, a, \{b.A\}), (\{b.A\}, b, \{A\})\}, \{b.A\}),$$

i.e., this operation changes only the initial marking, but does not affect the underlying net! This net in outlined in Figure 4.2(b). Finally,

$$[\![B]\!]_{\emptyset} = (\{B, b.A, A\}, \{a, b\}, \{(\{B\}, b, \{A\}), (\{A\}, a, \{b.A\}), (\{b.A\}, b, \{A\})\}, \{B\}).$$

Note that place $b.A$ has been kept, because there is a transition in the net $[\![b.A]\!]_{\{B\}}$ that reaches that place. Figure 4.2(c) shows this net. □

Example (2)



Example (3)

$B|b.A|C|C$, where $B = b.A$, $A = a.b.A$ and $C = a.C$

$$[\![B\,|\,b.A]\!]_\emptyset = $$
$$= (\{B, b.A, A\}, \{a, b\}, \{(\{B\}, b, \{A\}), (\{A\}, a, \{b.A\}), (\{b.A\}, b, \{A\})\}, \{B, b.A\}),$$

$$[\![C]\!]_\emptyset = (\{C\}, \{a\}, \{(\{C\}, a, \{C\})\}, \{C\})$$

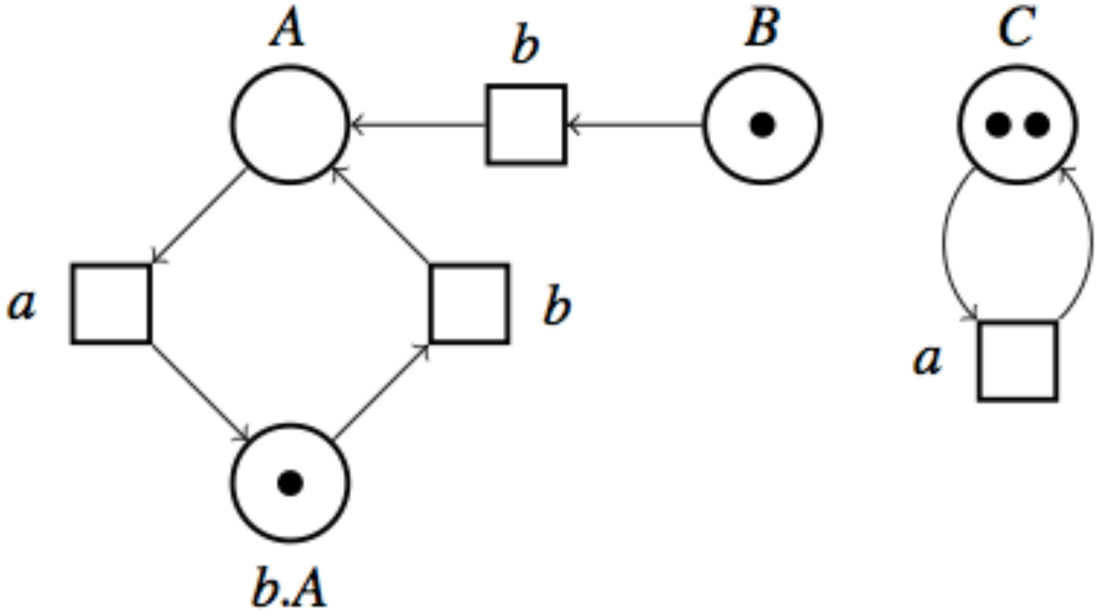
$$[\![C\,|\,C]\!]_\emptyset = (\{C\}, \{a\}, \{(\{C\}, a, \{C\})\}, \{C, C\})$$

And the whole net for $B\,|\,b.A\,|\,C\,|\,C$ is $[\![B\,|\,b.A\,|\,C\,|\,C]\!]_\emptyset = (S, A, T, m_0)$, where

$$S = \{B, b.A, A, C\},$$
$$A = \{a, b\},$$
$$T = \{(\{B\}, b, \{A\}), (\{A\}, a, \{b.A\}), (\{b.A\}, b, \{A\}), (\{C\}, a, \{C\})\},$$
$$m_0 = \{B, b.A, C, C\}.$$

Example (4)

- The resulting net is depicted below:



Example (5)

- Semi-counter $SC \doteq inc.(SC \mid dec.\mathbf{0})$.

$[\![SC]\!]_{\{SC\}} = (\{SC\}, \emptyset, \emptyset, \{SC\})$, and
$[\![dec.\mathbf{0}]\!]_{\{SC\}} = (\{dec.\mathbf{0}\}, \{dec\}, \{(\{dec.\mathbf{0}\}, dec, \emptyset)\}, \{dec.\mathbf{0}\})$.
Therefore, the net $[\![SC \mid dec.\mathbf{0}]\!]_{\{SC\}}$ is
$(\{SC, dec.\mathbf{0}\}, \{dec\}, \{(\{dec.\mathbf{0}\}, dec, \emptyset)\}, \{SC, dec.\mathbf{0}\})$.
The net $[\![inc.(SC \mid dec.\mathbf{0})]\!]_{\{SC\}}$ is
$(\{inc.(SC \mid dec.\mathbf{0}), SC, dec.\mathbf{0}\}, \{inc, dec\}, \{(\{inc.(SC \mid dec.\mathbf{0})\}, inc, \{SC, dec.\mathbf{0}\}),$
$(\{dec.\mathbf{0}\}, dec, \emptyset)\}, \{inc.(SC \mid dec.\mathbf{0})\})$.
Finally, the net $[\![SC]\!]_{\emptyset}$ is
$(\{SC, dec.\mathbf{0}\}, \{inc, dec\}, \{(\{SC\}, inc, \{SC, dec.\mathbf{0}\}), (\{dec.\mathbf{0}\}, dec, \emptyset)\}, \{SC\})$,

which is exactly the net on slide 9.

Coherence of the two net semantics

- The observation of the previous example holds in general: **operational and denotational net semantics coincide**

Theorem 5.14. (Operational and denotational semantics coincide) *For any BPP process p*, $Net(p) = [\![p]\!]_{\emptyset}$.