Free Choice Nets over Distributed Alphabets

Ramchandra Phawade

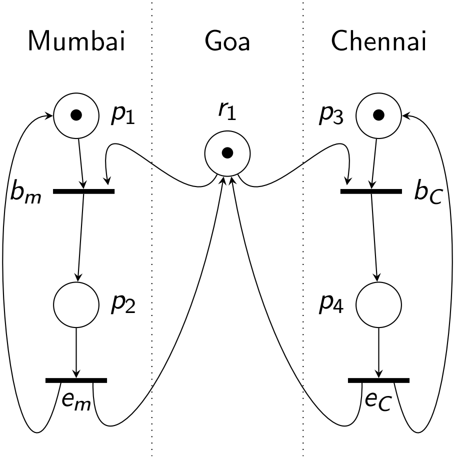
IIT Dharwad, India

July 20, 2018

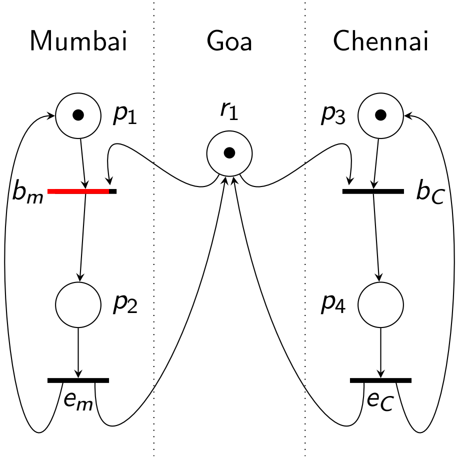
(With Kamal Lodaya)

FM Update Meeting 2018, BITS Pilani, Goa Campus

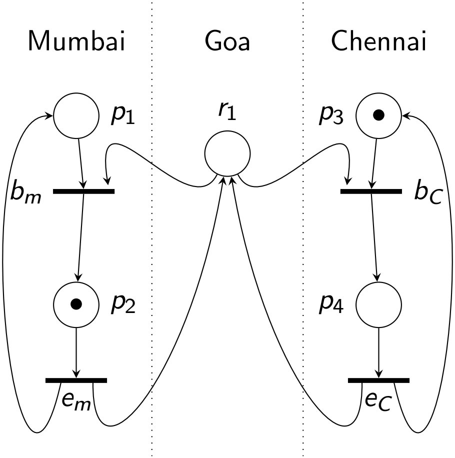
Example: Three friends and phone calls



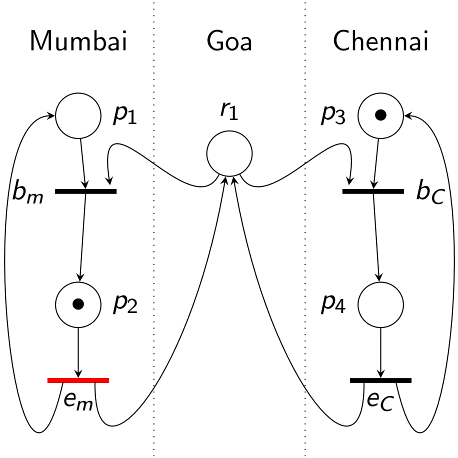
Example: Three friends and phone calls



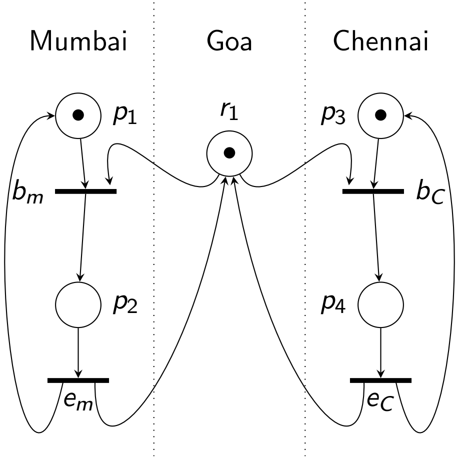
Example: Three friends and phone calls



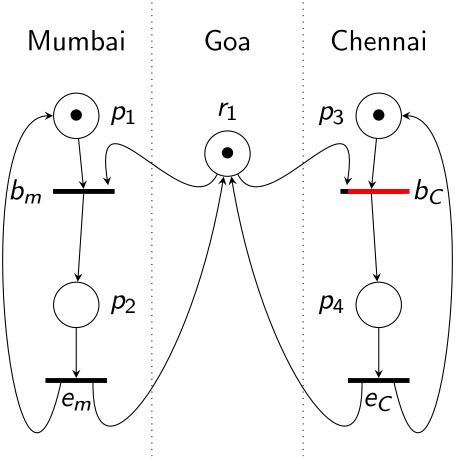
Example: Three friends and phone calls



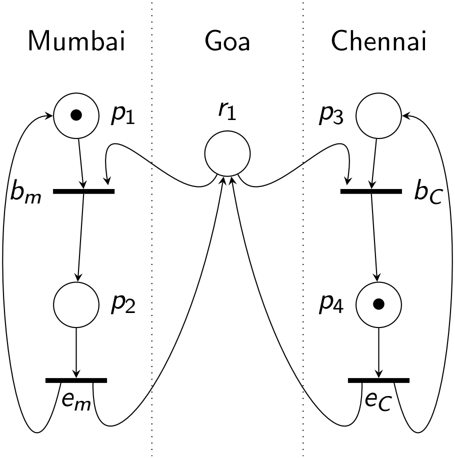
Example: Three friends and phone calls



Example: Three friends and phone calls

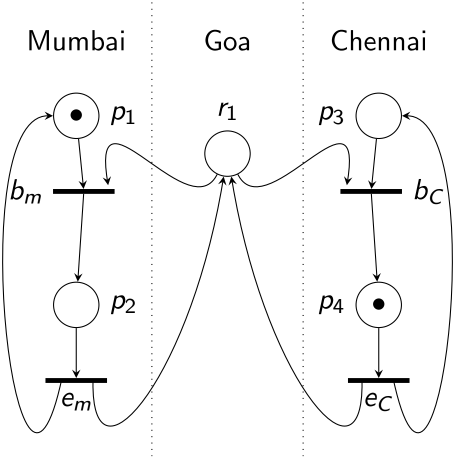


Example: Three friends and phone calls

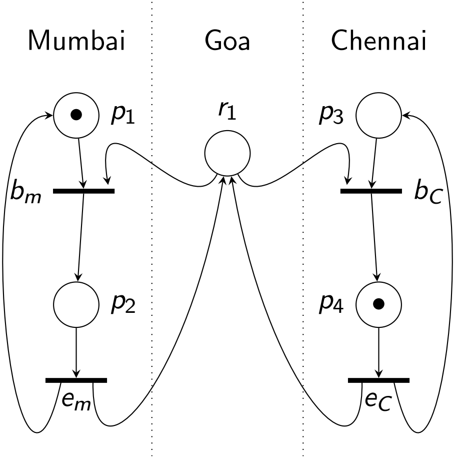


Example: Three friends and phone calls : Petri net

Places P



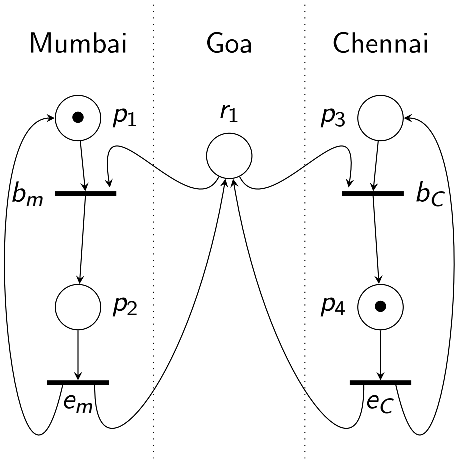
Example: Three friends and phone calls : Petri net



Places P

Transitions T

Example: Three friends and phone calls : Petri net

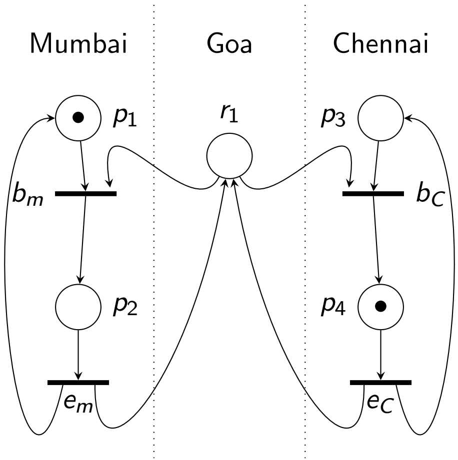


Places P

Transitions T

Flow relation F

Example: Three friends and phone calls : Petri net



Places P

Transitions T

Flow relation F

Marking

Example: Three friends and phone calls : Petri net

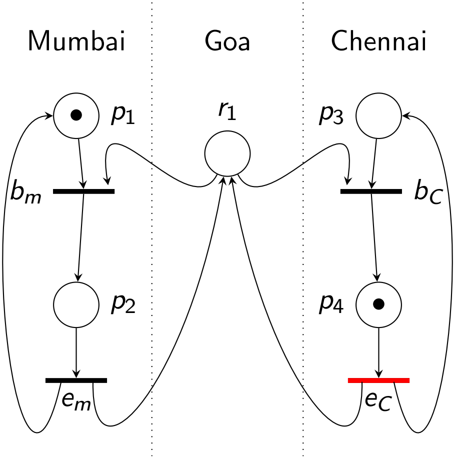
Places P

Transitions T

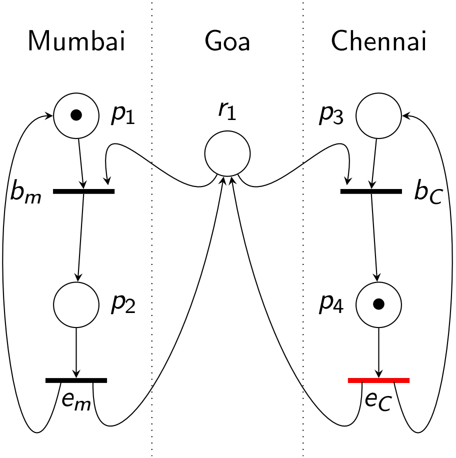
Flow relation F

Marking

Enabled transition



Example: Three friends and phone calls : Petri net



Places P

Transitions T

Flow relation F

Marking

Enabled transition

Firing of transition

Example: Three friends and phone calls : Petri net

Places P

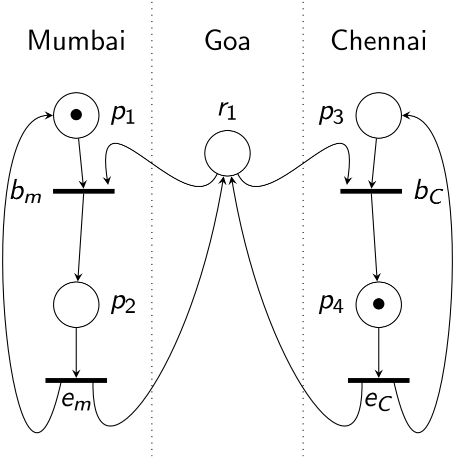
Transitions T

Flow relation F

Marking

Enabled transition

Firing of transition



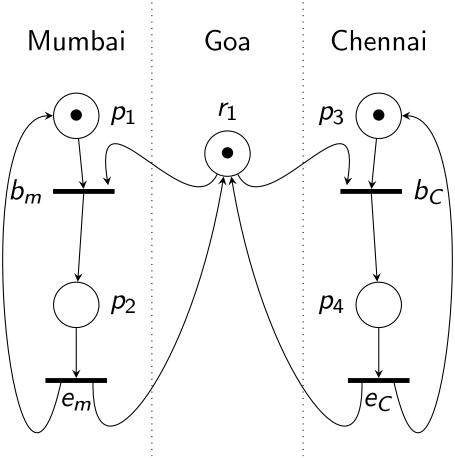
Example: Three friends and phone calls : Petri net

Places P

Transitions T

Flow relation F

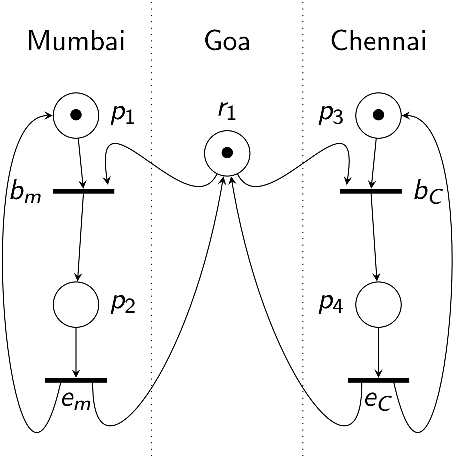
Marking



Enabled transition

Firing of transition

Example: Three friends and phone calls : Petri net



Places P

Transitions T

Flow relation F

Marking

Enabled transition

Firing of transition

Firing Sequence

Example: Three friends and phone calls : Petri net

Places P

Transitions T

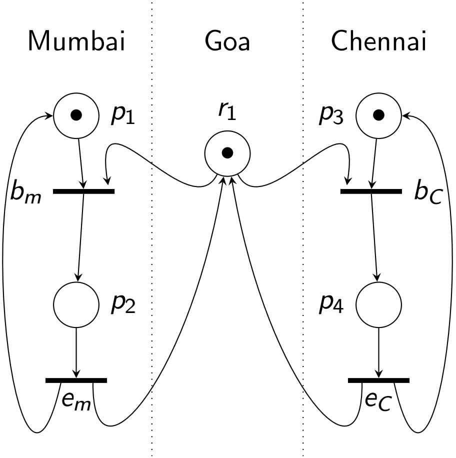
Flow relation F

Marking

Enabled transition

Firing of transition

Firing Sequence



$\text{Language}(N, \{p_1, r_1, p_3\}) = \{b_m e_m, b_C e_C\}^*$

Example: Three friends and phone calls : Petri net

Places P

Transitions T

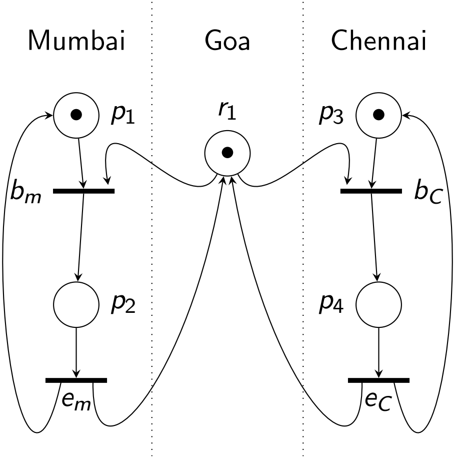
Flow relation F

Marking

Enabled transition

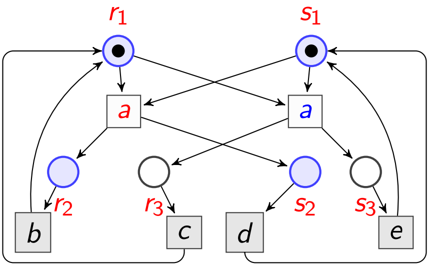
Firing of transition

Firing Sequence



$$\text{Language}(N, \{p_1, r_1, p_3\}) = \{b_m e_m, b_C e_C\}^*$$

Net and its language : Another example



- 1-bounded and labelled with $\Sigma = \{a, b, c, d, e, f\}$
- Initial marking $\{r_1, s_1\}$
- Final markings $\mathcal{G} = \{\{r_1, s_1\}, \{r_2, s_2\}\}$ [Pet76].
- Language $(a(bd + db) + a(ce + ec))^*(\varepsilon + a)$ [Jan87].
- Can we write this in terms of its components?

Free choice net

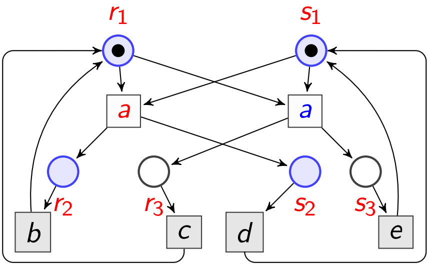


Figure: Free Choice Net

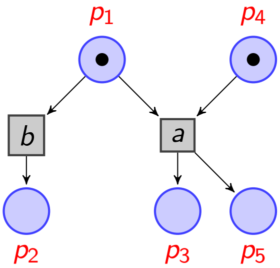


Figure: Non free choice net

If two transitions share a pre-place then their sets of pre-places are equal.

Free choice net and its S-decomposition

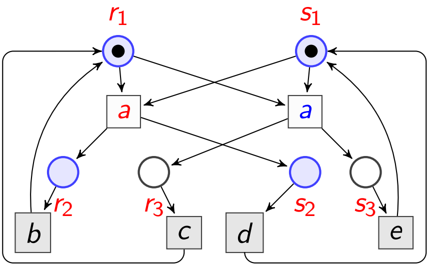


Figure: Free Choice Net

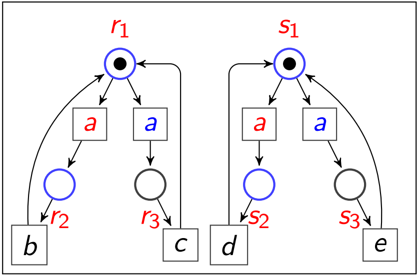


Figure: S-cover of the net

Free choice net and its S-decomposition

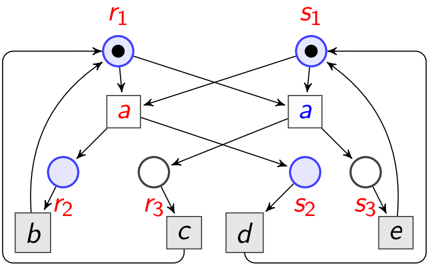


Figure: Free Choice Net

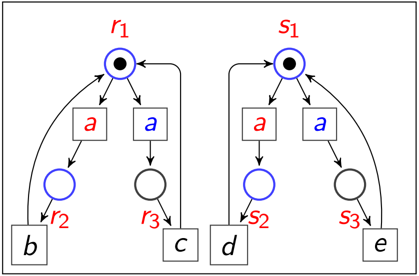


Figure: S-cover of the net

- $\Sigma = (\Sigma_1 = \{a, b, c\}, \Sigma_2 = \{a, d, e\})$
- Distributed alphabet over locations $Loc = \{1, 2\}$.
- S-cover respects distribution.

Automata over distributed alphabet–01

- A_1 with $G_1 = \{r_1, r_2\}$ over Σ_1.
- A_2 with $G_2 = \{s_1, s_2\}$ over Σ_2.
- $A = (A_1, A_2)$ over $\Sigma = (\Sigma_1, \Sigma_2)$ with
  $\Rightarrow$: global transitions, and
  G : Final states.
- possible product transitions $\mathcal{T} = \Pi_{i \in loc(a)} \rightarrow_a^i$
- possible product final states $\mathcal{G} = \Pi_{i \in Loc} G_i$

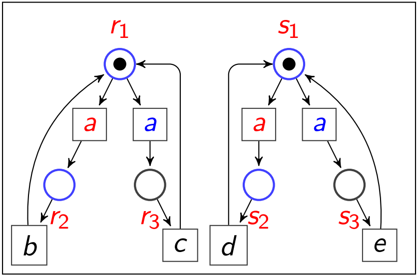
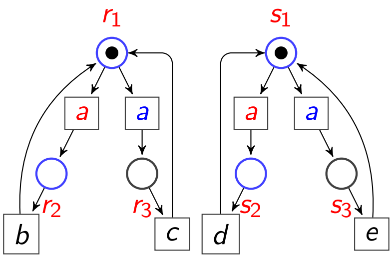


Figure: $A = (A_1, A_2)$ over Σ

Automata over distributed alphabet–02

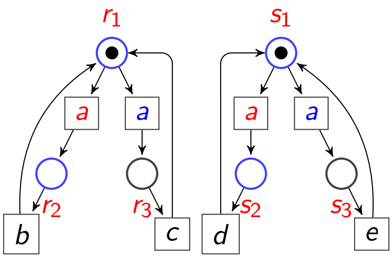
1. Direct products (DP): $\Rightarrow = \mathcal{T}, \quad G = \mathcal{G}$
2. Synchronous products : $\Rightarrow = \mathcal{T}, \quad G \subseteq \mathcal{G}$
3. Zielonka automata with PAC: $\Rightarrow \subseteq \mathcal{T}, \quad G = \mathcal{G}$
4. Zielonka automata: $\Rightarrow \subseteq \mathcal{T}, \quad G \subseteq \mathcal{G}$

Direct product



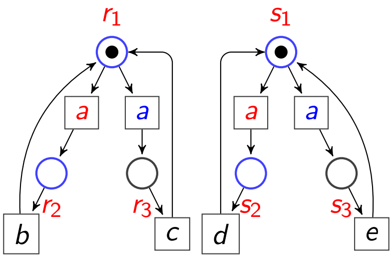
1. $\mathcal{G} = \{(r_1, s_1), (r_1, s_2), (r_2, s_1), (r_2, s_2)\}$.
2. $\mathcal{T} = \{(a, a), (a, a), (a, a), (a, a)\}$.
3. Choose: $G = \mathcal{G}$.
4. And $\Rightarrow = \mathcal{T}$.
5. DP : $r^*[\varepsilon + a + a(b + c) + a(d + e)]$ where $r = (a(bd + db) + a(be + eb) + a(cd + dc) + a(ce + ec))$.

Synchronous product



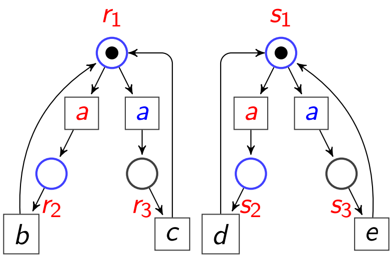
1. $\mathcal{G} = \{(r_1, s_1), (r_1, s_2), (r_2, s_1), (r_2, s_2)\}$.
2. $\mathcal{T} = \{(a, a), (a, a), (a, a), (a, a)\}$.
3. Choose: $G = \{(r_1, s_1), (r_2, s_2)\} \subseteq \mathcal{G}$.
4. And $\Rightarrow = \mathcal{T}$.
5. SP : $r^*[\varepsilon + a]$ where
   $r = (a(bd + db) + a(be + eb) + a(cd + dc) + a(ce + ec))$.

Zielonka Automata (ZA)



1. $\mathcal{G} = \{(r_1, s_1), (r_1, s_2), (r_2, s_1), (r_2, s_2)\}$.
2. $\mathcal{T} = \{(a, a), (a, a), (a, a), (a, a)\}$.
3. Choose: $G = \{(r_1, s_1), (r_2, s_2)\} \subseteq \mathcal{G}$.
4. And $\Rightarrow = \{(a, a), (a, a)\} \subseteq \mathcal{T}$.
5. ZA : $r^*[\varepsilon + a]$ where $r = (a(bd + db) + a(ce + ec))$.

ZA with Product Acceptance



1. $\mathcal{G} = \{(r_1, s_1), (r_1, s_2), (r_2, s_1), (r_2, s_2)\}$.
2. $\mathcal{T} = \{(a, a), (a, a), (a, a), (a, a)\}$.
3. We have chosen : $\quad G = \mathcal{G}$
4. And $\Rightarrow = \{(a, a), (a, a)\} \subseteq \mathcal{T}$.
5. ZA with PAC : $r^*[\varepsilon + a + ab + ad]$ where
   $r = (a(bd + db) + a(ce + ec))$.

Automata over distributed alphabet–03

Lemma ([Muk11])

A language is accepted by synchronous product iff it can be expressed as union of direct product languages.

Automata over distributed alphabet–03

Lemma ([Muk11])

A language is accepted by synchronous product iff it can be expressed as union of direct product languages.

Lemma ([Pha18])

A language is accepted by Zielonka automaton iff it can be expressed as union of ZA with product acceptance condition languages.

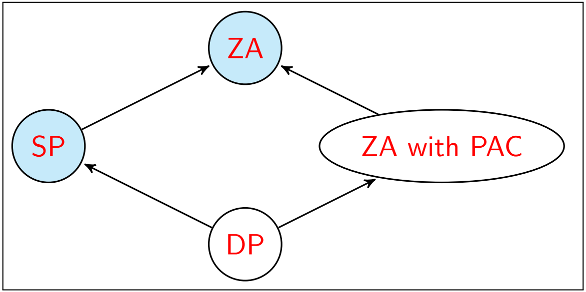
Automata over distributed alphabet–03

Lemma ([Muk11])

A language is accepted by synchronous product iff it can be expressed as union of direct product languages.

Lemma ([Pha18])

A language is accepted by Zielonka automaton iff it can be expressed as union of ZA with product acceptance condition languages.



Properties of Zielonka Automaton: Capturing free choice

[Pha16] Same source property: If any two global moves share a pre-state then their sets of pre-states are same.

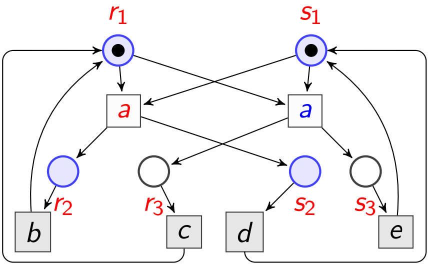


Figure: Free Choice Net

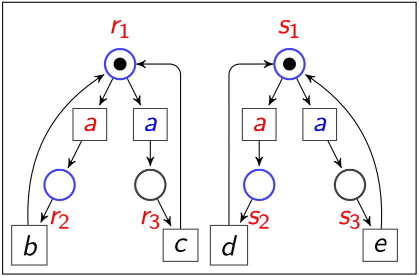


Figure: ZA with same source

Equivalence of Nets and Zielonka automaton

1. Union of all local states is set of places of net
2. Initial state of ZA maps to Initial marking
3. Final states of ZA map to Final markings of net
4. Global transitions of ZA map to transitions of net

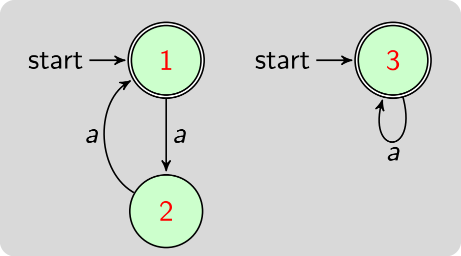
Equivalence of Nets and Zielonka automaton

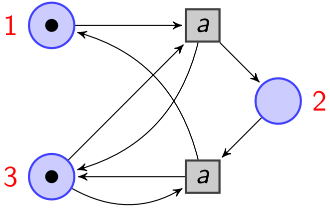
1. Union of all local states is set of places of net
2. Initial state of ZA maps to Initial marking
3. Final states of ZA map to Final markings of net
4. Global transitions of ZA map to transitions of net
5. Same Source Property of ZA maps to Free choice property of nets.

Equivalence of Nets and Zielonka automaton

1. Union of all local states is set of places of net
2. Initial state of ZA maps to Initial marking
3. Final states of ZA map to Final markings of net
4. Global transitions of ZA map to transitions of net
5. Same Source Property of ZA maps to Free choice property of nets.
6. product acceptance condition of ZA maps to product condition on final markings of net. [Pha16]
7. Subset acceptance condition of ZA maps to final markings of nets need not satisfy product condition.

Direct products to Free choice?





Properties of Direct products -01

Definition (matching)

For global $a \in \Sigma$, an a-matching is a subset of tuples $\Pi_{i \in loc(a)} P_i$, such that if a place p appears in a tuple, it does not appear in another tuple. And
each pre-place with outgoing local a-moves appear in a tuple of matching.

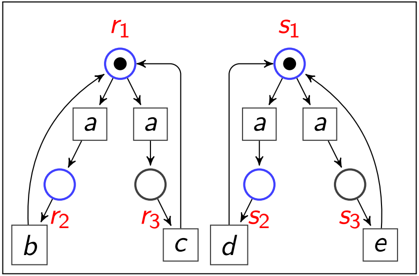
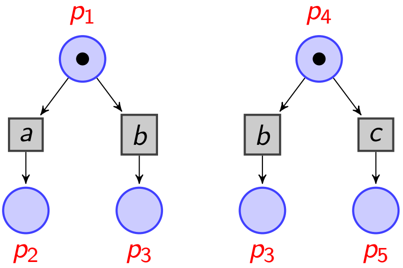


Figure: $A = (A_1, A_2)$ over Σ

Properties of Direct products -02

Definition (conflict-equivalent matching, [PL14])

We call a matching **conflict-equivalent**, if whenever p, p' are related by the matching, they are conflict-equivalent.



$s_1 = a + b$ $\qquad$ $s_1 = b + c$

Properties of Direct products -03

Definition (consistency with matching)

- A product state R is in an a-matching if its projection $R{\downarrow}loc(a)$ is in the matching.
- A run of A is said to be **consistent with a matching of labels** if for all global actions a and every prefix of the run $R^0 \stackrel{v}{\Rightarrow} R \stackrel{a}{\Rightarrow} Q$, the pre-places $R{\downarrow}loc(a)$ are in the matching.

Properties of Direct products -03

Definition (consistency with matching)

- A product state R is in an a-matching if its projection $R{\downarrow}loc(a)$ is in the matching.
- A run of A is said to be **consistent with a matching of labels** if for all global actions a and every prefix of the run $R^0 \overset{v}{\Rightarrow} R \overset{a}{\Rightarrow} Q$, the pre-places $R{\downarrow}loc(a)$ are in the matching.

If $R \overset{a}{\Rightarrow} Q$

then

pre-places of a-labelled product move should be in matching relation.

Direct product with matching : Example

- $matching(a) = \{(r_1, s_1)\}$
- conflict-equivalent matching
- consistency with matching
- possible product transitions $\mathcal{T} = \Pi_{i \in loc(a)} \rightarrow^i_a$
- possible product final states $\mathcal{G} = \Pi_{i \in Loc} G_i$

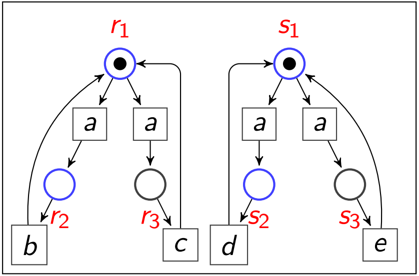


Figure: $A = (A_1, A_2)$ over Σ

Direct product representable free choice nets–01

Definition (Distributed choice [PL14])

For each cluster the set of post-configurations of a-labelled transitions is same as the product of sets formed by projecting post-states of a-labelled transitions over locations of a.

$\{(r_2, s_2), (r_2, s_3), (r_3, s_2), (r_3, s_3)\} = \{r_2, r_3\} \times \{s_2, s_3\}$

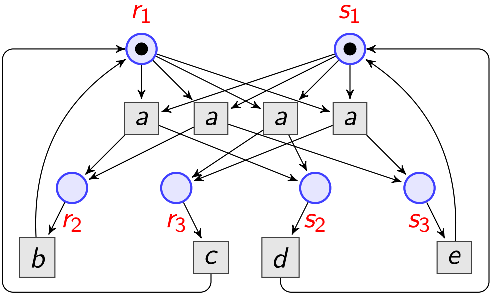
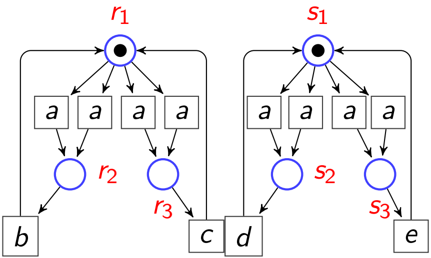


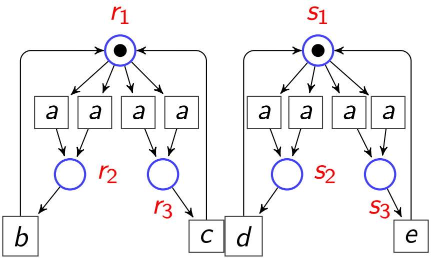
Figure: Free Choice Net with distributed choice

Direct product representable free choice nets–02



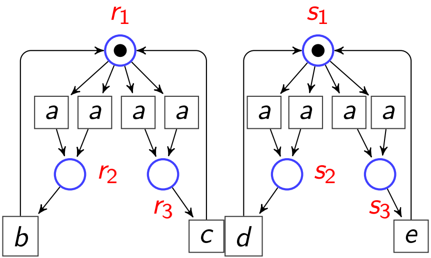
1. Final states of DP map to Final markings of net
2. Global transitions of DP to transitions of net

Direct product representable free choice nets–02



1. Final states of DP map to Final markings of net
2. Global transitions of DP to transitions of net
3. Conflict-equivalent matchings maps to Free choice property of nets.

Direct product representable free choice nets–02



1. Final states of DP map to Final markings of net
2. Global transitions of DP to transitions of net
3. Conflict-equivalent matchings maps to Free choice property of nets.
4. product acceptance condition of DP maps to product condition on final markings of net. [Pha16].
5. Subset acceptance condition of DP maps to final markings of nets which may not satisfy product condition [Pha18].

Relating matchings of DP and same source of ZA–01

Relating matchings of DP and same source of ZA–01

Definition (Product moves property of ZA [Pha18])

For all a-labelled global moves of ZA having same set of pre-states, their set of post-configurations is same as the product of sets formed by projecting post-states of a-labelled transitions over locations of a.

Relating matchings of DP and same source of ZA–01

Definition (Product moves property of ZA [Pha18])

For all a-labelled global moves of ZA having same set of pre-states, their set of post-configurations is same as the product of sets formed by projecting post-states of a-labelled transitions over locations of a.

Theorem (same source to matchings [Pha18])

Let Σ be a distributed alphabet and A be Zielonka automata with product moves property. Then we can construct a Direct product B with matchings, linear the size of A such that,

1. *if A has same source then B has conflict-equivalent matchings,*
2. *in addition, if A is live then B is consistent with matchings, and $Lang(A) = Lang(B)$.*

Relating matchings of DP and same source of ZA–02

The reverse direction....

Theorem (matchings to same source and Product moves [Pha18])

Let Σ be a distributed alphabet.
Let B be a Direct product with conflict equivalent and consistent matchings.
Then for the language of B we can construct a Zielonka Automata A, having same source and product moves property.
The constructed Zielonka Automata A is exponential in the size of system B having matching of labels.

Relating matchings of DP and same source of ZA–02

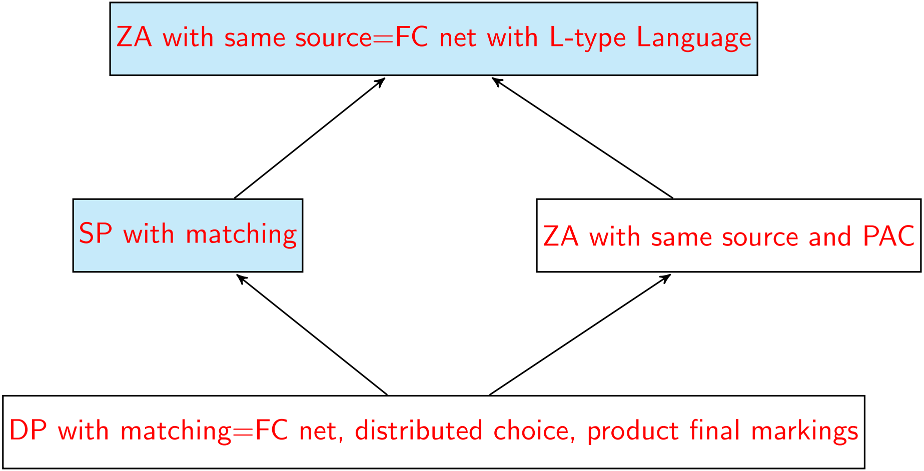
The reverse direction....

Theorem (matchings to same source and Product moves [Pha18])

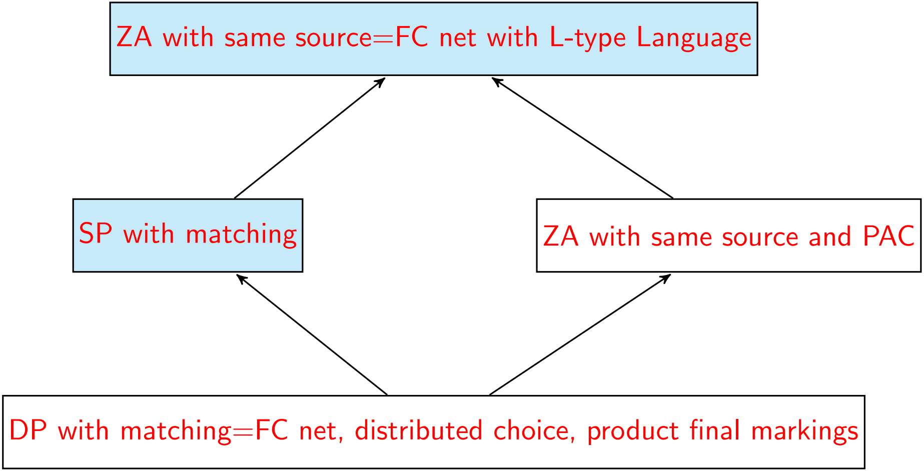
Let Σ be a distributed alphabet.
Let B be a Direct product with conflict equivalent and consistent matchings.
Then for the language of B we can construct a Zielonka Automata A, having same source and product moves property.
The constructed Zielonka Automata A is exponential in the size of system B having matching of labels.

Now, two characterizations for Labelled Free choice nets with distributed choice.

Nets and automata so far



Nets and automata so far



In addition, ZA with same source has product moves property then it is equivalent to SP with matching.

Recap

1. Hierarchy of Free choice nets labelled over distributed alphabet
2. Two characterizations of FC-nets with distributed choice
3. syntactic characterizations of all four classes (another talk)

Future work(s)

1. Free choice Asynchronous (Zielonka) automata, does Zielonka's proof gets simplified?
2. Logical characterization of free choice nets labelled over distributed alphabets.
3. Connections of product systems with matchings with negotiations.

Thank you.

References

Matthias Jantzen.
Language theory of petri nets.
In *ACPN*, volume 254 of *LNCS*, 1987.

Madhavan Mukund.
Automata on distributed alphabets.
In Deepak D'Souza and Priti Shankar, editors, *Modern Applications of Automata Theory*. World Scientific, 2011.

James L. Petersen.
Computation sequence sets.
Journal of Computing and Systems Science, 13(1):1–24, 1976.

Ramchandra Phawade.
Kleene theorems for labelled free choice nets without distributed choice.
In Lawrence Cabac, Lars Michael Kristensen, and Heiko Rölke, editors, *Proc. PNSE*, volume 1591 of *CEUR Workshop Proceedings*, pages 132–152. CEUR-WS.org, 2016.

Ramchandra Phawade.
Kleene theorems for labelled free choice nets labelled with distributed alphabets.
In Ekkart Kindler Daniel Moldt and Heiko Rölke, editors, *Proc. PNSE*, volume 2138 of *CEUR Workshop Proceedings*, pages 77–98. CEUR-WS.org, 2018.

Ramchandra Phawade and Kamal Lodaya.
Kleene theorems for labelled free choice nets.
In Daniel Moldt and Heiko Rölke, editors, *Proc. PNSE*, volume 1160 of *CEUR Workshop Proceedings*, pages 75–89. CEUR-WS.org, 2014.

Antimirov Derivatives

- $\mathrm{Der}_a(0) = \emptyset$
- $\mathrm{Der}_a(1) = \emptyset$
- $\mathrm{Der}_a(b) = \{\varepsilon\}$ if $b = a$ $\emptyset$ otherwise
- $\mathrm{Der}_a(s_1 + s_2) = \mathrm{Der}_a(s_1) \cup \mathrm{Der}_a(s_2)$
- $\mathrm{Der}_a(s_1^*) = \mathrm{Der}_a(s_1) \cdot s_1^*$
- $\mathrm{Der}_a(s_1 \cdot s_2) = \left\{ \begin{array}{ll} \mathrm{Der}_a(s_1) \cdot s_2 \cup \mathrm{Der}_a(s_2) & \text{if } \varepsilon \in Lang(s_1) \\ \mathrm{Der}_a(s_1) \cdot s_2 & \text{otherwise} \end{array} \right\}$