Concurrency Theory, 4 Sep 2019

Traces

$D = \overline{I}$ reflexive, symmetric dependence relation

(Σ, I), I irreflexive & symmetric independence relation

$u\,a\,b\,v \sim_1 u\,b\,a\,v$ if $a\,I\,b$

$\tilde{\sim}_I$: reflexive, transitive closure $\sim_1$

$[w] = \{w' \mid w' \tilde{\sim} w\}$ is the trace containing w

Labelled partial order representation of a trace

$(E, \leq, \lambda)$ $\quad \lambda : E \to \Sigma$

$w = a_1 a_2 \ldots a_n \quad E = \{(a_i, i) \mid 1 \leq i \leq n\}$

$(a,i) < (b,j)$ if $a D b \wedge i < j$

$\Downarrow$

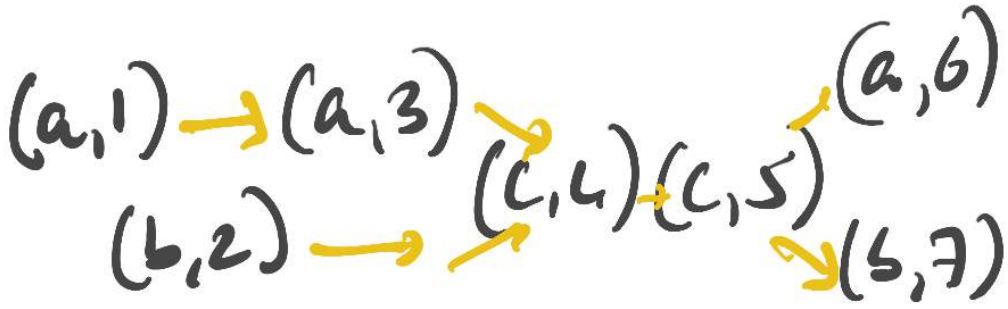
$\leq$ is reflexive, transitive closure of $<$

$\lambda((a,i)) = a$

$\Sigma = \{a, b, c\}$

$I = \{(a,b), (b,a)\}$

abaccab



Given $T_w = (E_w, \leq_w, \lambda_w)$ constructed from w

$[w] = lin(T_w)$ Valid linearizations

Consider $w_1 \leq w_2$ if $\exists u \quad w_1 u = w_2$

Word Prefix

Trace Prefix

$[u] \sqsubseteq [v]$

$\exists u' \in [u] \quad \exists v' \in [v]$

$u \leq v$

Consider (Σ, I) as before

$[b] \sqsubseteq [ab]$

$b \not\leq ab$

In P.O. terms

$[u] \sqsubseteq [v]$ if T_u is a p.o. prefix of T_v

$(E_u, \leq_u, \lambda_u)$ $(E_v, \leq_v, \lambda_v)$

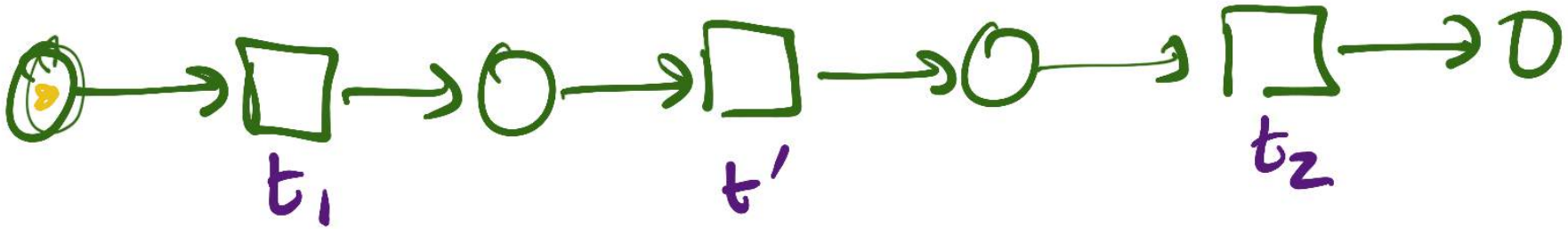
$h: E_u \to E_v$ that preserves

immediate ordering ---

Better definition of language for a concurrent system

ENS : $\Sigma = T$

$$I = \{ (t_1, t_2) \mid {}^\bullet t_1^\bullet \cap {}^\bullet t_2^\bullet = \phi \}$$



$${}^\bullet t_1^\bullet \cap {}^\bullet t_2^\bullet = \phi$$

Digression

$L \subseteq \Sigma^*$ is a word language

$Tr(\Sigma, I) = \Sigma^*/\approx$

Trace language is a subset of $Tr(\Sigma, I)$, so a word language closed w.r.t $\approx$

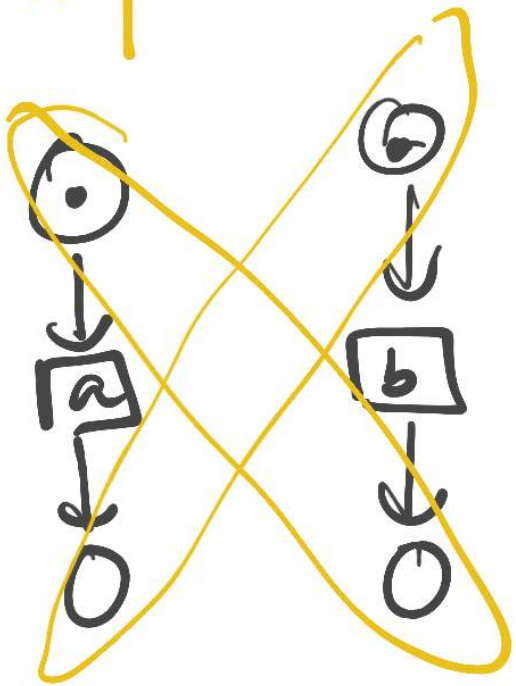
Regular trace language? — Trace language
— Regular as a word language

ENS $\rightarrow (T, I_N)$

Σ I

Behaviour is a regular trace language (prefix closed)

Final Markings?



Behaviour allows

a, b

but not ab / ba

Problematic

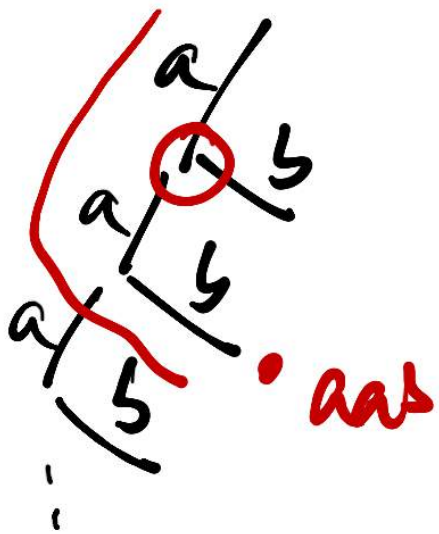
Converse

Find a good machine model to recognize all regular trace languages.

Finite state automaton

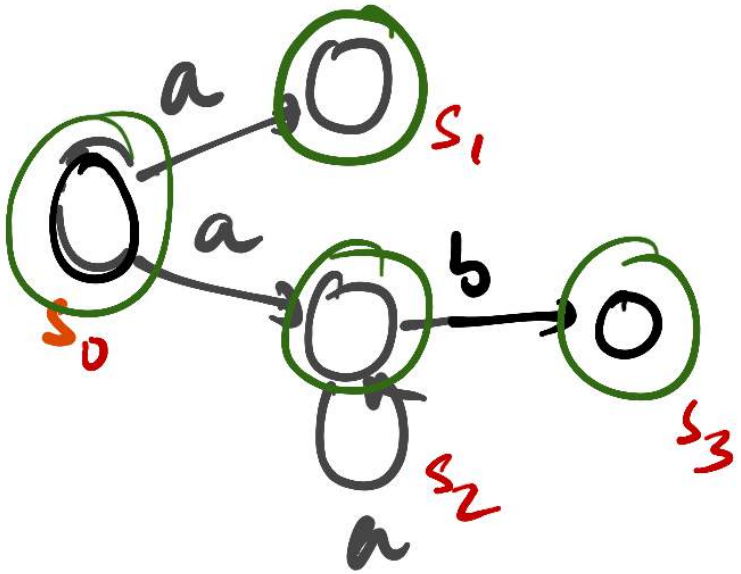
Prefix closed language → Tree

$L = \{a, ab, aab, \cdots\}$

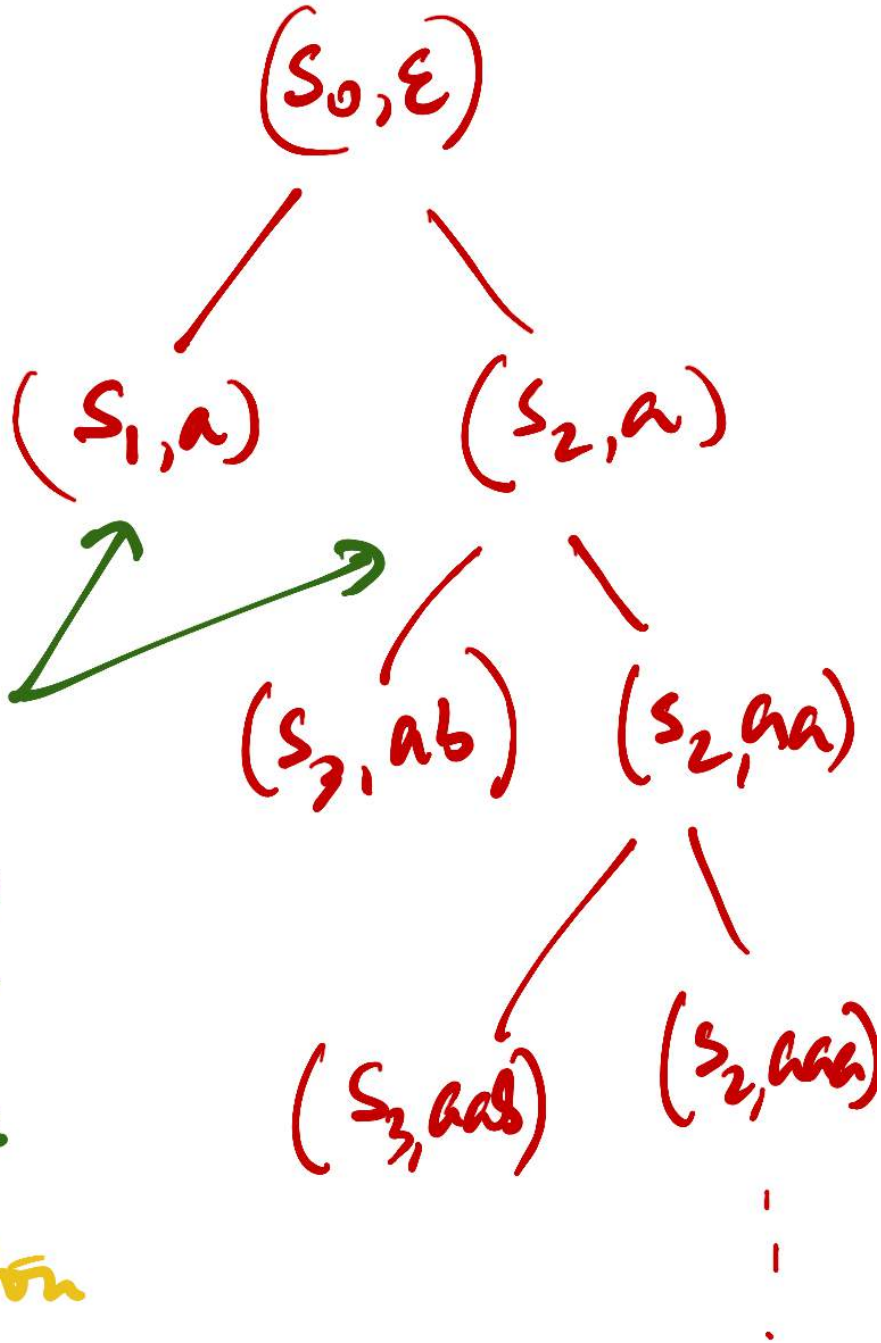


$a \cup a^+b$

What about nondeterminism



$a \cup a^+b$



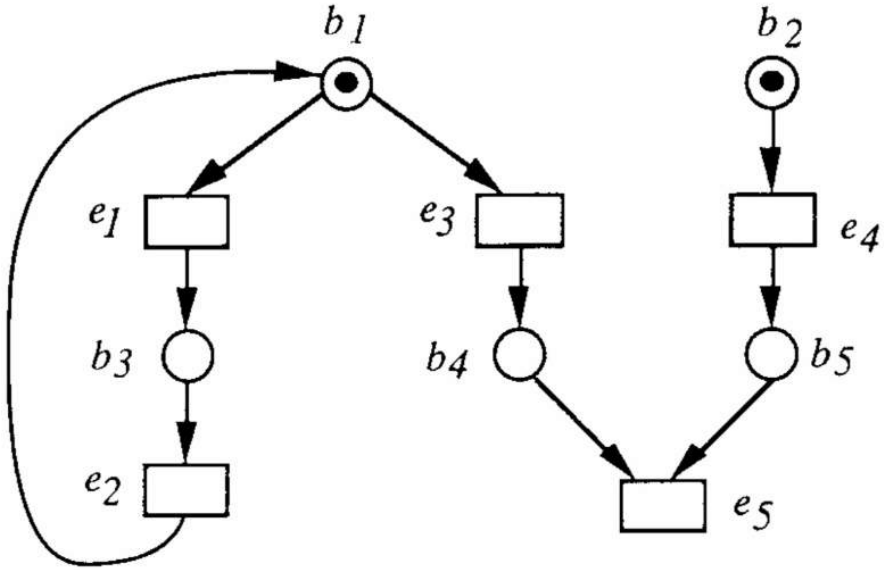
same word, different contexts

Unfolding of the automaton

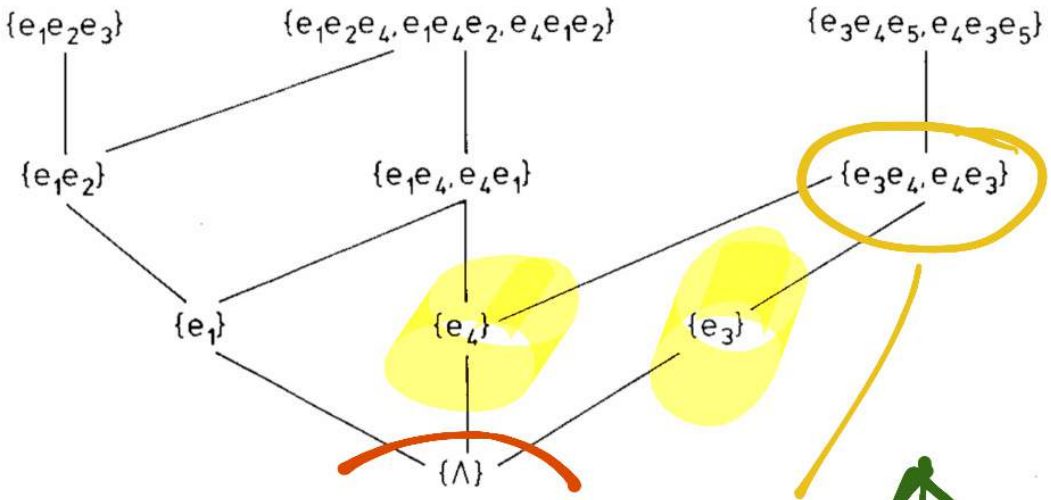
How does one unfold a net?

Tree associated with the prefix closed language

$\Rightarrow$ Traces ordered by prefix



"Tree" $\uparrow$ traces



$[e_3, e_4] = e_3e_4, e_4e_3$

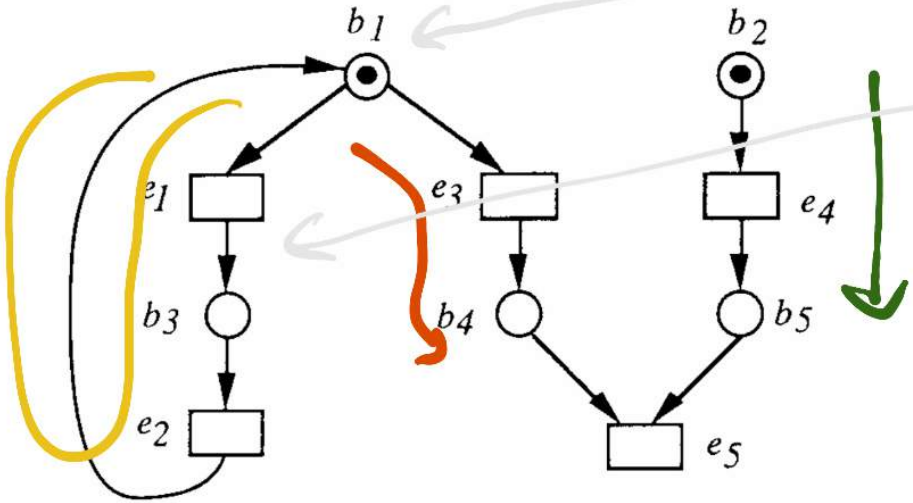
Word ≡ Computation of an automaton

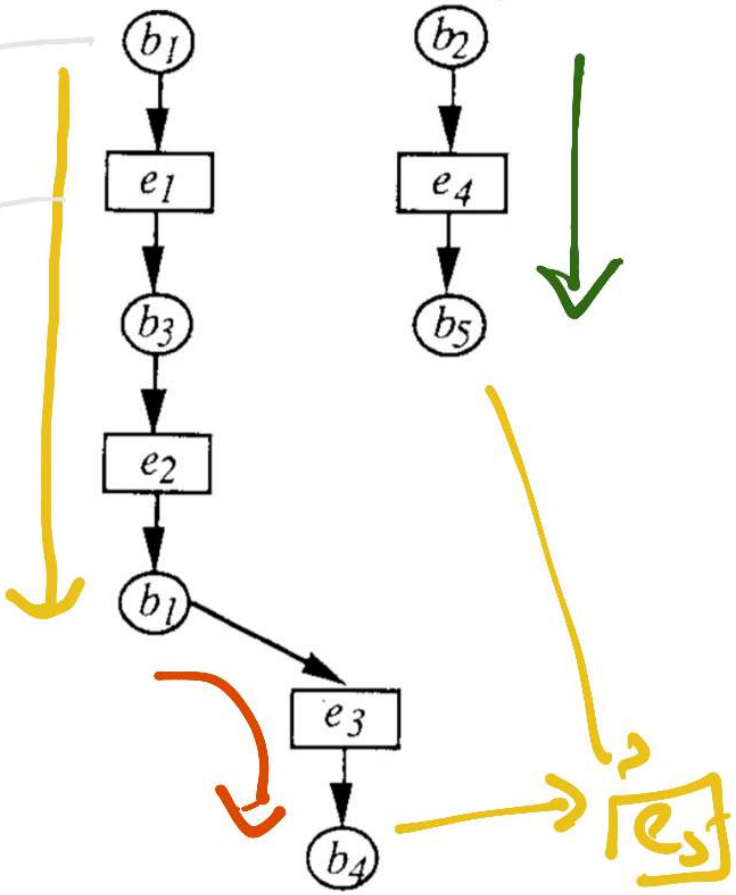
Trace = " " " net

Word ≡ Sequential automaton

Net representation of a trace?

Non sequential Process





e1 e4
↓
e2
↓
e5

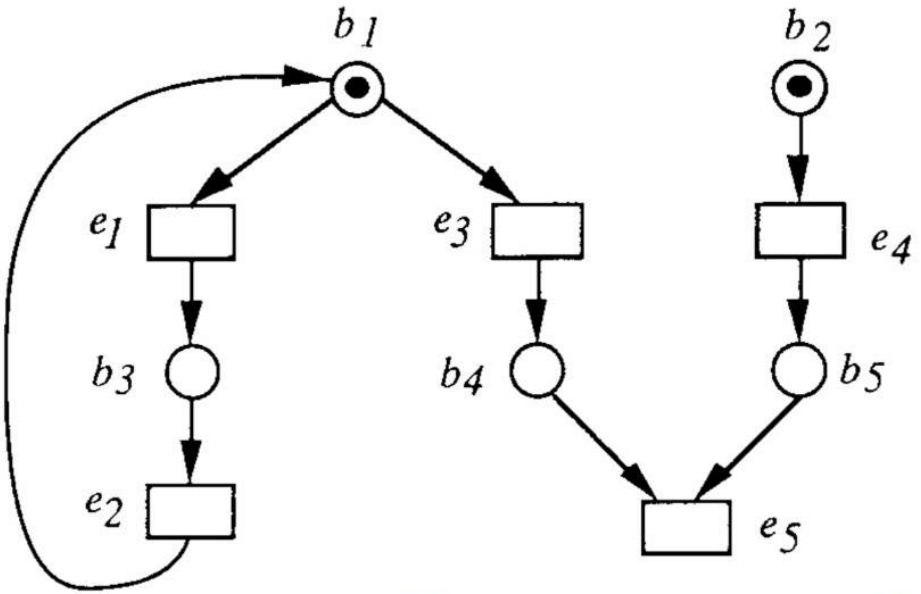
Building a process

Record each copy of a place / transition with its context (past)

ENS:

$((P, T, F), M_{in})$

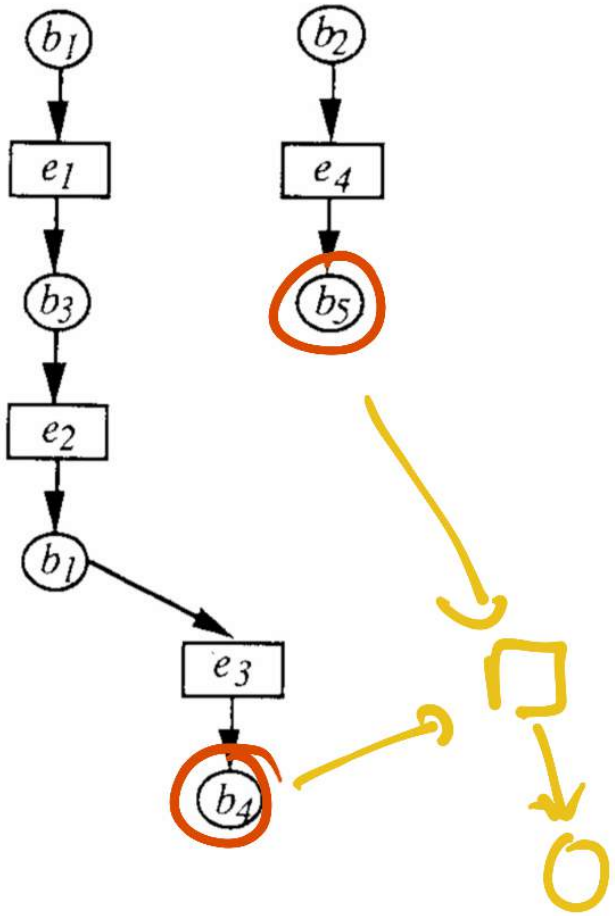
$((B, E, F), c_{in})$



For each firing sequence S, construct (N_S, M_S)

What is N_ε?

Should have (ϕ, ϕ, ϕ)

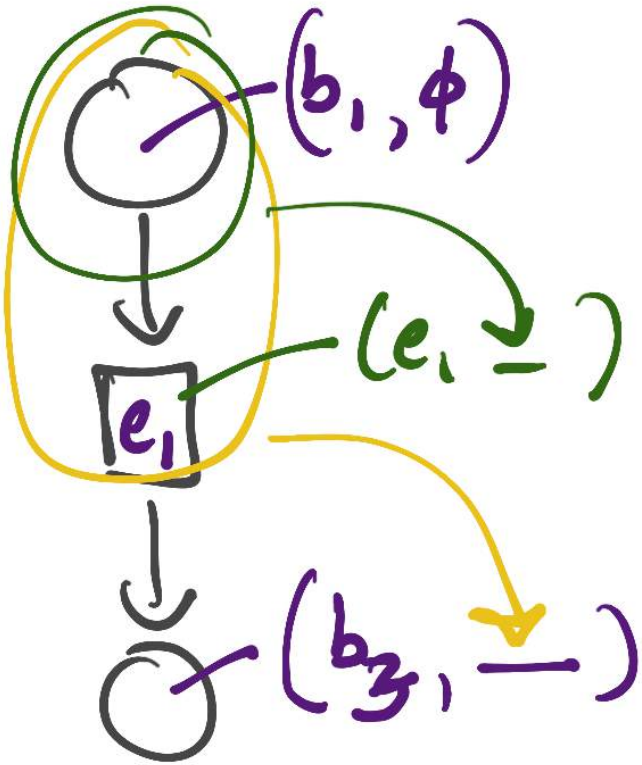


$N_{e_1e_2e_3e_4} = N_{e_1e_2e_4e_3}$ --

M ___ = last b_4, b_5

$N_\varepsilon = (\phi, \phi, \phi)$

N_{e_1}



$N_\varepsilon = (\phi, \phi, \phi)$ $M_\varepsilon = \phi$ $C_\varepsilon =$ Initially marked places

$\{(b_1, \phi), (b_2, \phi)\}$

$\lambda_\varepsilon = \phi$

Inductively

$N_S = (P_S, T_S, F_S)$, λ_S, M_S, C_S

Add $(t, -)$ s.t. $t \in T$

$\forall p \in {}^\bullet t \quad \exists (p, x) \in C_S$

New $\hat{t} = (t, Y)$, $Y = \{ (p, x) \mid p \in {}^\bullet t, (p, x) \in C_S$

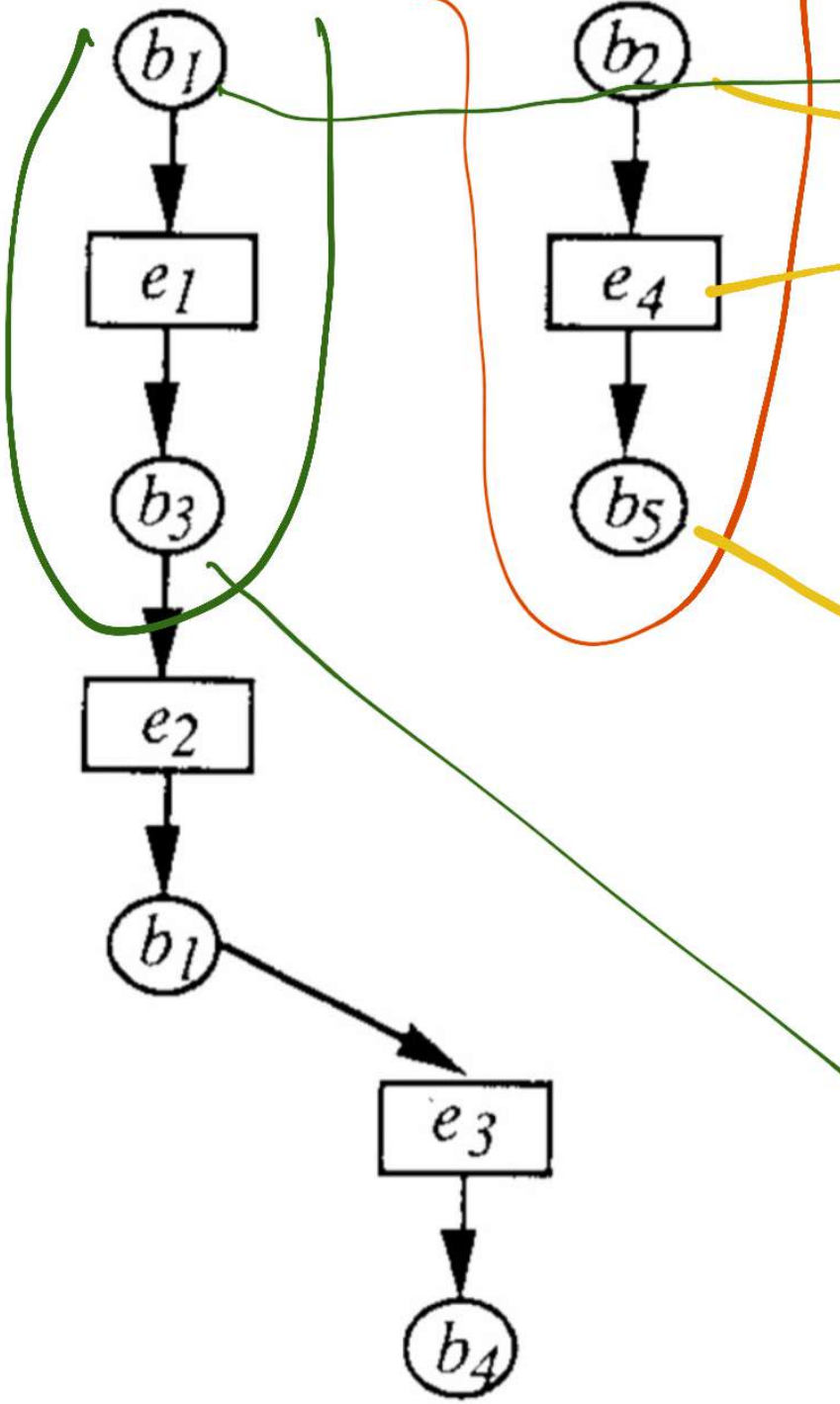
$\forall p \in t^\bullet$, add $(p, (t, Y))$

$$N_{gt} = (P_g \cup I \cup O,\ E_g \cup \{(t,\gamma)\},\ F_g \cup \dots)$$

Inputs to t

$$O = \{(p,(t,\gamma)) \mid p \in t^-\}$$

Needed to go from N_E to N_t



N_E $C_E = (b_1, \phi), (b_2, \phi)$

$(e_4, (b_2, \phi))$

$I = (b_2, \phi)$

$O = (b_5, (e_4, (b_3, \phi)))$

$I = (b_1, \phi)$

$(e_1, (b_1, \phi))$

$O = (b_3, (e_1, (b_1, \phi)))$