Sisteme cu membrane (1)

Prof Emeritus Marian Gheorghe

Department of Computer Science
University of Bradford
United Kingdom

Sumar

- Modele si contextul modelarii
- Modele cu rescriere
- Sistemele cu membrane (P sisteme), model cu rescriere
- Tipuri de sisteme cu membrane
- Modelarea cu P sisteme. Exemple.

Modelare & Inginerie Software

- **Ingineria Software**:
- *Termen* introdus la Conferinta NATO pe aceasta tema (construirea sistemelor de operare IBM), 1968 (B Randell, P Naur)
- *Definitie*: “SE is an **engineering discipline** that is concerned with **all aspects of software production** from the **early stages of system specification to maintaining the system**” - Ian Sommerville

- **Ingineria** se ocupa de proiectarea, dezvoltarea si intretinerea unor sisteme complexe (mecanica, civila, electronica…); folosind metode si instrumente specifice; construind tehnologii adecvate
- **IS** cuprinde atat aspecte tehnice privind dezvoltarea software (sciere de cod, testare, instalare), aspecte privind planificarea activitatilor, comunicarea, definirea unui proiect (”software engineering is software too” !?)
- **Componente ale proiectelor de IS**: “requirements, design, coding, testing and maintenance”

- Unde este modelarea in IS?

Modelare & Inginerie Software

- … In majoritatea aspectelor **IS !**
- *Modele ale proceselor IS (componente):* cascada (waterfall) – specific discilplinelor ingineresti; iterative; agile
- *Modele associate componentelor:* definirea si implementarea problemelor*;* testare si verificare; mentinere software

- Tipuri de modele
- Informale: “requirements & design” – limbaj natural structurat; notatii ad-hoc (mind maps)
- Grafice (diagramatice): UML
- Formale (matematice)

Modele Formale

- Utilizeaza o definitie si notatie formala (matematica)
- Semantica precisa (de cele mai multe ori formala)
- Arie de utilizare si aplicabilitate specifica: sisteme dinamice (cu stari), concurente si distribuite

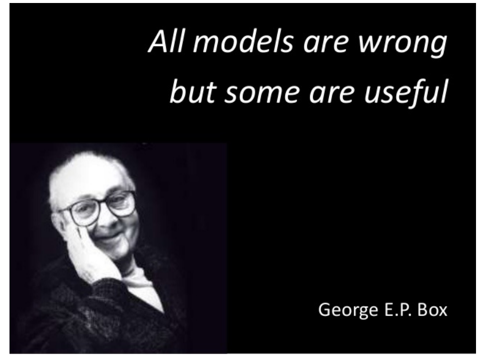
- Cele mai utilizate/cunoscute:
- Z
- Automatele finite si tipuri de gramatici independente de context
- Retele Petri
- Formalisme algebrice si logice
- …

- Scopul utilizarii MF: definirea precisa a problemei (+ghidarea implementarii); verificare/analiza formala; testare

- Exista limitari ale MF?

Limitarile Modelelor Formale

- G E P Box (1919 – 2013)
- British statistician



!!

- Necesitatea de a invata un limbaj formal
- Dificultatea de a extinde un model formal
- Dificultatea/Imposibilitatea de a captura elemente din afara domeniului considerat initial

- … Totusi sunt utile!!

Modele cu Rescriere

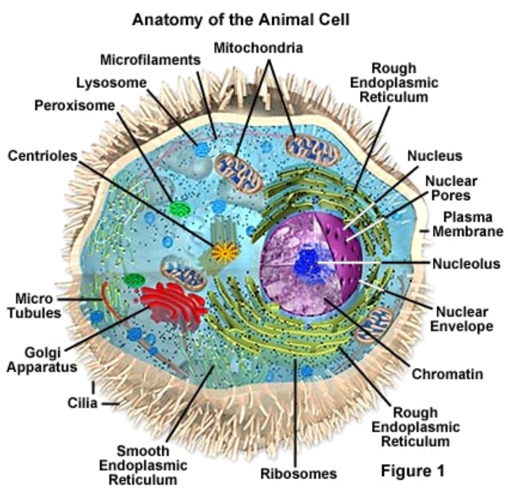
- Gramaticile generative (formale) – ierarhia Chomsky, sistemele Lindenmayer

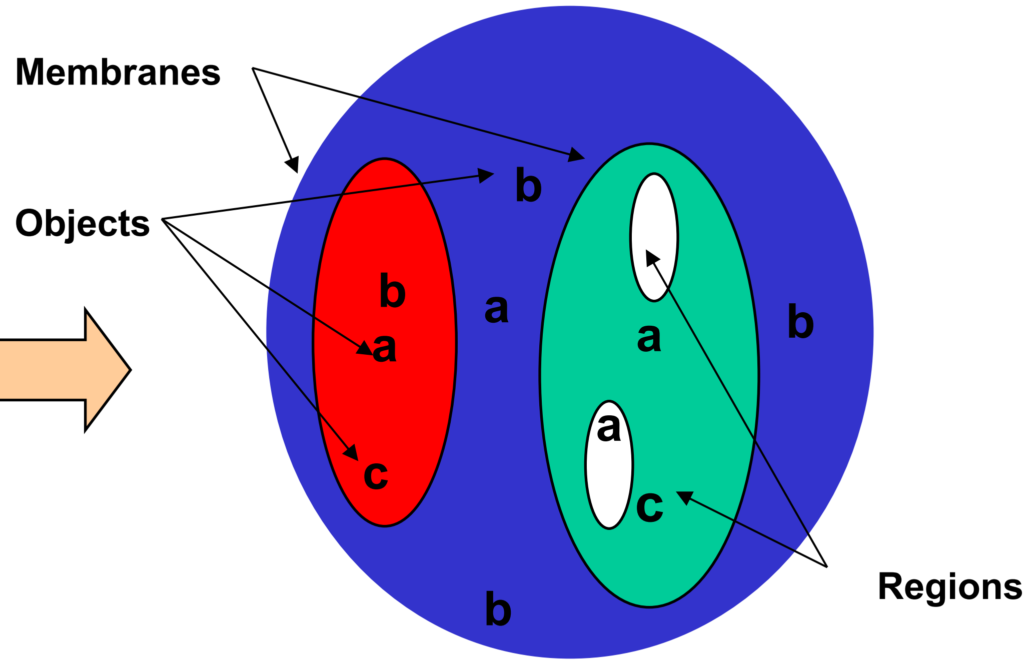
- Sisteme cu membrane (Gh Paun, Computing with membranes, J Comput Syst Sci 61,1, 108-143, 2000).

- …

- Natural Computing (L Kari, G Rozenberg, The many facets of natural computing, CACM, 51, 10, pp 72-83, 2008)

De la Celule la Sisteme cu Membrane





All cells are enclosed by membranes; the cell membrane acts as the defining principle of what constitutes a cell and the rest of the world. Cells need to be able to transport proteins, DNA, and ions across the membrane.

Science, 310, 1451, 2005

Sisteme cu Membrane

Un model de calcul inspirat (bazat) pe structura si functionarea celulei vii (**Membrane or P systems**)

Trei caracteristici esentiale ale Sistemelor cu Membrane:

- ***structura*** (ierarhica, dar nu numai) de **membrane** delimitand **regiuni/compartimente** (*arbore*)
- ***multiseturi de obiecte*** si
- ***seturi finite de reguli*** asociate compartimentelor

Un P sistem **calculeaza**, evoluand de la o configuratie (stare) la alta, aplicand regulile, in compartimente, utilizand strategia paralelismului maxim

Regulile pot rescrie sau muta obiecte dintr-un compartiment in altul, dar pot modifica structura (introducand/stergand compartimente)

Sistem cu Membrane – Definitie

$$P = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, i_0) \text{ - P system}$$

where

O – an alphabet (finite set)

μ – a membrane structure with n membranes (regions)

$w_1, \ldots, w_n$ – multisets over O; w_i – initial values

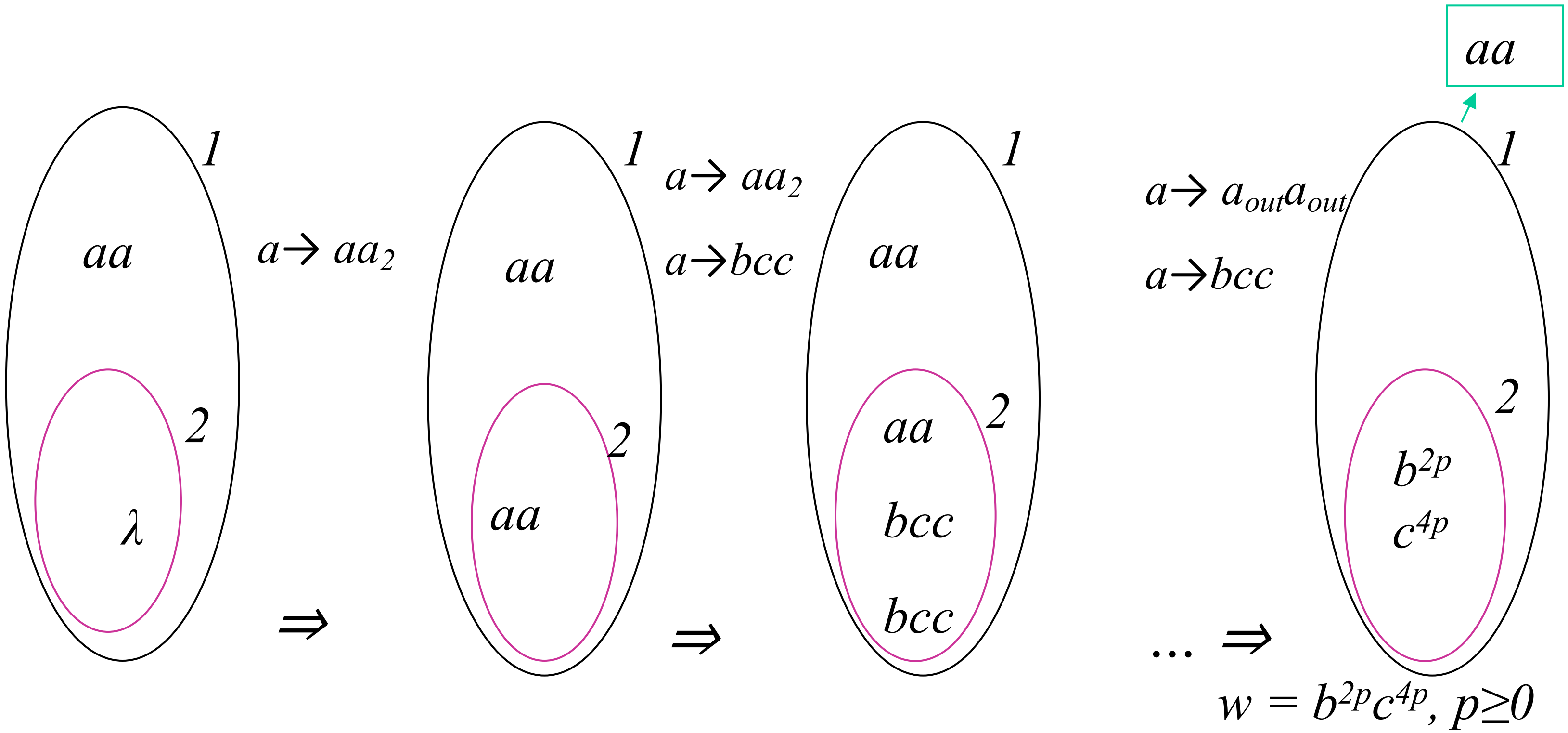
$R_1, \ldots, R_n$ – sets of rules

i_0 – the output cell (or environment)

R_i – evolution and communication rules: $v \rightarrow w$; v, w – strings over O + some indications of target regions in w

Exemplu – Doua compartimente

$P = (\{a,b,c\}, [_1[_2]_2]_1, aa, \lambda, \{a \to aa_2, aa \to a_{out}a_{out}\}, \{a \to bcc\}, 2)$



Caracteristici:
structura ierarhica (arbore); reguli de rescriere si comunicare; nedeterminism; paralelism maxim. Rezultatul: multiset (*b: 2p & c: 4p*)

Calculand cu Sistemele cu Membrane

$P = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, i_0)$ - P sistem

Rezultatul calculat de P.

Secventa de configuratii: $C_0, \ldots, C_p$ (C_i – n-tuple de obiecte); configuratia initiala este $C_0 = (w_1, \ldots, w_n)$; iar C_{i+1} se obtine din C_i aplicand regulile disponibile, folosind paralelism maxim, obiectelor din cele n compartimente.

$C_0=(aa,\lambda)$, $C_1=(aa,aa)$, $C_2=(aa,aabccbcc)$, $\ldots C_p=(\lambda,b^{2p-2}c^{4p-4})$

Rezultatul se obtine in i_0 atunci cand nu se mai poate aplica nicio regula; interpretarea: numar de simboluri ($\{6p|p\geq 0\}$).

Un P sistem calculeaza o multime de numere.

Variante de P Sisteme (1)

$$P = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, i_0) \text{ - P sistem}$$

* Obiecte:

- Simboluri distinse (numar de *b* si *c* ($\{(2p,4p) | p \geq 0\}$)); produs cartezian de numere
- Considera siruri de simboluri in loc de multiseturi (sau ambele) – regasim L sistemele

* Multimi de reguli:

- Rescrierea si comunicarea sunt separate
- Comunicarea ”bio”: (simport: $(i; x, y; j)$; antiport: $(i; x, j; y)$)
- Catalisti ($cx \to cy$); activatori, inhibitori (multiseturi de simboluri)
- Prioritati
- …

Variante de P Sisteme (2)

$$P = (O, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n, i_0) \text{ - P sistem}$$

* Structuri diferite: μ desemneaza un graf (tissue P systems); celule neuronale – spiking neural systems

* Transformari structurale: ($[x]_i$ desemneaza celula/compartimentul i continand multisetul x)
- Diviziune celulara/compartiment $[x]_i \rightarrow [y]_i[z]_i$
- Dizolvare (moarte) $[x]_i \rightarrow x$ ($x=\lambda$)
- Polarization $w[x]_i^a \rightarrow [wx]_i^b$; $[wx]_i^a \rightarrow w[x]_i^b$

* Functia de calcul: paralelism maxim sau limitat, secvential

Ce Calculeaza P Sistemele

$NOP_n(x)$ – familia de multimi de numere naturale calculate de P sistemele de tip celular (cell P systems) cu cel mult *n* membrane (compartimente); * - numar arbitrar de componente;

x = ncoo – reguli independente de context (ne-cooperative, un obiect in lhs), x = coo – dependente de context

(1) $NOP_*(ncoo) = NOP_1(ncoo) = NCF$ – cele mai simple P sisteme

(2) $NOP_*(coo) = NOP_1(coo) = NRE$

Obs.

- Doar un compartiment (membrana/regiune) este necesar(a) – ierahia nu conteaza
- Fara comunicare
- Partial reflectata ierarhia Chomsky

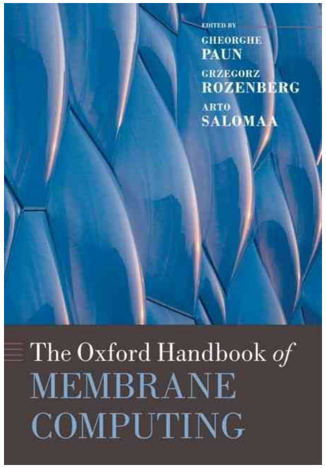
Rezultate Teoretice si Aplicatii

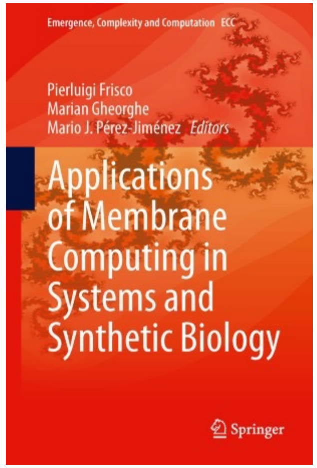
- Investigatii teoretice: putere de calcul, complexitate pentru zecile de variante de P sisteme; relatii cu alte modele de calcul (retele Petri, X-masini, R sisteme) – domeniu bine definit.

- Aplicatii
- Algoritmi de cautare si sortare, probleme de comunicare - broadcasting
- Probleme de sincronizare (sincronizare celulara, Byzantine agreement)
- Probleme NP-complete (Subset sum, drum Hamiltonian, SAT)
- Probleme cu agenti modelati ca P sisteme
- Bio-chimie: circuite de semnalizare, retele de gene – variante stocastice

Publicatii

://dl.acm.org/doi/10.1145/3431234





A survey of nature-inspired computing: Membrane computing, ACM Computing Survey, 54, 1, January 2022, pp 1-31

Bulletin IMCS: http://membranecomputing.net/IMCSBulletin/

Application: Synchronisation Problem

Synchronise the compartments of a membrane system (initially defined for one dimensional cellular automaton)

Problem: membrane system with n compartments and given initial states; provide a solution to the problem of finding a state of the system whereby every compartment will contain the same multiset and this is obtained for the first time

Solution idea: a nondeterministic solution consisting of

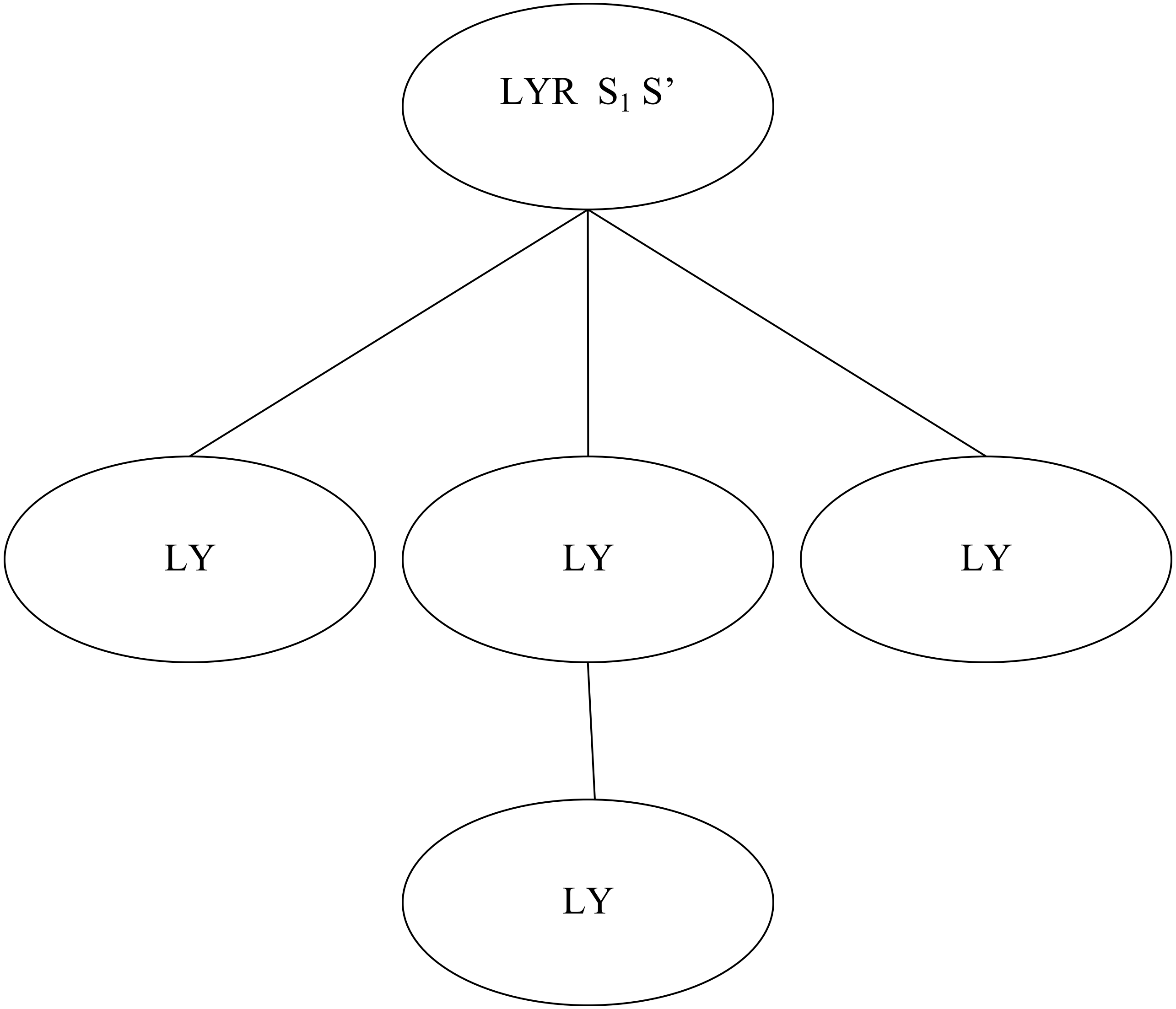
a. a signal travels down on the longest path
b. returns back by counting the number of regions
c. goes down by spreading the same number of objects at each level and decreasing by one when a new level is entered
d. finally the synchronisation state is reached

Bernardini, Gheorghe, Margenstern, Verlan; 2008, IJFCS

Modelul Folosind P Sisteme

P sistem de tip celular/arbore (3 niveluri), avand

- Reguli de rescriere si comunicare $x \rightarrow (y,here)(z,out)(w,in);$ uneori in^*
- Compartimente cu aceleasi reguli
- Multiseturile initiale sunt aceleasi, mai putin radacina (compartimentul exterior)
- Simbolul (obiectul) a numara adancimea sistemului
- Sistemul isi sincronizeaza toate compartimentele atunci cand F este in fiecare dintre acestea



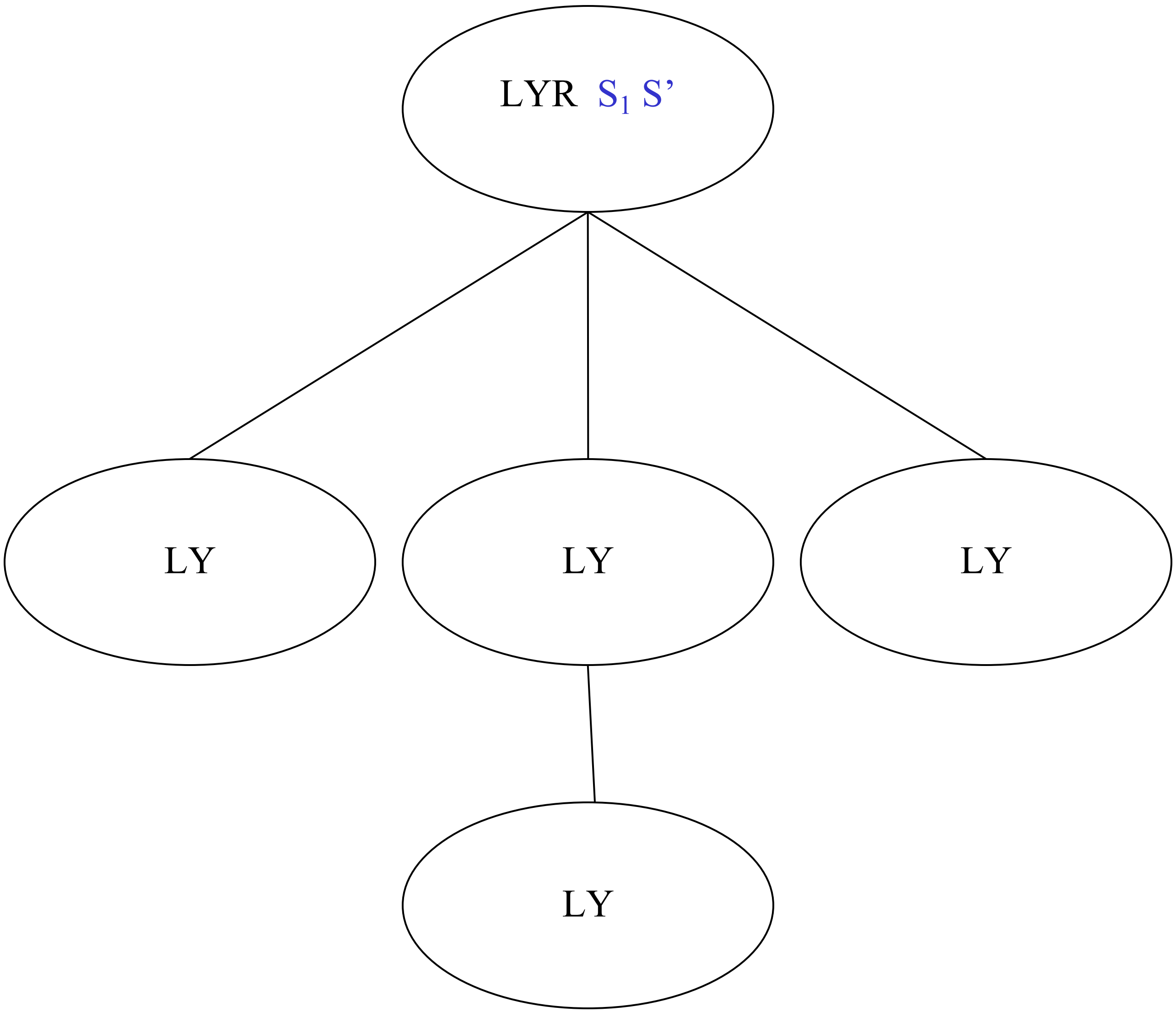
$S_1 \rightarrow (S_1, in)$
$S' \rightarrow (S_2, here)$
$S_2 \rightarrow (S_2, in)$

$S_1\ S_2 \rightarrow (S_3 a, out)$
$S_3 \rightarrow (S_3 a, out)$
$a \rightarrow (a, out)$
$S_3 R \rightarrow (S_5 b, here)$
$a \rightarrow (b, here)$

$LYS_5 b \rightarrow (S_5, in^*)(S_6, here)$
$LYS_5 b \rightarrow (S_7, here)$
$b \rightarrow (b, in^*)(b', here)$

$S_6 b' \rightarrow (S_6, here)$
$S_7 b \rightarrow (S_7, here)$

$S_6 \rightarrow F$
$S_7 \rightarrow F$



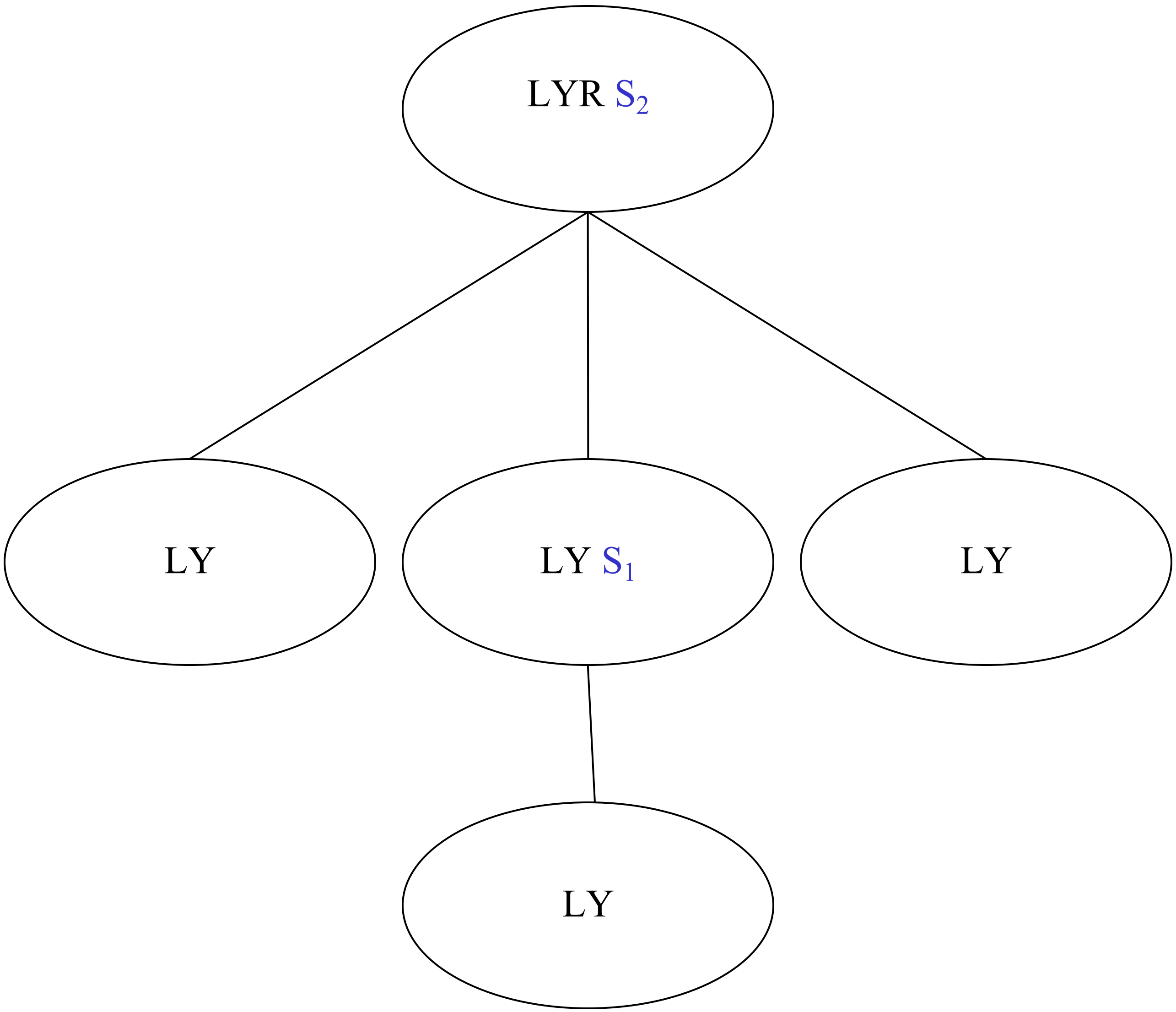
$S_1 \rightarrow (S_1,\mathrm{in})$
$S' \rightarrow (S_2,\mathrm{here})$
$S_2 \rightarrow (S_2,\mathrm{in})$

$S_1\ S_2 \rightarrow (S_3a,\mathrm{out})$
$S_3 \rightarrow (S_3a,\mathrm{out})$
$a \rightarrow (a,\mathrm{out})$
$S_3R \rightarrow (S_5b,\mathrm{here})$
$a \rightarrow (b,\mathrm{here})$

$LYS_5b \rightarrow (S_5,\mathrm{in}^*)(S_6,\mathrm{here})$
$LYS_5b \rightarrow (S_7,\mathrm{here})$
$b \rightarrow (b,\mathrm{in}^*)(b',\mathrm{here})$

$S_6b' \rightarrow (S_6,\mathrm{here})$
$S_7b \rightarrow (S_7,\mathrm{here})$

$S_6 \rightarrow F$
$S_7 \rightarrow F$



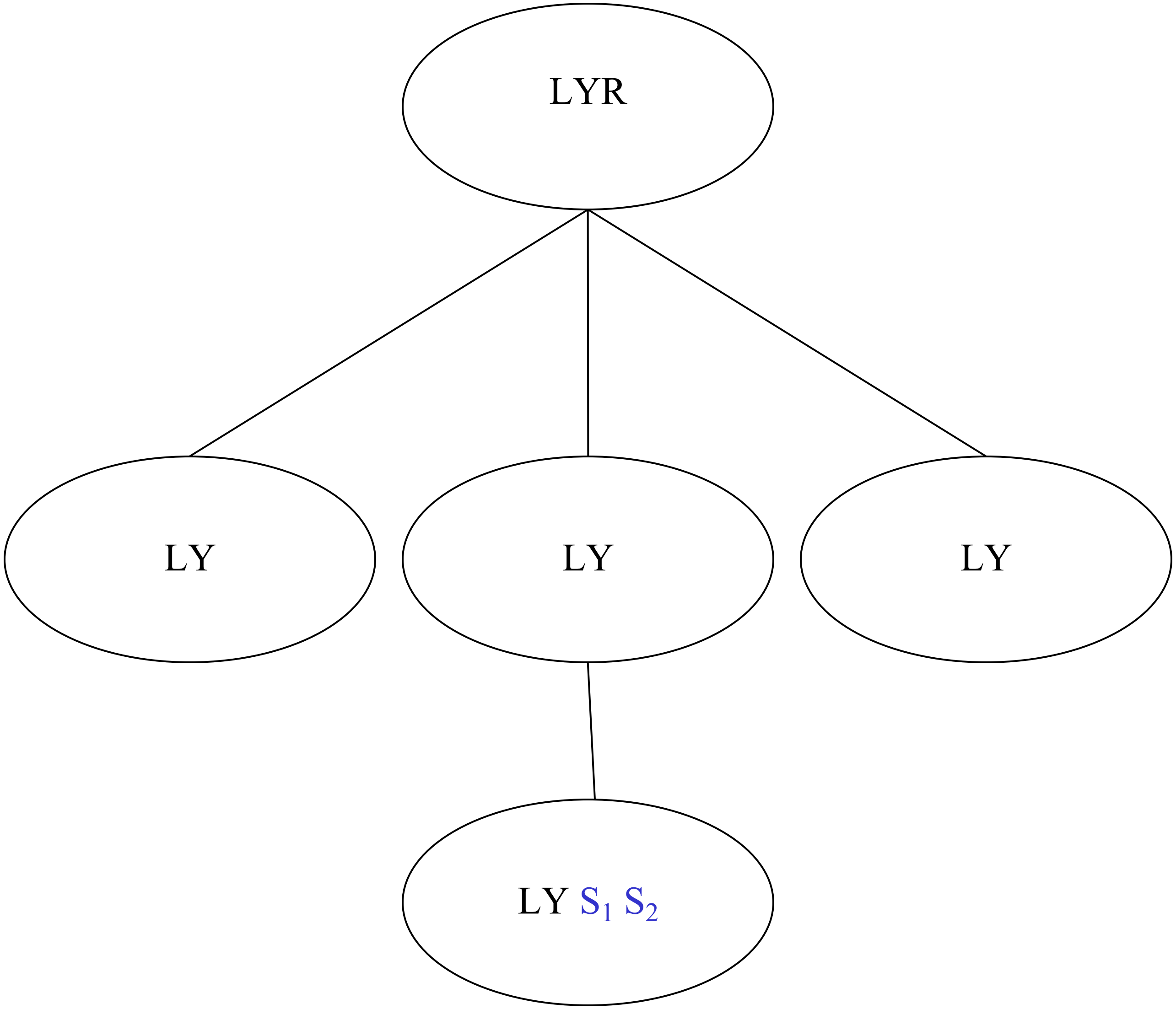
$S_1 \rightarrow (S_1,\text{in})$
$S' \rightarrow (S_2,\text{here})$
$S_2 \rightarrow (S_2,\text{in})$

$S_1\ S_2 \rightarrow (S_3a,\text{out})$
$S_3 \rightarrow (S_3a,\text{out})$
$a \rightarrow (a,\text{out})$
$S_3R \rightarrow (S_5b,\text{here})$
$a \rightarrow (b,\text{here})$

$LYS_5b \rightarrow (S_5,\text{in}^*)(S_6,\text{here})$
$LYS_5b \rightarrow (S_7,\text{here})$
$b \rightarrow (b,\text{in}^*)(b',\text{here})$

$S_6b' \rightarrow (S_6,\text{here})$
$S_7b \rightarrow (S_7,\text{here})$

$S_6 \rightarrow F$
$S_7 \rightarrow F$



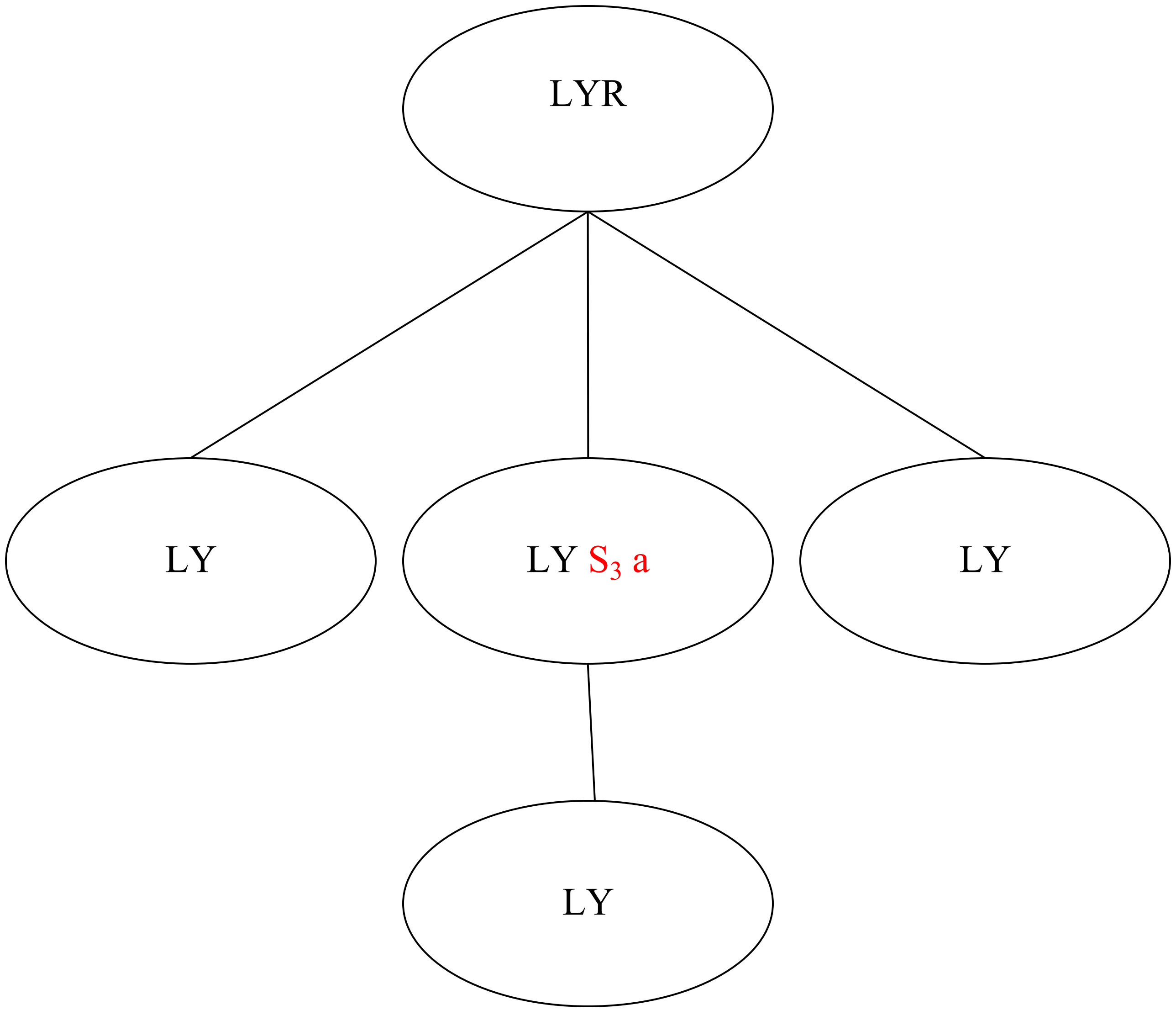
$S_1 \rightarrow (S_1,in)$
$S' \rightarrow (S_2,here)$
$S_2 \rightarrow (S_2,in)$ ___________

$S_1\ S_2 \rightarrow (S_3a,out)$
$S_3 \rightarrow (S_3a,out)$
$a \rightarrow (a,out)$
$S_3R \rightarrow (S_5b,here)$
$a \rightarrow (b,here)$___________

$LYS_5b \rightarrow (S_5,in^*)(S_6,here)$
$LYS_5b \rightarrow (S_7,here)$
$b \rightarrow (b,in^*)(b',here)$____

$S_6b' \rightarrow (S_6,here)$
$S_7b \rightarrow (S_7,here)$_________

$S_6 \rightarrow F$
$S_7 \rightarrow F$



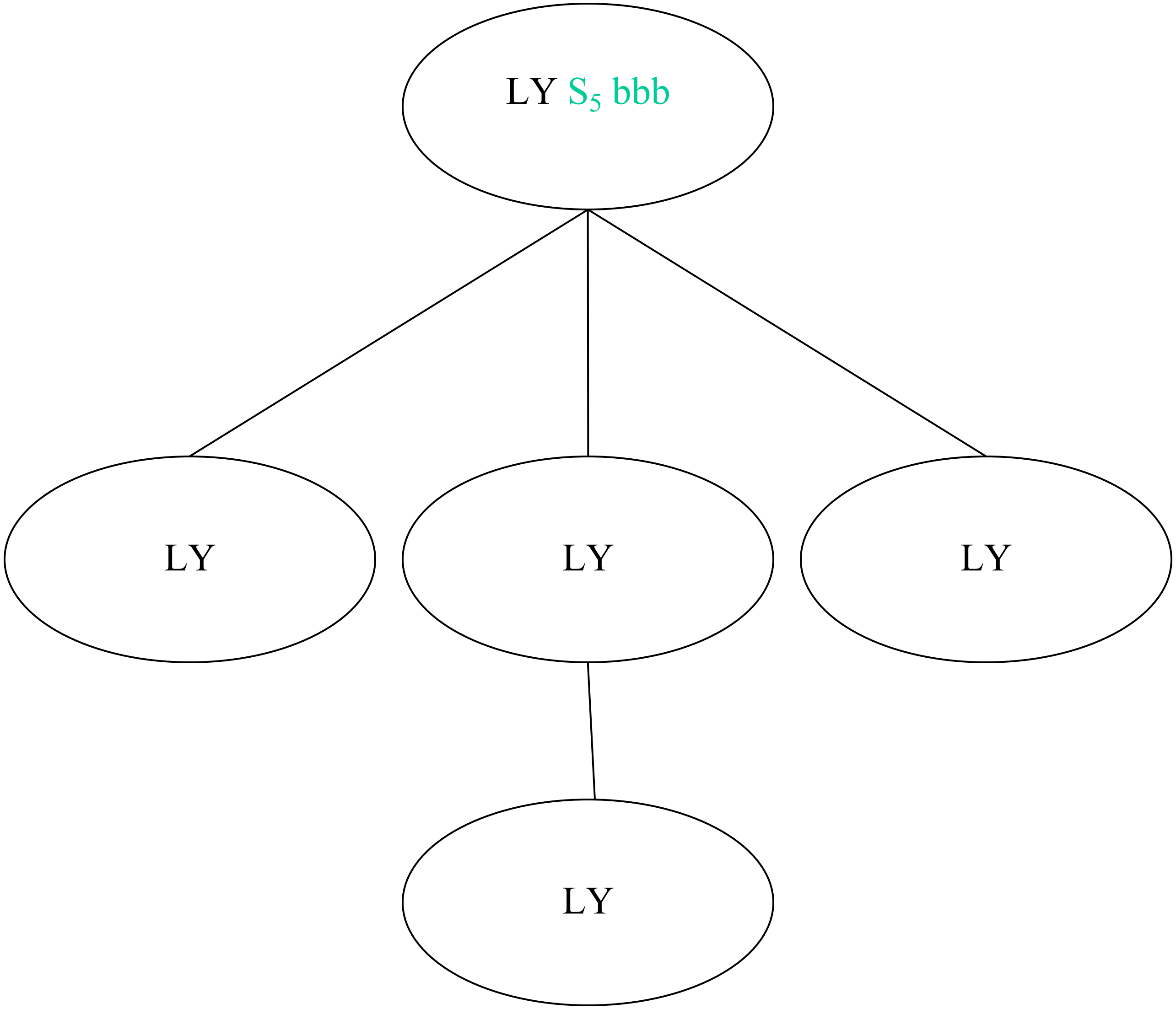
$S_1 \rightarrow (S_1,in)$
$S' \rightarrow (S_2,here)$
$S_2 \rightarrow (S_2,in)$ ___________

$S_1\ S_2 \rightarrow (S_3a,out)$
$S_3 \rightarrow (S_3a,out)$
$a \rightarrow (a,out)$
$S_3R \rightarrow (S_5b,here)$
$a \rightarrow (b,here)$___________

$LYS_5b \rightarrow (S_5,in^*)(S_6,here)$
$LYS_5b \rightarrow (S_7,here)$
$b \rightarrow (b,in^*)(b',here)$_____

$S_6b' \rightarrow (S_6,here)$
$S_7b \rightarrow (S_7,here)$_________

$S_6 \rightarrow F$
$S_7 \rightarrow F$



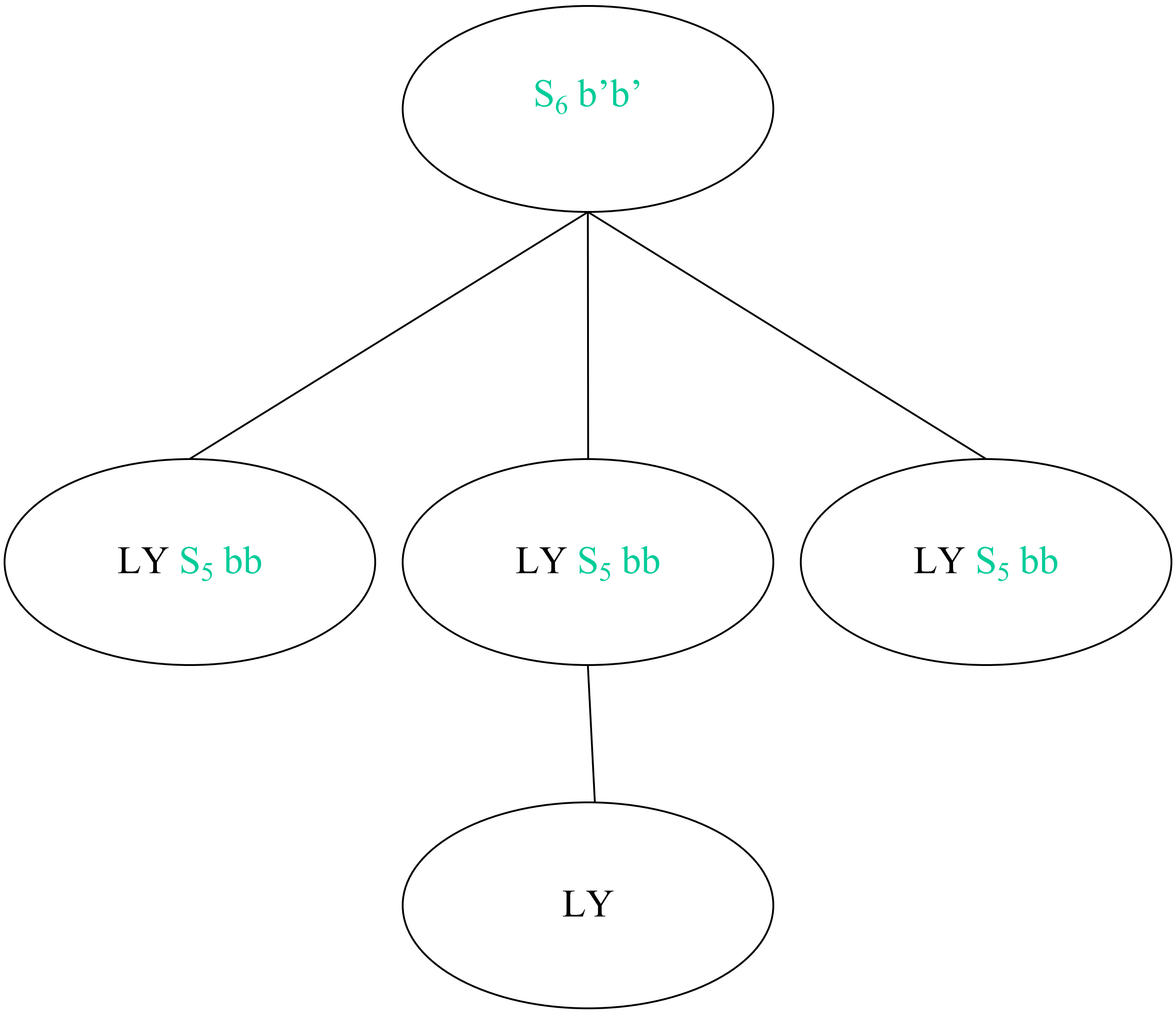
$S_1 \to (S_1,in)$
$S' \to (S_2,here)$
$S_2 \to (S_2,in)$ ___________

$S_1\ S_2 \to (S_3a,out)$
$S_3 \to (S_3a,out)$
$a \to (a,out)$
$S_3R \to (S_5b,here)$
$a \to (b,here)$ ___________

$LYS_5b \to (S_5,in^*)(S_6,here)$
$LYS_5b \to (S_7,here)$
$b \to (b,in^*)(b',here)$ _____

$S_6b' \to (S_6,here)$
$S_7b \to (S_7,here)$ _________

$S_6 \to F$
$S_7 \to F$



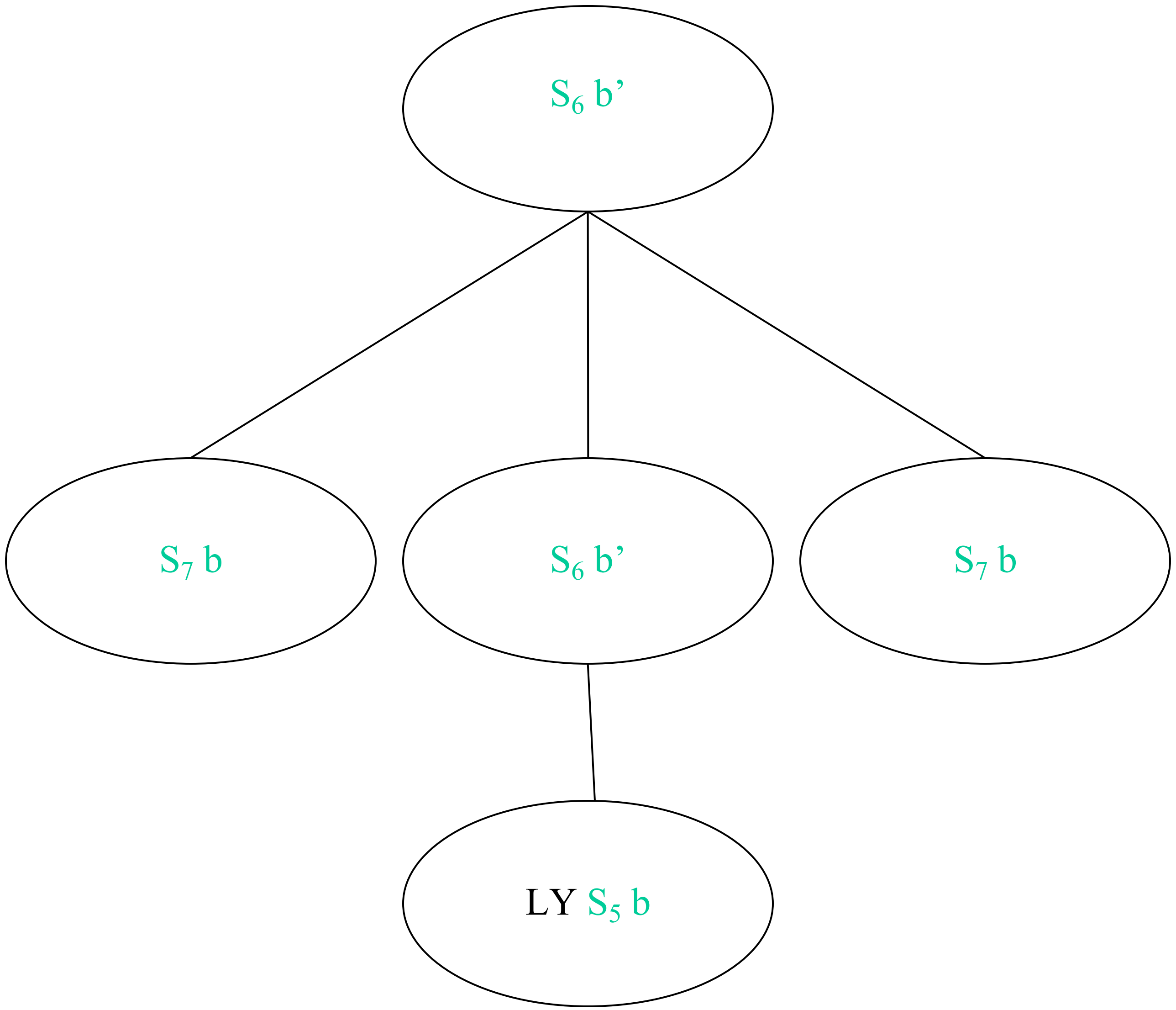
$S_1 \rightarrow (S_1, in)$
$S' \rightarrow (S_2, here)$
$S_2 \rightarrow (S_2, in)$

$S_1\ S_2 \rightarrow (S_3 a, out)$
$S_3 \rightarrow (S_3 a, out)$
$a \rightarrow (a, out)$
$S_3 R \rightarrow (S_5 b, here)$
$a \rightarrow (b, here)$

$LYS_5 b \rightarrow (S_5, in^*)(S_6, here)$
$LYS_5 b \rightarrow (S_7, here)$
$b \rightarrow (b, in^*)(b', here)$

$S_6 b' \rightarrow (S_6, here)$
$S_7 b \rightarrow (S_7, here)$

$S_6 \rightarrow F$
$S_7 \rightarrow F$



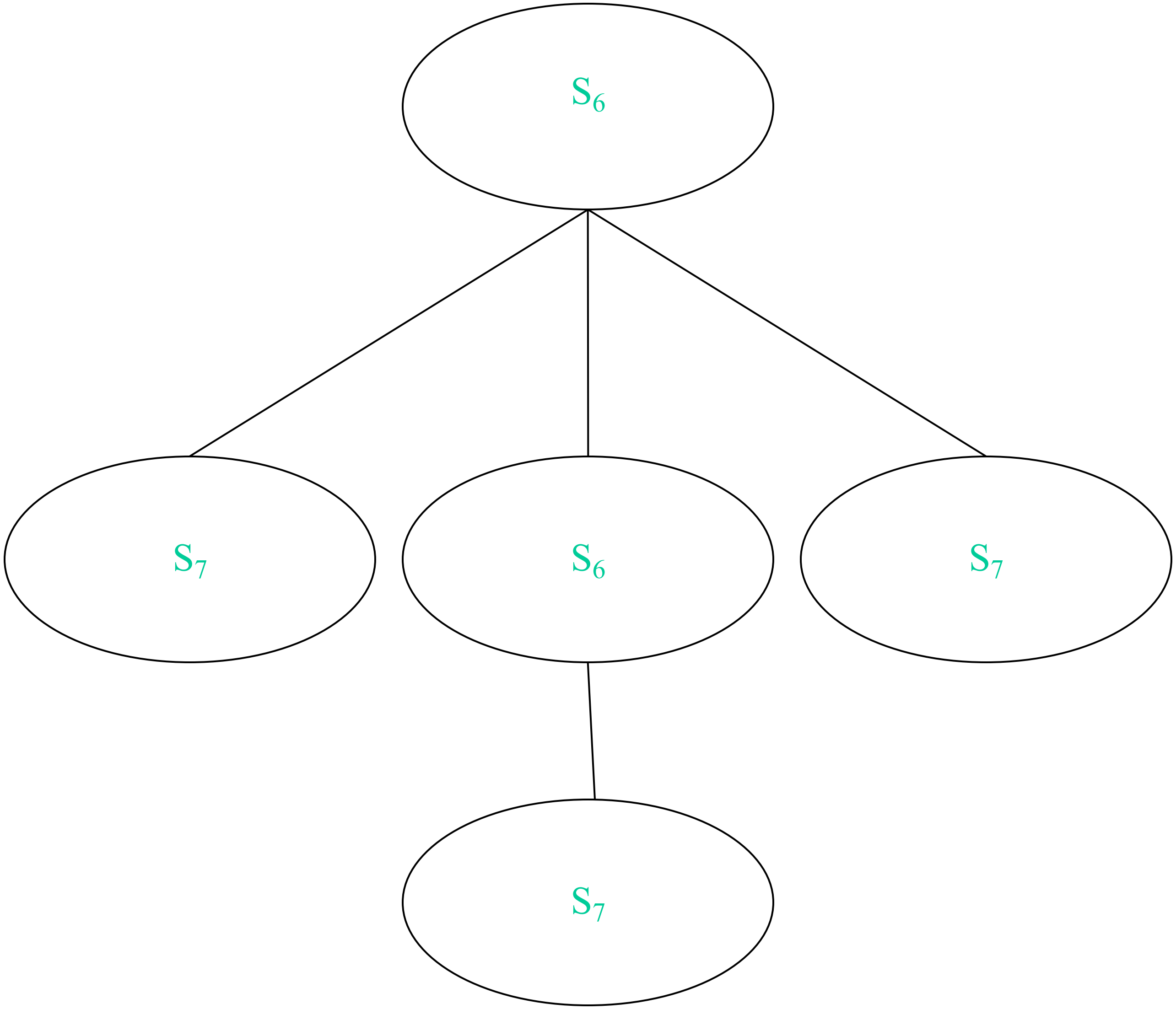
$S_1 \rightarrow (S_1,in)$
$S' \rightarrow (S_2,here)$
$S_2 \rightarrow (S_2,in)$ ___________

$S_1\ S_2 \rightarrow (S_3a,out)$
$S_3 \rightarrow (S_3a,out)$
$a \rightarrow (a,out)$
$S_3R \rightarrow (S_5b,here)$
$a \rightarrow (b,here)$___________

$LYS_5b \rightarrow (S_5,in^*)(S_6,here)$
$LYS_5b \rightarrow (S_7,here)$
$b \rightarrow (b,in^*)(b',here)$____

$S_6b' \rightarrow (S_6,here)$
$S_7b \rightarrow (S_7,here)$_________

$S_6 \rightarrow F$
$S_7 \rightarrow F$



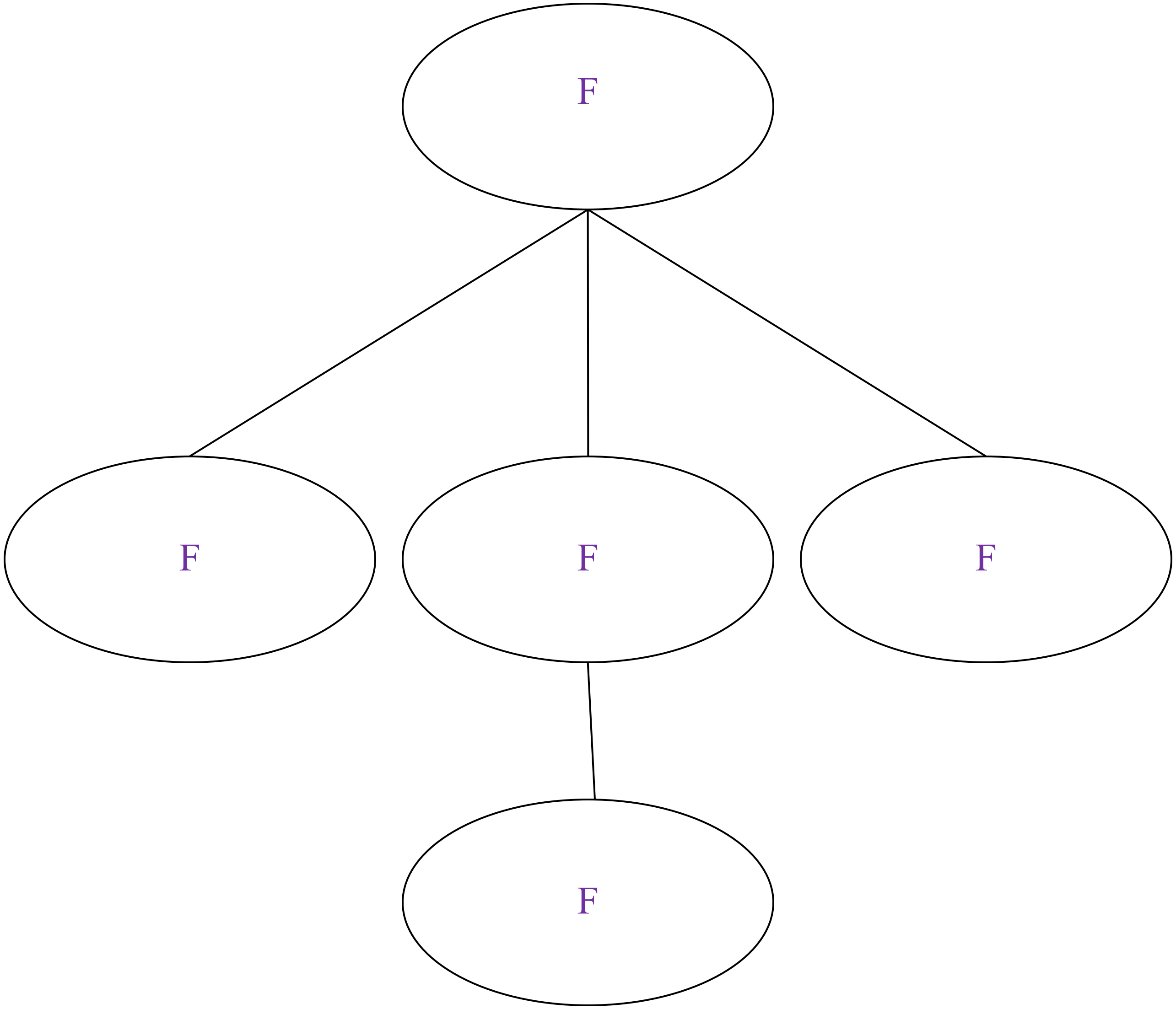
$S_1 \rightarrow (S_1,in)$
$S' \rightarrow (S_2,here)$
$S_2 \rightarrow (S_2,in)$ ___________

$S_1\ S_2 \rightarrow (S_3a,out)$
$S_3 \rightarrow (S_3a,out)$
$a \rightarrow (a,out)$
$S_3R \rightarrow (S_5b,here)$
$a \rightarrow (b,here)$___________

$LYS_5b \rightarrow (S_5,in^*)(S_6,here)$
$LYS_5b \rightarrow (S_7,here)$
$b \rightarrow (b,in^*)(b',here)$_____

$S_6b' \rightarrow (S_6,here)$
$S_7b \rightarrow (S_7,here)$_________

$S_6 \rightarrow F$
$S_7 \rightarrow F$



$S_1 \rightarrow (S_1, in)$
$S' \rightarrow (S_2, here)$
$S_2 \rightarrow (S_2, in)$ ___________

$S_1\ S_2 \rightarrow (S_3 a, out)$
$S_3 \rightarrow (S_3 a, out)$
$a \rightarrow (a, out)$
$S_3 R \rightarrow (S_5 b, here)$
$a \rightarrow (b, here)$___________

$LYS_5 b \rightarrow (S_5, in^*)(S_6, here)$
$LYS_5 b \rightarrow (S_7, here)$
$b \rightarrow (b, in^*)(b', here)$_____

$S_6 b' \rightarrow (S_6, here)$
$S_7 b \rightarrow (S_7, here)$_________

$S_6 \rightarrow$ F
$S_7 \rightarrow$ F

Observatii

Problema este o generalizare a unei paradigme utilizate spre a exemplifica sincronizarea nodurilor unei retele de procese distribuite care comunica local, intre vecini, fara control global – cunoscuta si ca 'firing squad synchronization problem'; modelata de obicei cu automate celulare, celule asezate pe o linie.

Premize: aceleasi reguli (rescriere/comunicare si un broadcasting local), stari initiale specific

Final: aceeasi stare in fiecare compartiment (proces)

Complexitate: Solutia este lineara in raport cu adancimea sistemului (lungimea celui mai lung drum in arbore)

Kernel P System – Introducere Informala

• Structura dinamica, sub forma de graf

• Utilizeaza multiseturi de obiecte

• Regulile pot avea garzi (conditii booleene)
  • rescriere si comunicare
  • structurale (ex. diviziune celulara/de membrane)

• Fiecare compartiment are un tip din care deriva, iar tipul compartimentului defineste setul de reguli. Fiecare instanta are propriul multiset initial.

• Vom ilustra utilizarea modelului cu o problema NP-completa - 'Partition problem'.

Partition Problem

Given a finite set $V=\{v_1, \ldots, v_n\}$, a function *weight* on V with positive integer values,

$weight : V \rightarrow Z^+$, such that $weight(v_i)=k_i$,

and for any subset W of V, $weight(W)$ means the sum of the weights of the elements of W.

For a given k. from Z^+ decide whether there exists a partition V_1, V_2 of V, such that $weight(V_1) = weight(V_2) = k$.

Obviously, $weight(V) = 2k$.

Schita Solutiei

Ideea. Se genereaza toate submultimile – brute force. Se verifica daca una dintre submultimile stricte, W, are proprietatea $weight(W)=k$.

Modelul. kernel P (kP) system. Vom folosi un kP sistem cu doua tipuri, t_1 si t_2, din care initializam doua instante It_1 si It_2. It_1 va functiona ca o interfata unde primim raspunsul, iar It_2 va fi masinaria supusa procesului de divizare celulara unde obtinem toate submultimile. Cele 2^n submultimi se obtin in n pasi, simuland o recursie.

Regulile recursiei:

$[A_i]_2 \rightarrow [B_i\ A_{i+1}]_2 [A_{i+1}]_2$; $1 \leq i < n$ – cazul iterativ

$[A_n]_2 \rightarrow [B_n\ X]_2 [X]_2$; -- cazul de baza

Complexitatea algoritmului: lineara in raport de cardinalul lui V.

kP Sisteme: Tipuri

Se dau n compartimente, fiecare cu un tip t_i, $i=1, \ldots , n$. Fiecare t_i este dat de un set de reguli, R_i, si o strategie de executie, s_i.

Strategia, s_i, poate fi paralelism maxim, arbitrar, executie secventiala, selectie.

Obs.

1. Cand sunt instantiate tipurile atunci se introduce multiseturile initiale.
2. kP sistemul obtinut poate avea *diferite strategii de executii* in compartimente.
3. Regulile de comunicare implica tipuri, iar la executie se alege o instanta arbitrara, daca sunt mai multe de acelasi tip (nedeterminism).

Modelul de Tip kP Sistem (1)

Modelul folosit pentru Partition Problem implica doua tipuri

- t_1 pentru interfata
- t_2 pentru generarea tuturor submultimilor.

Strategia de executie, s_i, $i=1,2$, este **paralelism maxim**.

Modelul de Tip kP Sistem (2)

$kP=(A, \mu, C_1, C_2, 0)$, unde μ conecteaza C_1, C_2, de tipuri t_1 si respectiv t_2. Rezultatul se obtine in mediu.

Regulile tipului t_1, R_1 (trimit raspunsul in mediu, identificat cu 0)

$r_{1,1}$: $S \rightarrow (yes,0)\ \{\geq T\}$ – exista cel putin o solutie

$r_{1,2}$: $S \rightarrow (no,0)\ \{\geq F \wedge <T\}$ – nu exista solutie

Regulile tipului t_2, R_2, sunt

- diviziune de membrane (genereaza 2^n compartimente de tip t_2)

$r_{2,i}$: $[A_i\,]_2 \rightarrow [B_i\ A_{i+1}\,]_2\,[A_{i+1}\,]_2\ ;\ 1 \leq i < n$

$r_{2,n}$: $[A_n\,]_2 \rightarrow [B_n\ X\,]_2\,[X\,]_2\ ;$

- rescriere (identifica un subset care se "potriveste" cu complementara - ponderi)

$r_{2,i,j}$: $v_i\ v_j \rightarrow v\ \{=B_i \wedge\ \neq B_j \wedge =X \vee =B_j \wedge\ \neq B_i \wedge =X\},\ 1 \leq i < j \leq n$

$r_{2,n+1}$: $X \rightarrow Y$

- Comunicare (F – nu este solutie; T – solutie)

$r_{2,n+2}$: $Y \rightarrow (F,1)\ \{\geq v_1 \vee \ldots \geq v_n \vee \neq v^k\}$ – apare cel putin un v_i sau v-uri in nr $\neq k$

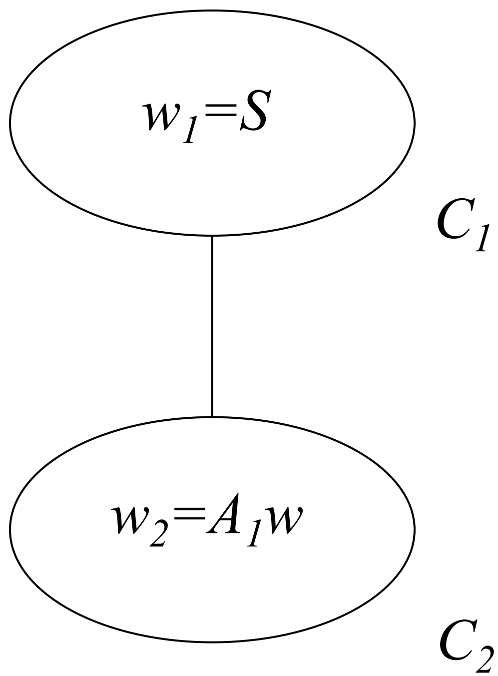
$r_{2,n+2}$: $Y \rightarrow (T,1)\ \{<v_1 \wedge \ldots <v_n \wedge =v^k\}$ – niciun v_i si v-uri in numar de k

Modelul de Tip kP Sistem (3)

Multiseturile initiale din C_1, C_2, sunt

$w_1 = S$

$w_2 = A_1 v_1^{k1} \ldots v_n^{kn} = A_1 w$ (notatie)

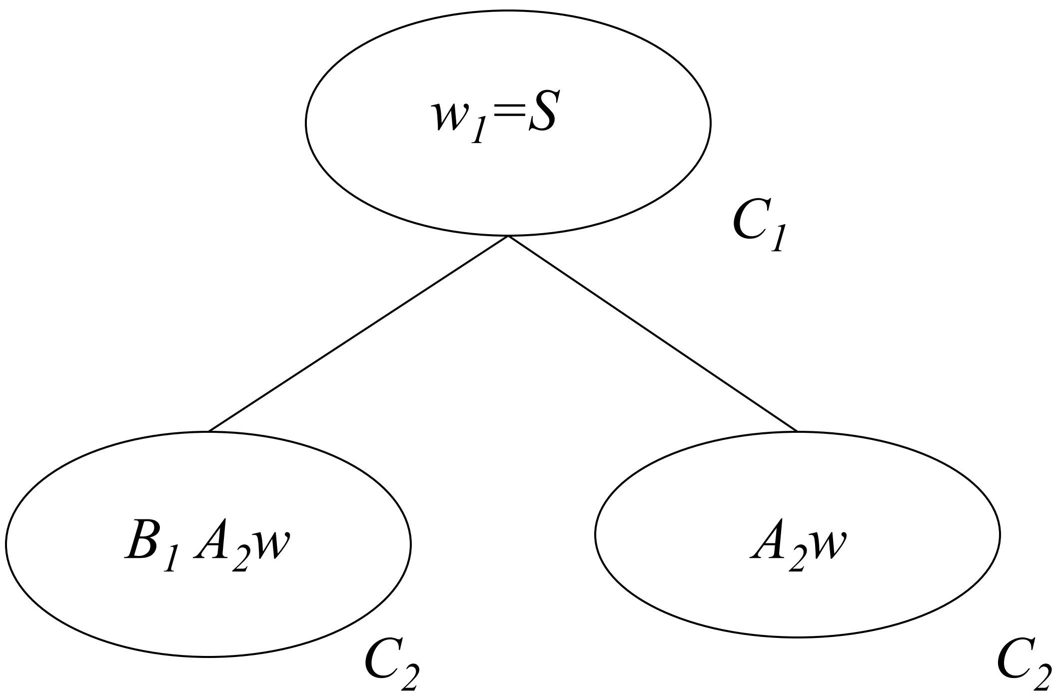


Pasul 1

Aplicam in C_2,

$r_{2,1}$: $[A_1]_2 \rightarrow [B_1\ A_2]_2\ [A_2]_2$

Se obtin doua compartimente C_2 – submultimile cu max 1element

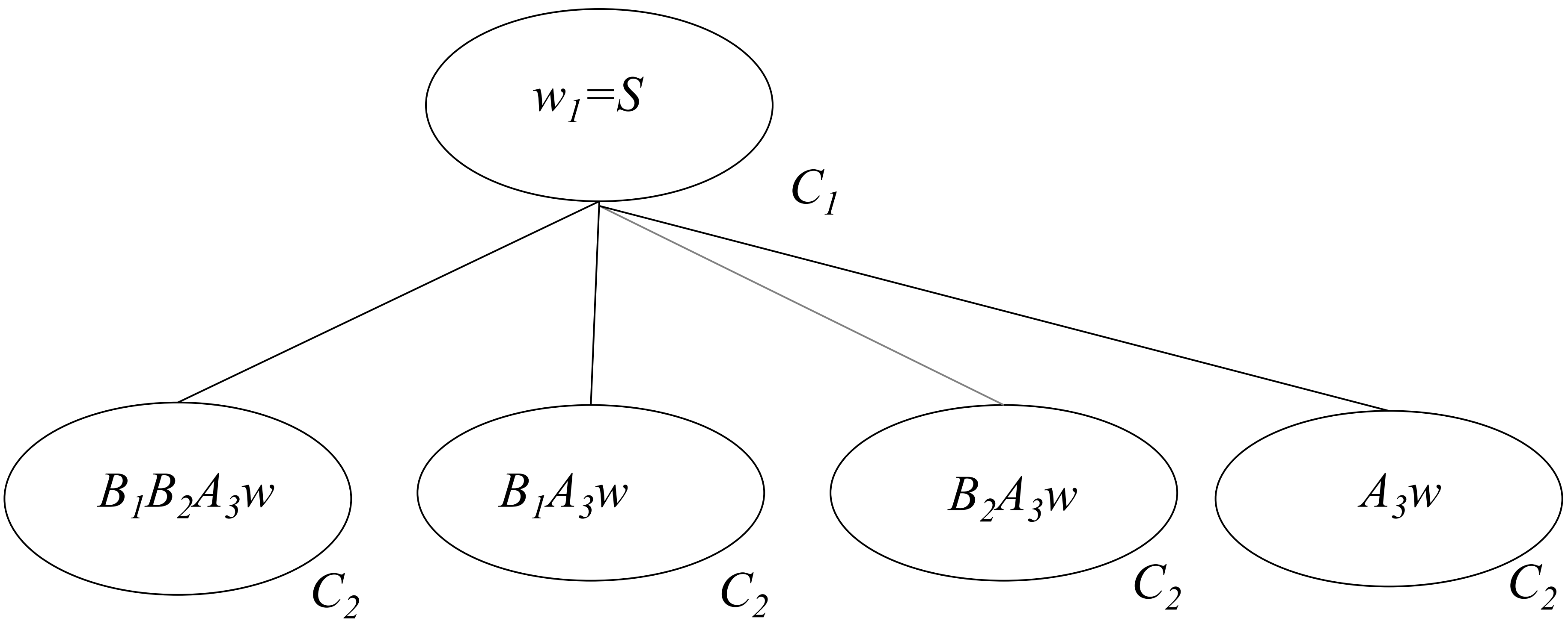


Pasul 2

Aplicam in C_2,

$r_{2,2}$: $[A_2\,]_2 \rightarrow [B_2\;A_3\,]_2\,[A_3\,]_2$

Se obtin patru compartimente C_2 – submultimile cu max 2 elemente



Pasul n

Aplicam in C_2,

$r_{2,n}$: $[A_n\,]_2 \rightarrow [B_n\ X]_2\,[X\,]_2$

Se obtin 2^n compartimente C_2 – submultimile cu max n elemente. Compartimentele C_2 sunt toate legate de C_1

Pasii Finali

Aplicam in C_2,

$r_{2,i,j}$; $r_{2,n+1}$ si apoi una dintre $r_{2,n+2}$ sau $r_{2,n+3}$

In final $r_{1,1}$ sau $r_{1,2}$ in C_1 trimite rezultatul (*yes* sau *no*) in mediu.

Concluzii

- Modelarea si relatia cu Ingineria Software

- Modele cu rescriere. Sisteme cu Membrane

- Cum se modeleaza cu Sistemele cu Membrane (exemple)

- Varietate de aplicatii

- Alte topici: (1) verificarea si testarea folosind sistemele cu membrane; (2) instrumente de modelare, verificare si testare