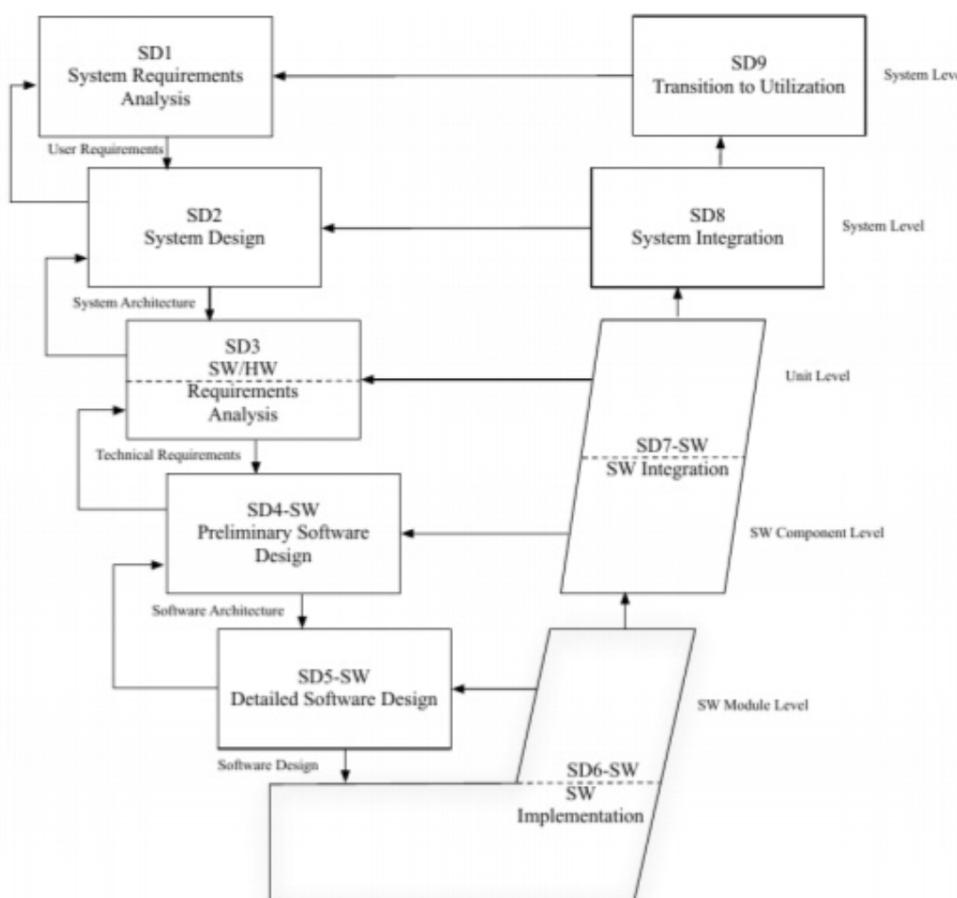


È POSSIBILE USARE LE PTPN COME NUCLEO PER LA
REALIZZAZIONE DI APPROCCIO **MODEL DRIVEN DEVELOPMENT** CHE
PERMETTE, AD ESEMPIO, DI AUTOMATIZZARE LA GENERAZIONE DEL
CODE.

QUESTO APPROCCIO È REUSABILE MANTENENDO UN CYCLE DI SVILUPPO DEL
CODE BASATO SULLE PRATICHE INDUSTRIALI DEL V-MODEL

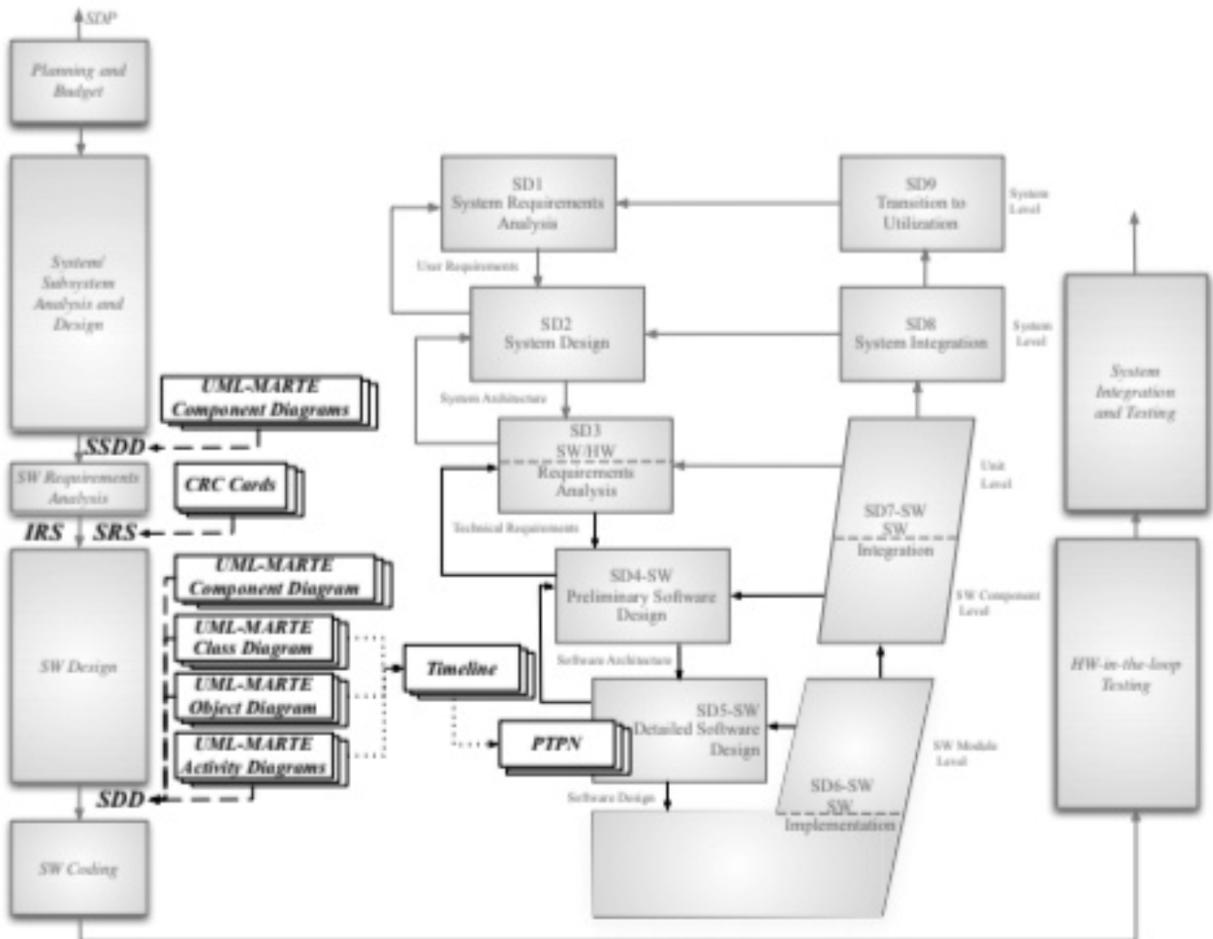


SULLA PARTE CHE SERVIRÀ ABBONAMENTO SI HA L'INTEGRAZIONE
NE DI DESIGN E FEEDBACK MENTRE DALL'alto verso il basso
SI HA LA SCOMPOSIZIONE DEL SISTEMA IN MODULI

AL V-MODEL SI VUONO AD INTEGRARE ULTERIORI METODI FORMALI
DI SVILUPPO CHE SERVANO DELL'ESECUZIONE DELLE PROCEDURE SPECIFICHE, COME
AD ESEMPI IL PROCESSO DI DOCUMENTAZIONE V-STD-998

AS ESCREVO I CICLOS DI DESARROLLO CON IL V-MODEL

SUPPORTATO DA UNA MODELLIZZAZIONE TRAVIG UML-MARTE



VEDIAMO BREVIAMENTE I PASSI CHE SONO DESCRITTI NEL V-MODEL:

- SD1 (SYSTEM REQUIREMENTS ANALYSIS)

CONSISTE NELLA DETERMINAZIONE DEI REQUISITI UTENTE AD ALTO LIVELLO DEL SISTEMA

- SD2 (SYSTEM DESIGN)

IDENTIFICA LE UNITÀ DEL SISTEMA E VI ALLOCASI I REQUISITI DI SISTEMA

SD3 (SW / HW REQUIREMENTS Analysis)

DETERMINARE VERSO CUI SI DEDICHERÀ L'ANALISI DI:
HARDWARE CONFIGURATION ITEMS (HCI), COMPUTER SOFTWARE CONFIGURATION ITEMS (CSCI) E
FIRMWARE CONFIGURATION ITEMS (FCI).

SD4-SW (PRELIMINARY SW DESIGN)

DEFINIRE L'ARCHITETTURA SOFTWARE DI Ciascun CSCU CON
LE INSIEME DI TUTTI I COMponenti CON LEIRI FUNZIONALI
ASSOCIAZIONI E I TEMPI DI RISORSO / DOGANDI PRESORTI.

SD5-SW (DETAILED SW DESIGN)

DEFINIRE IL DESIGN SOFTWARE DI Ciascun CSCU
NELL'AMBITO RISORSE E TIME-REQUIREMENTS A Ciascun modulo
SOFTWARE.

SD5.2-SW (ANALYSIS OF RESOURCES AND TIME REQUIREMENTS)

SI ESEGUE UNA SIMULAZIONE DEL MODELO PTPN E SI ESEGUE
L'ANALISI DELLO SPAZIO DEGLI STATI.

SD6-SW (SW IMPLEMENTATION)

SI IMPLEMENTANO LE NORME DI FORZA DEI TASK-SET E LE FUNZIONI DI ENTRY-POINT (mentre con BUSYWAIT) E SI ESEGUE UNIT TESTING FACENDO PROFILING DEL TIEMPO DI ESECUZIONE DI MODULI A BASSO LIVELLO.

SDT-SW (INTEGRATION-SW COMPONENT SCOPES)

IMPLEMENTAZIONE DELLE ATTIVITÀ SUPPORTATE DAL NUCLEO FORTESSA DELL' PTPN, CON VARIETÀ DELL' ORARIO

- **FAILURES**: DEVIAZIONI DI UN COMPONENTE DAL SERVIZIO/RISULTATO ATTESO. POSSONO ESSERE DI VARI TIPI:

UNSEQUENCED EXECUTION: un corso non che va a norme dei regolamenti specifico

TIMEFRAMES VIOLATION: prendo un parametro temporale assoluto dei valori fuori dai loro intervalli nominati.

DEADLING MISS: prendo un job che rispetta i suoi regolamenti temporali

DEFECTS: errori in un componente che possono causare fallimenti nel sistema.

TASK PROGRAMMING DEFECT: errore nel controllo della concorrenza e nell'integrazione dei task

CYCLED SCHEDULING: detrazione di risorse computazionali dovuti a task aggiornati.

AI FINI DEL TESTING SONO PRESENTI DUE STANDARDS DI CERTIFICAZIONE CHE

PRESENTAZIONE DEI CRITERI DI COPERTURA MISURATI TRAMITE IL **CONTROL FLOW GRAPH** DEL CODICE. AD ESEMPIO SI CONTROLLANO TUTTI GLI STATEMENT / DECISIONI / CONDIZIONI. QUESTO PORTA AD UN BUON CONTROLLO SULLE STRUTTURE E SUL DATA-FLOW MA TUTTAVIA LIMITA LA COPERTURA DELLA GRANDE VARIETÀ DI COMPORTAMENTI CHE SI OTTENGONO DALLA CONCERNENZA OPPURE DALL'ESECUZIONE INTEMPORELLA DELLE ARCHITETTURE DIVERGENTI DI UN CSCI.

MANTENENDO IL **STATE CLASS GRAPH** DI UN MODELLO PTAU DI UN CSCI È POSSIBILE ASSESSARE UNA QUANTITÀ DI COMPORTAMENTI DELLE ARCHITETTURE DIVERGENTI DELLE CSC. SI POSSONO PUNTO OTTENERE:

- **GENERATORI DI CODICE PREDEFINITI** => SCG FORNISCE UNA MISURA DI COPERTURA STRUTTURALE
- **GENERATORI DI CODICE NON PREDEFINITI** => SCG FORNISCE UNA MISURA DI FUNZIONALITÀ.

VEDIAMO QUAI POSSONO ESSERE I CRITERI DI SELEZIONE DEI TEST SU SCG:

- **ALL-MARKINGS**: RICHIEDE LA COPERTURA DI OGNI CLASSE MARCATURA RAGGIUNGIBILE E GARantisce la copertura di tutti i possibili stati di concordanza. PER OGNI CLASSE MARCATURA RAGGIUNGIBILE m , SELEZIONA UNA CLASSE S_m E INCLUSA NEL TEST qualunque percorso da un **CONTROLLABILE STARTING POINT CSP** AD S_m .

- **ALL-MARKING ECOS**: RICHIEDE LA COPERTURA DI OGNI ARCO TRA DUE QUALSiasi MARCATURE RAGGIUNGIBILI. GARantisce la copertura di tutte le possibili transizioni fra stati concorrenti. PER OGNI ARCO TRA m_1 E m_2 SELEZIONA UNA CLASSE S_1 CON MARCATURA m_1 CON UN EVENTO CHE PORTA ALLA CLASSE S_2 CON m_2 E INCLUSO NEL TEST qualsiasi percorso partendo dal CSP E

- **ALL-CLASSES**: RICHIEDE LA COPERTURA DI TUTTE LE CLASSI DI STATO RAGGIUNGIBILI.
DISTINGUE GLI STATI DI CONCERNZA ASSOCIAZIONI CON I TEMPI DIVERSI
CEDUTI DALL'ESECUZIONE DI DIVERSE SEQUENZE DI FIRME DI TRANSIZIONI.
- **ALL-CLASS-GOALS**: RICHIEDE LA COPERTURA DI TUTTI GLI STATI DELLA CLASSE DI STATO RAGGIUNGIBILI.
DISTINGUE GLI STATI DI CONCERNZA ASSOCIAZIONI CON I TEMPI DIVERSI
CEDUTI DALL'ESECUZIONE DI DIVERSE SEQUENZE DI FIRME DI TRANSIZIONI.
- **ALL-SYMBOLIC RUNS**: RICHIEDE LA COPERTURA DI TUTTO LE SYMBOLIC RUNS CHE INIZIANO CON IL RILASCO DI UN JOB E TERMINANO O CON IL SUO COMPLETAMENTO O CON UNA DEADLINE MISS. DISTINGUE GLI STATI DI CONCERNZA E LE TRANSIZIONI VISITATE IN DIVERSE ESECUZIONI DI JOB.
- **ALL-SYMBOLIC-EXECUTIONS**: RICHIEDE LA COPERTURA DI QUALSIASI SEQUENZA DI EVENTI CHE INIZIA CON IL RILASCO DI UN JOB E TERMINA CON IL SUO COMPLETAMENTO O CON UNA DEADLINE MISS. NON DISTINGUE PERCORSI CON LA STESSA SEQUENZA DI SPARZO E DIVERSE STARTING CLASS.

COME È POSSIBILE DETERMINARE I TIME INPUTS CHE FORZANO LA IMPLEMENTAZIONE CLOUD TEST (UT) AL FINE DI ESEGUIRE OGNI TEST CASE SELEZIONATO?

È POSSIBILE CONTROLLARE I TEMPI DI RILASCO PERIODICO E ASINCRONI, MENTRE I TEMPI DI COMPUTAZIONE SONO SPESO SOLO PROTICO DA CONTROLLARE.

SÌ HA QUINDI LA PRATICA DEL **GUIDED TESTING** DA SÌ ESPANDE SGAI AL FINE DI IDENTIFICARE LE CLASSI STATO CHE POSSONO ESSERE SCELTI COME STARTING POINT.
SÌ ESPANDONO LE AREE DI UNA TRACCIA DELLE PTPN PER DETERMINE LE RESTRIZIONI TEMPORALI PER LE CLASSI INIZIALI.

. SDT-SW seconda parte:

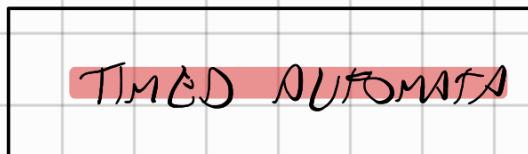
TESTA L'INTEGRAZIONE DI CSCU, HCU E FO ALL'INTERNO DI CIASCUA UNITÀ.

. SD8 (SYSTEM INTEGRATION)

TESTA L'INTEGRAZIONE DI TUTTE LE UNITÀ ALL'INTERNO DEL SISTEMA

. SD9 (TRANSITION TO UTILIZATION)

PORTA IL SISTEMA DA UNO STATO COMPLESSO IN OPERAZIONE NEL SITO DI APPLICAZIONE



PERMETTE ALLA RAPPRESENTAZIONE DI SYSTEM CONCURRENTI CON
L'ESPRESSIONISTICA COMPATIBILE CON LE TPN.

UN TIMED AUTOMATA TA È UN AUTOMA FINITO PROMESSO DA:

. UN INSERIMENTO DI CLOCKS

. VICENZE DI TEMPO SULLE TRANSIZIONI (DETTA AVVENTO/GENERE/CONSUMO CONDITIONS)

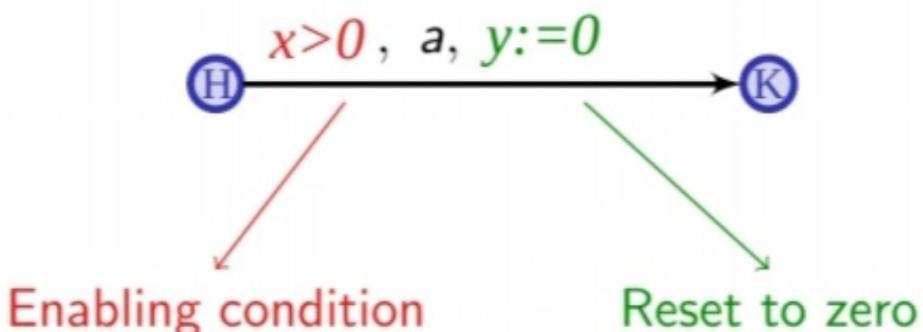
. CLOCK MIGRATIONS SULLE TRANSIZIONI

(SISTEMA)

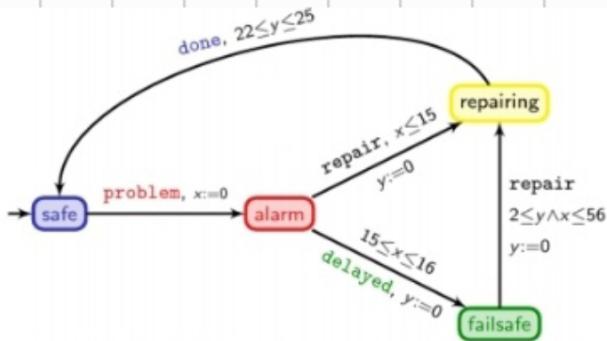
UN TA MINIMALE È DATO DA:

DUE LOCATIONS H E K, DUE CLOCKS X E Y, UNA TRANSIZIONE DA H

A K ABILITA SE $x > 0$, ESIGUENDO DALL'ESTATE A E UN CLOCK DI RESET Y A 0



ESEMPIO:



A possible execution of the TA

safe	$\xrightarrow{23}$	safe	$\xrightarrow{\text{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\text{delayed}}$	failsafe
x	0	23		0		15.6		15.6
y	0	23		23		38.6		0
								...
failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\text{repair}}$	repairing	$\xrightarrow{22.1}$	repairing	$\xrightarrow{\text{done}}$	safe
...	15.6	17.9		17.9		40		40
	0	2.3		0		22.1		22.1

quando un TA è un tupla (L, l_0, Σ, X, E) dove:

- L è l'insieme finito delle loczioni

- $l_0 \in L$ è l'insieme delle loczioni iniziali

Si scrive anche iniziale

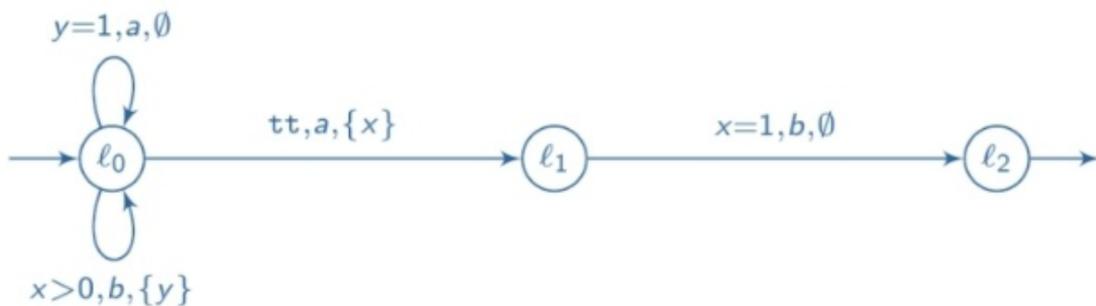
~~NON-HSICRO STATE TRANS~~

X E' l'insieme dei CLOCKS

$E \subseteq L \times \sum G(x) \times 2^X \times L$ E' l'insieme degli ARCS dove
 $G(x)$ e' l'insieme delle CONSTANTS

4

An example with three locations, two actions, two clocks, and two transitions



DEFINIZIONE DI CLOCK INITIATION $V_0 R_{\geq 0}$ CHE ASSIGNA UN CERO VALORE

A chosen clock:

SE UNA VALUTAZIONE V SODDISFA UN CONDIZIONE $\phi \in G(x)$ ALLORA
SI DICE VALID

$$\text{. } V + T \text{ è una valutazione} \quad (V + T)(x) = V(x) + T \quad \forall x \in X, \forall T \in R_{\geq 0}$$

DEFINIZIONE DI STATE DI UN TA CON COPPIA (L, V) DOVE:

- . L è una LOCATION
- . V è una VALUTAZIONE

ESISTONO DUE TIPI DI TRANSIZIONI TRA STATI:

① DELAY TRANSITION: $(L, V) \xrightarrow{T} (L, V + T)$

② DISCRETE TRANSITION: $(L, V) \xrightarrow{\alpha} (L', V')$ SE $\exists (L_0, g, R, L') \in E$ SUCESSO
 $V \models g \in V(x)=\emptyset$ SE $x \in R$ E $v'(x) = v(x)$ SE $x \notin R$

SI DEFINISCE UNA RUN DI UN TA CON SEQUENZA DI NOTIZIANDI DELLA FORMA

DISCRETE TRANSITIONS:

$$(L_0, V_0) \xrightarrow{T_1, z_1} (L_1, V_1) \xrightarrow{T_2, z_2} \dots \xrightarrow{T_n, z_n} (L_n, V_n)$$

SI DEFINISCE UNA TIME SEQUENCE UNA SEQUENZA FINITA DI VALORI
 REALI NON NEGATIVI: $S = (t_1)(t_2) \dots (t_n)$

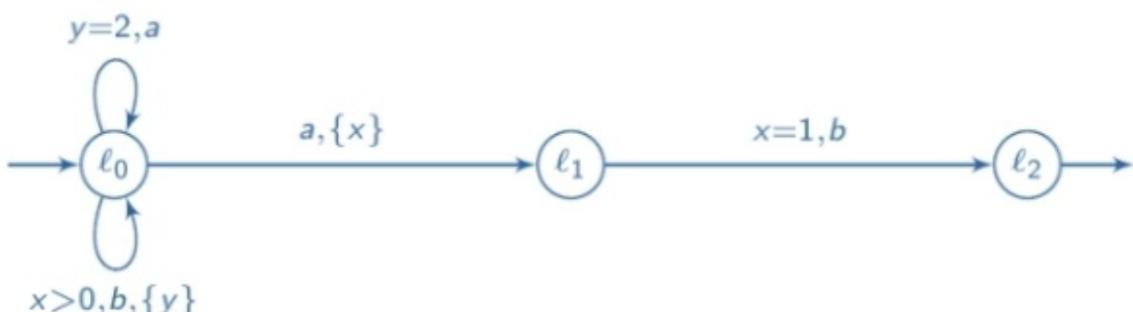
SI DEFINISCE UNA TIMED WORD UNA SEQUENZA FINITA DI COPPIE DI UNA AZIONE E
 UN VALORE TEMPORALE APPARTENENTE AD UNA SEQUENZA TEMPORALE:

$$w = (z_1, t_1)(z_2, t_2) \dots (z_n, t_n)$$

UNA TIMED WORD È ACCETTATA DA UN TA SE $\exists p = (L_0, V_0) \xrightarrow{T_1, z_1} (L_1, V_1) \xrightarrow{T_2, z_2} \dots (L_n, V_n)$
 CON $L_0 = \ell_0$ E $t_i = \sum_{1 \leq j \leq i} T_j$

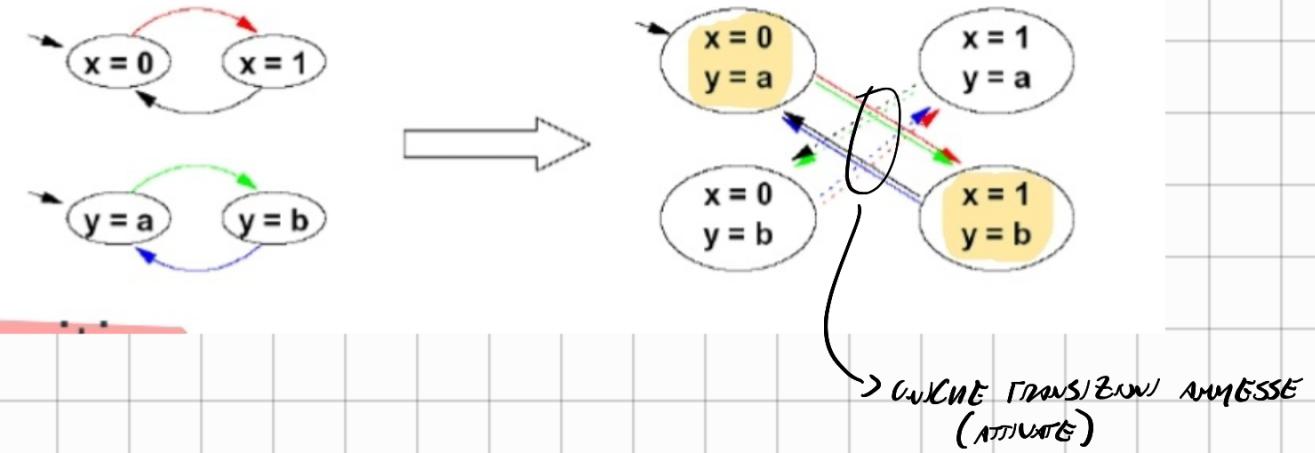
ESEMPIO:

- A run of the example TA is $(l_0, (0, 0)) \xrightarrow{x, y, 0.1, b} (l_0, (0.1, 0)) \xrightarrow{0.2, b} (l_0, (0.3, 0)) \xrightarrow{1, a}$
 $(l_0, (1.3, 1)) \xrightarrow{0.2, b} (l_0, (1.5, 0)) \xrightarrow{0, a} (l_1, (0, 0)) \xrightarrow{1, b} (l_2, (1, 1))$
- An accepted timed word is $w = (b, 0.1)(b, 0.3)(a, 1.3)(b, 1.5)(b, 2.5)$
- Always true guards and empty reset sets are omitted in the representation

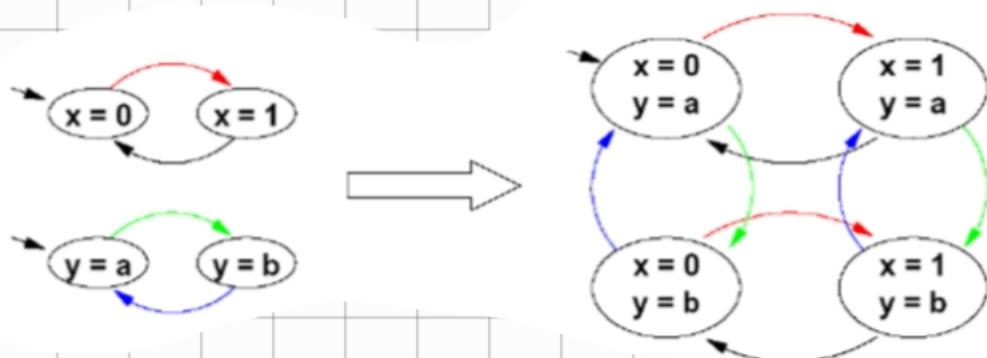


ESISTONO NUMEROSI VARIANTI DEI TA, ALCUNE DI QUESTE SONO:

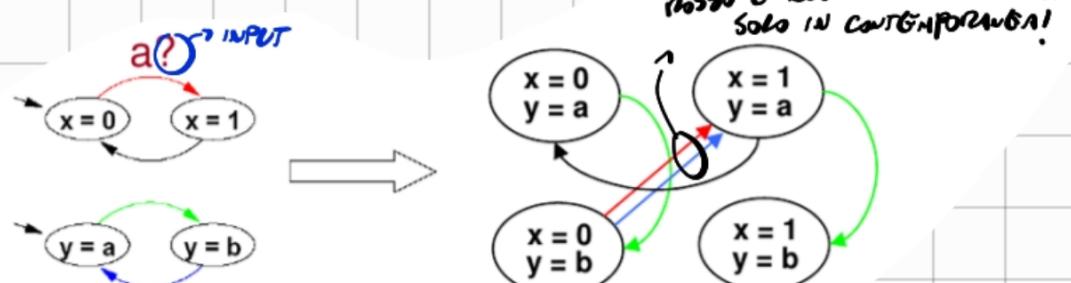
- **COMPOSIZIONE SINCRONA**: AD OGNI PASSO ENTRA(NTI) OLTRE UNO VALORE
MULTIZESTE



- **COMPOSIZIONE ASINCRONA**: OLTRE UNO PASSO POSSONO EVOLOVERSI SEPARATAMENTE IN MODI DIFFERENTI.



- **COMPOSIZIONE CON SINCRONIZZAZIONE ESPULSA**:



a!
dove

PIROZIONAMENTO DELLA REGIONE

$\forall g \in G$ definiamo $[g]$ l'insieme delle valutazioni $\{v \in R_{\geq 0}^x \mid V = g\}$

$\exists \forall Y \subseteq X$ sia $[Y \leftarrow 0]_V$ la valutazione tale che: $[Y \leftarrow 0]_V(x) = 0$ se $x \in Y \in [Y \leftarrow 0]_V(x) = v(x)$ se $x \notin Y$

allora una **PIROZIONE FINITA** R di $R_{\geq 0}^x$ è un insieme di regioni (per l'insieme delle cause G) se:

1. $\forall g \in G$ e $\forall r \in R$ si ha $[g] \subseteq r$ oppure $[g] \cap r = \emptyset$

2. $\forall r, r'$ se $\exists v \in R_{\geq 0}^x$ con $v \in r$ allora $\exists t \in R_{\geq 0}^x$ con $t \in r'$ $V \neq R$

3. $\forall r, r' \in R$, $\forall Y \subseteq X$, se $\underbrace{R_{[Y \leftarrow 0]} \cap r'}_{\rightarrow \text{UNA REGIONE OTTENUTA DA } R \text{ RELATIVO A } \text{causes IN } Y \text{ DI }} + \emptyset \Rightarrow R_{[Y \leftarrow 0]} \subseteq r$

\rightarrow UNA REGIONE OTTENUTA DA R RELATIVO A causes IN $Y \subseteq X$

R DEFINISCE quindi una **RELAZIONE DI EQUIVALENZA** SULLE VALUTAZIONI dove una classe DI EQUIVALENZA È DEFINITA **REGIONE**.

SE DUE VALUTAZIONI SONO EQUIVALENTI, ALLORA I LORO COMPORTAMENTI FUTURI SONO EQUIVALENTI:

1. DUE VALUTAZIONI EQUIVALENTI SUODISSENO (O STESSI) VINCI DI CLOCK

• LO SCORRERE DEL TEMPO HA DISTINZIONE DUE VALUTAZIONI EQUIVALENTI

3. RESETTING IL CLOCK HA DISTINZIONE DUE VALUTAZIONI EQUIVALENTI

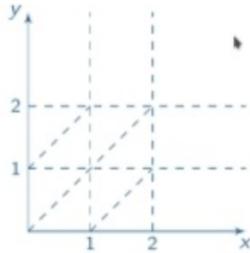
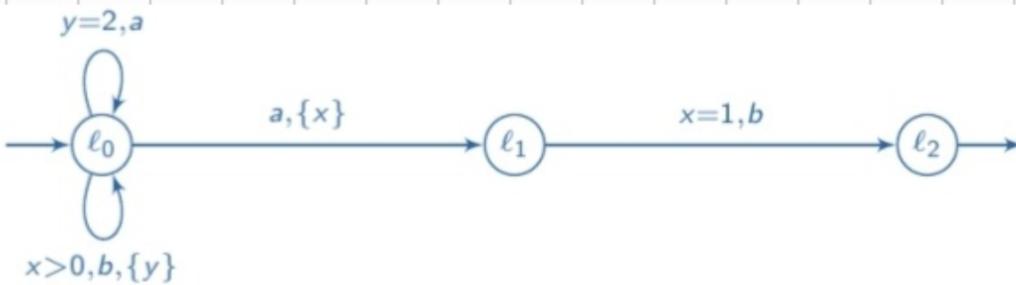
DUE VALUTAZIONI V E V' SONO EQUIVALENTI SE $\forall x, y \in X$:

. $V(x) > M \Leftrightarrow V'(x) > M$ DOVE M È UNA COSTANTE MASSIMALE PER TUTTI I CLOCCHI

. $V(x) \leq M \Rightarrow (V(x)) \sqcap \left(\{V(x)\} = 0 \Leftrightarrow \{V'(x)\} = 0 \right)$
PUNTE FRAZIONARIE

. $V(x) \leq M \wedge V(y) \leq M \Rightarrow (\{V(x)\} \leq \{V(y)\}) \Leftrightarrow \{V'(x)\} \leq \{V'(y)\})$

ESEMPIO:



LE FORME POSSIBILI DI REGIONI VINCOLATE CON 2 CLOCCHI SONO
VINCOLI DISCONSIETTI SONO:

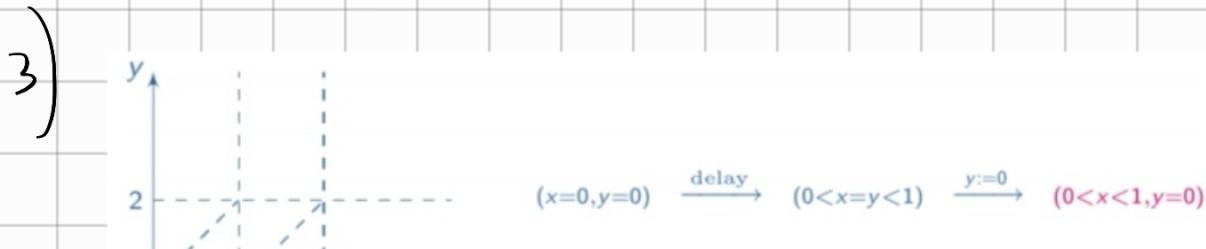
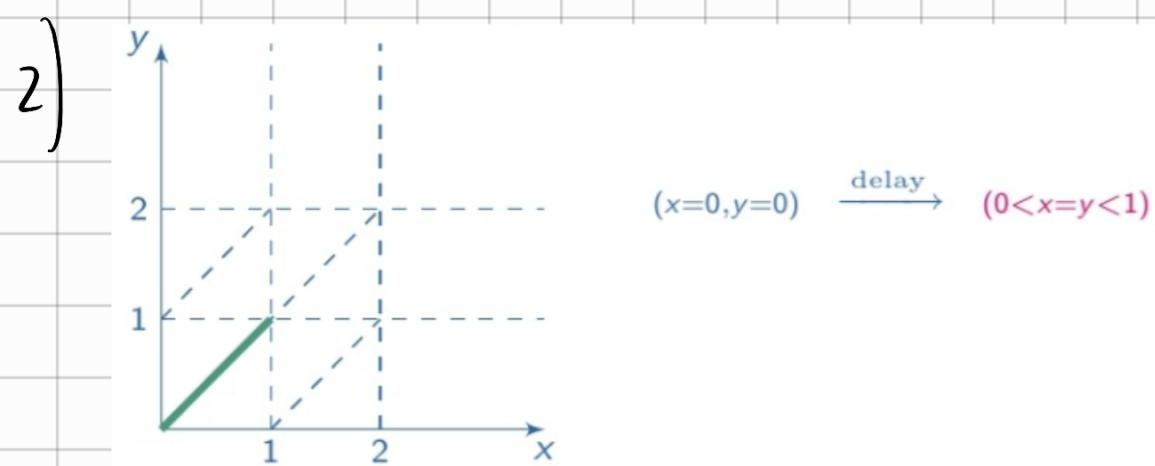
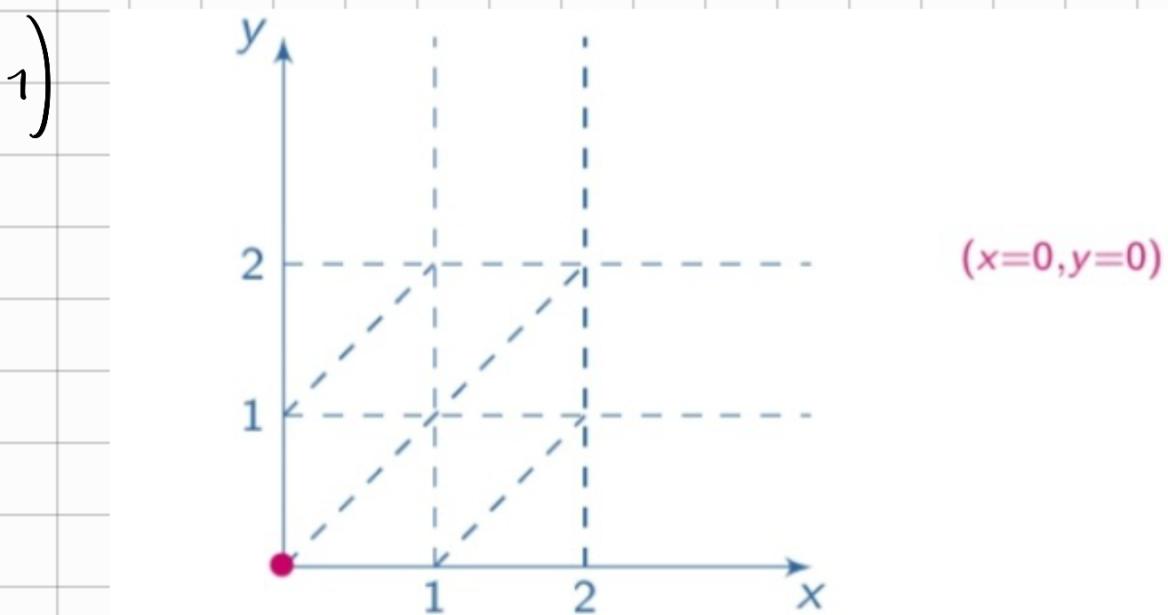


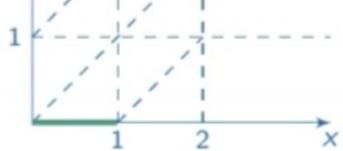
COSTRUIMOS IL REGION GRAPH, IL QUALE È UN AUTOMATO FINITO CON
STATI SULLE REGIONI E LE TRANSIZIONI SONO:

$R \xrightarrow{r} R'$ se $\exists v \in R, v' \in R', r \in \mathbb{N}_{>0}$ tali che $v' = v + r$

$R \xrightarrow{y} R'$ se $R_{[y=0]} \subseteq R'$

IL QUALE RAPPRESENTA LE POSSIBILI SCELTE DEL TEMPO DI UN SISTEMA:

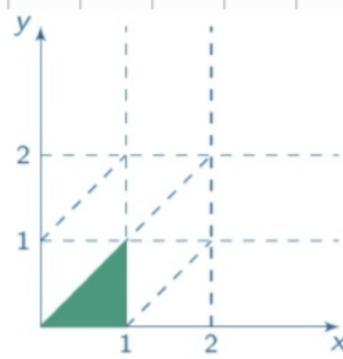




$$(x=0, y=0) \xrightarrow{\text{delay}} (0 < x = y < 1) \xrightarrow{y:=0} (0 < x < 1, y=0)$$

$$\xrightarrow{\text{delay}} (0 < y < x < 1)$$

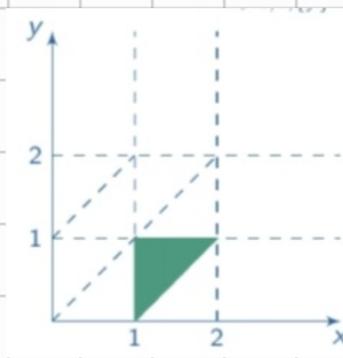
4)



$$(x=0, y=0) \xrightarrow{\text{delay}} (0 < x = y < 1) \xrightarrow{y:=0} (0 < x < 1, y=0)$$

$$\xrightarrow{\text{delay}} (0 < y < x < 1)$$

5)

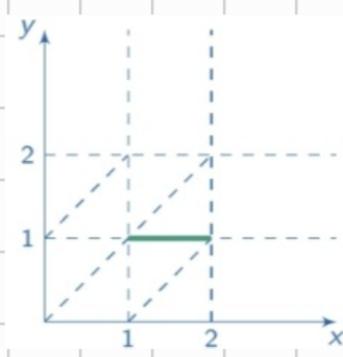


$$(x=0, y=0) \xrightarrow{\text{delay}} (0 < x = y < 1) \xrightarrow{y:=0} (0 < x < 1, y=0)$$

$$\xrightarrow{\text{delay}} (0 < y < x < 1) \xrightarrow{\text{delay}} (0 < y < 1 = x) \xrightarrow{\text{delay}} (1 < x <$$

$$2, 0 < y < 1, \{x\} < \{y\})$$

6)



$$(x=0, y=0) \xrightarrow{\text{delay}} (0 < x = y < 1) \xrightarrow{y:=0} (0 < x < 1, y=0)$$

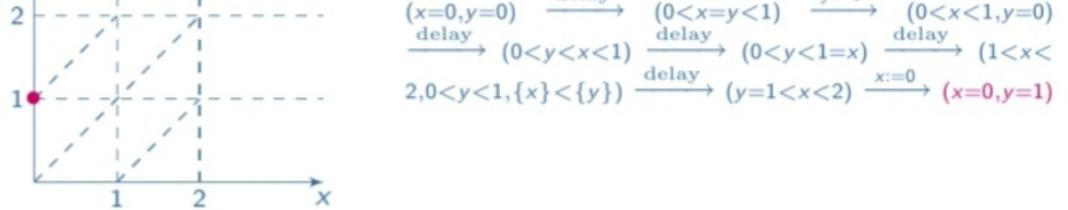
$$\xrightarrow{\text{delay}} (0 < y < x < 1) \xrightarrow{\text{delay}} (0 < y < 1 = x) \xrightarrow{\text{delay}} (1 < x <$$

$$2, 0 < y < 1, \{x\} < \{y\}) \xrightarrow{\text{delay}} (y=1 < x < 2)$$

7)



delay $y:=0$

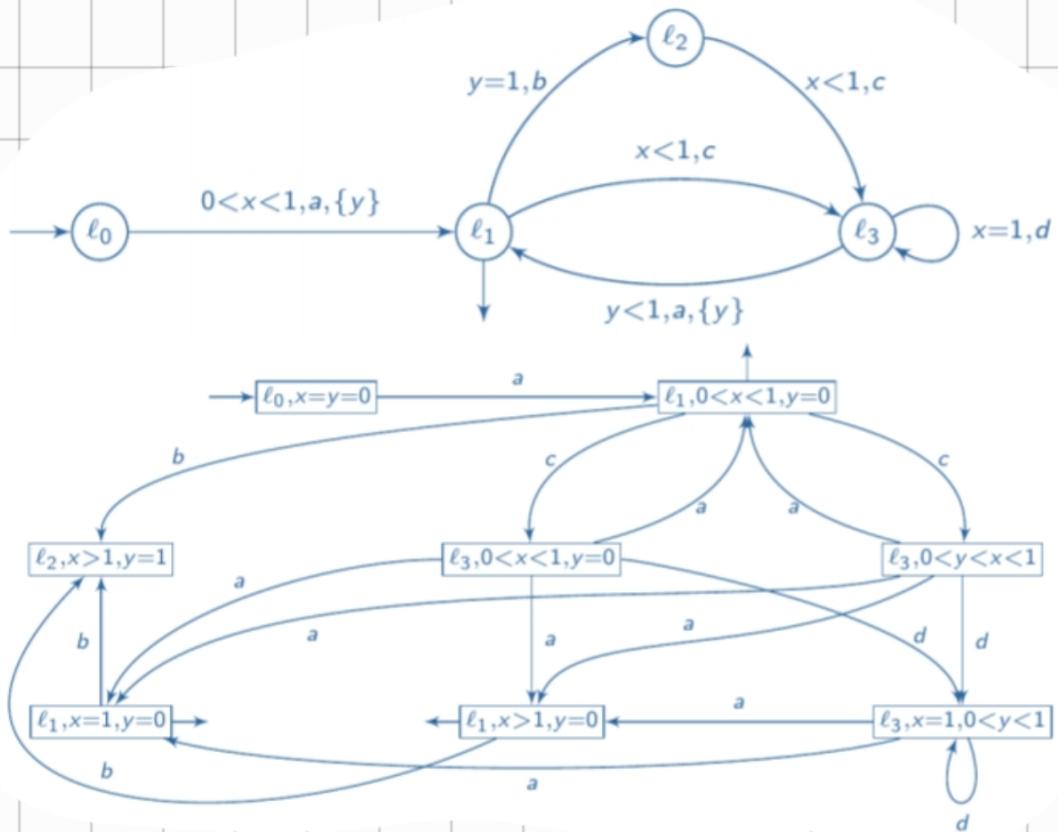


REGION AUTOMATION

È UN AUTOMATO FINITO DOVE L'INSIEME DEGLI STATI È $L \times R$ E LE TRANSIZIONI:

- $(L, R) \xrightarrow{a} (L', R')$ SE $\exists L \xrightarrow{a} L'$ è una transizione nel TA con $R \subseteq [y]$ e $R \xrightarrow{y} R'$ è una transizione nel REGION GRAPH
- $(L, R) \xrightarrow{r} (L', R')$ SE $R \xrightarrow{r} R'$ è una transizione del REGION GRAPH

ESEMPPIO:



ZONE GRAPH

SUL CIE IL NUMERO DI REGIONI È ESPONENZIALE NEL NUMERO DI CLOCK
 E IL VALORE DI COSTANTE MASSIMA NELLE GUARDE
 LO ZONE GRAPH FORNISCE UNA CONFILO PIÙ EFFICIENTE. È UN DBM CHE
 RAPPRESENTA L'UNIONE DELLE REGIONI:

An example: a TA and its zone graph

