Verificación Automática de Escenarios Condicionales

Daniel Monteverde

Director: Víctor Braberman

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

25 de junio de 2007

Resumen

El lenguaje VTS[ABKO04], es una notación visual para la definición y verificación de requerimientos complejos basado *patrones de eventos* que se utilizan para la especificación de *patrones de mal comportamiento*. De esta manera se describen aquellos escenarios genéricos, denominados *escenarios VTS existenciales*, que violan los requerimientos del modelo analizado. Una herramienta permite editar visualmente un escenario VTS existencial y transformarlo en un autómata temporizado observador para realizar la verificación con un *modelchecker* para sistemas temporizados como Kronos y Uppaal.

Con una extensión al lenguaje VTS se construyen *escenarios VTS condicionales*[BKO05a] brindando un mecanismo novedoso y eficaz para definir *triggered scenarios* y relacionar antecedentes con consecuentes. Los escenarios condicionales permiten expresar que si una ejecución del sistema cumple un sub-escenario (el antecedente), la ejecución también debe cumplir alguno de los sub-escenarios consecuentes, logrando describir propiedades complejas no expresables con otras notaciones (p.ej., [UKM02, HM02, AEN99, SC02]).

En este trabajo desarrollamos el mecanismo de verificación de los escenarios VTS condicionales mediante la construcción de escenarios negativos, expresados con escenarios VTS existenciales, que describen todos los casos en los cuales el escenario condicional podría no cumplirse. El resultado del trabajo consiste en la formalización de un conjunto de reglas para la generación de los antiescenarios: definición de las reglas, el algoritmo que las implementa y las pruebas formales que establecen la corrección de reglas. También se extiende la herramienta visual para soportar escenarios condicionales, y se construye una herramienta, desarrollada en Java, que implementa el algoritmo como componente “back-end”.

Contenido

Capítulo 1

Introducción

Las técnicas de verificación tienen como propósito determinar si un sistema de software o de hardware se comporta según su especificación. Si bien las técnicas basadas en simulación y pruebas (*testing*) son actualmente las más difundidas, éstas analizan una parte relativamente pequeña del comportamiento del sistema y son inadecuadas cuando el número de estados posibles del sistema es muy grande, como sucede en los sistemas concurrentes. En estos sistemas el comportamiento no determinístico introducido por la concurrencia puede conducir, en diferentes momentos, a distintos comportamientos frente a un mismo estímulo. En el caso de los sistemas de tiempo real, la correctitud del sistema no depende sólo del resultado lógico del cómputo sino también del cumplimiento de ciertos requerimientos temporales. Por el otro lado, existen métodos de verificación formal mediante los cuales se "demuestra" formalmente que el comportamiento del sistema cumple cierta propiedad. Una de las principales características de estos métodos es que trabajan sobre un modelo del sistema, en el cual se representan las propiedades del sistema relacionadas con el comportamiento a verificar. Dentro de estas técnicas formales se identifican las técnicas basadas en métodos algorítmicos, también denominadas en forma genérica "model-checking".

1.1 Verificación con técnicas de Model-checking

Las técnicas de model-checking son empleadas para resolver el problema de verificación formal que puede expresarse como: dado un sistema $\mathcal{S}$, un requerimiento $\mathcal{R}$, y una relación de satisfacción $\models$ vale que $\mathcal{S} \models \mathcal{R}$? En caso que el sistema no satisfaga el requerimiento la mayoría de estas técnicas devuelven un "contraejemplo". La *Figura 1.1* presenta un esquema general de la verificación mediante estas técnicas.

Lo que diferencia a las distintas técnicas entre si es la forma en que se expresan $\mathcal{S}$, $\mathcal{R}$ y la estrategia para determinar si $\mathcal{S} \models \mathcal{R}$. En particular, las técnicas basadas en la teoría de autómatas temporizados constituyen las más usadas para la modelización y verificación formal de sistemas concurrentes de tiempo real.

1.2 Técnicas basadas en autómatas temporizados

En la verificación basada en autómatas temporizados[AD94] el sistema está modelado por un autómata temporizado[1] y los requerimientos se expresan en la lógica temporizada TCTL[2]. La semántica del autómata se expresa en función de un *Sistema de transiciones etiquetadas* (STE), que consiste en un grafo decorado

[1] En general, resultado de la composición paralela de autómatas de menor tamaño llamados componentes o módulos.
[2] La lógica TCTL es un lenguaje adecuado y ampliamente usado para especificar propiedades (requerimientos) temporales.

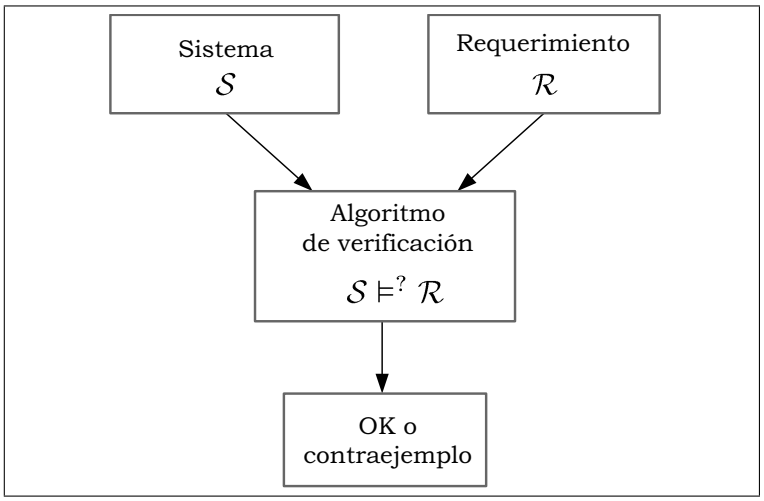


Figura 1.1: Esquema de general de verificación con Model-checking

donde los nodos son los estados del sistema (potencialmente no enumerables) y los ejes corresponden a la ocurrencia de un evento o al paso de una determinada cantidad de tiempo. Sobre este grafo se interpretan las fórmulas TCTL para determinar si el sistema cumple o no el requerimiento.

Dentro de este grupo de técnicas existe una variante donde se utilizan autómatas *observadores* para describir los requerimientos del sistema. En este contexto, se utilizan autómatas que capturen la negación del requerimiento (es decir, todos los comportamientos que violan el requerimiento) y una fórmula TCTL (en general, mucho más simple que las usadas en el enfoque clásico), que expresa la alcanzabilidad de un conjunto de estados considerados erróneos. El problema de verificación se traduce, entonces, en un problema de alcanzabilidad sobre el STE generado por la composición del modelo de sistema y el autómata observador (*Figura 1.2*). Esta variante, es la base para la estrategia de verificación del lenguaje VTS[ABKO04].

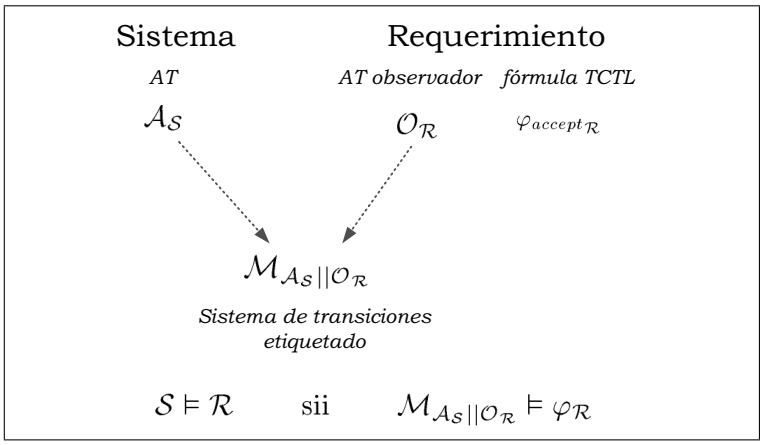


Figura 1.2: Verificación basada en autómatas temporizados *observador*

1.3 Lenguaje VTS

El lenguaje VTS *(Visual Timed event Scenarios)* es una notación visual para definir requerimientos de tiempo real basada en *escenarios* (también denominados *patrones de eventos*). Un escenario es básicamente la definición de un orden parcial de eventos relevantes que puede ser visto como un *patrón*, en el sentido que representa un conjunto de ejecuciones del sistema. Mediante este formalismo se logra expresar de forma

simple y abstracta las dependencias de causalidad entre los eventos del sistema, permitiendo también expresar restricciones temporales explícitas.

Los escenarios se interpretan en función de una semántica existencial. Concretamente, VTS es empleado para responder al tipo de preguntas de la forma "hay una ejecución posible del sistema que corresponda a este escenario genérico?". En los casos donde VTS se utiliza para la verificación de violaciones de requerimientos, el escenario representa la negación o el complemento de los requerimientos. En este sentido, la semántica de VTS esta fuertemente relacionado al concepto de *escenario negativo* o de *mal comportamiento*.

La *Figura 1.3* representa el mecanismo de verificación para el lenguaje VTS. Dado un escenario E que modela la propiedad a verificar, interpretado como un escenario de *mal comportamiento*, una herramienta del lenguaje transforma el escenario $E_{\neg\mathcal{R}}$ en un *observador* (modelado como un autómata temporizado). El *Tableau* [3] definido por el autómata $\mathcal{A}_{E_{\neg\mathcal{R}}}$ y la condición de aceptación φ_{accept} (expresada como una fórmula TCTL), junto con el autómata del modelo del sistema $A_{\mathcal{S}}$, alimentan un *modelchecker* (un programa) para sistemas temporizados como Kronos[BDM+98] y Uppaal[BLL+95]. El *modelchecker* analiza todas las ejecuciones posibles del sistema y determina si todas ellas cumplen la condición de aceptación.

De esta forma, dado el sistema $\mathcal{S}$ y un escenario E (que viola un requerimiento $\mathcal{R}$) se determina que $\mathcal{S} \models E$ si y sólo si existe una ejecución de $\mathcal{S}$ que satisface a E. Por lo tanto, $\mathcal{S} \models \mathcal{R}$ si y sólo si $\mathcal{S} \not\models E_{\neg\mathcal{R}}$. Es decir, si se verifica que no existe ninguna ejecución que viole el requerimiento.

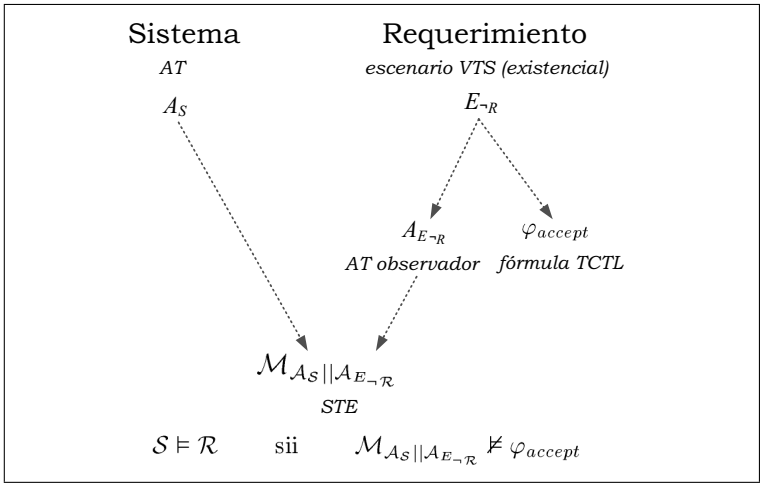


Figura 1.3: Verificación de requerimientos con VTS

1.4 Objetivo del trabajo y contribuciones

En este trabajo se desarrolla de manera práctica una extensión del lenguaje VTS para representar requerimientos mediante "escenarios condicionales"[BKO05a]. Los escenarios condicionales permiten expresar que si una ejecución del sistema cumple un sub-escenario (el antecedente), la ejecución también debe cumplir alguno de los sub-escenarios consecuentes, logrando describir propiedades complejas no expresables con otras notaciones.
La estrategia de verificación de los escenarios condicionales consiste en la generación de escenarios negativos que se formulan utilizando escenarios VTS existenciales. Estos antiescenarios describen todos los casos genéricos en los cuales el escenario condicional podría no cumplirse. Si ninguno de estos antiescenarios es

[3] En este trabajo *tableau* se refiere al algoritmo que transforma el problema de VTS a un problema de autómatas.

satisfecho entonces se comprueba que el requerimiento expresado como un escenario condicional se satisface. La *Figura 1.4* representa el mecanismo de verificación para esta extensión del lenguaje.

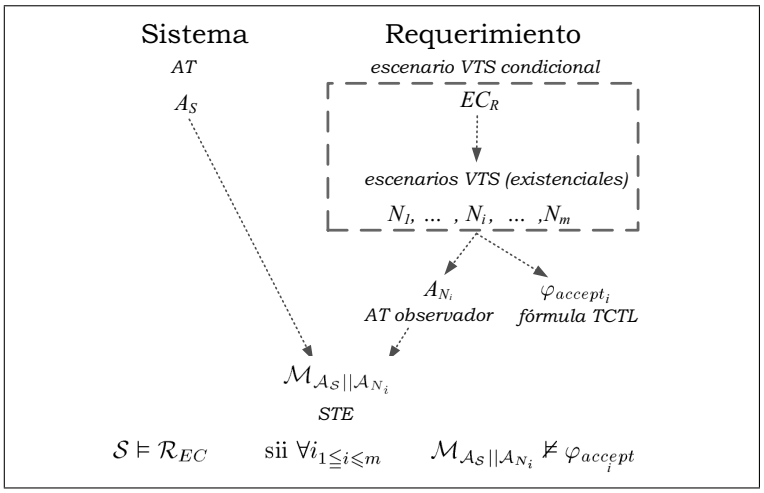


Figura 1.4: Verificación de requerimientos de escenarios condicionales

En la *Figura 1.5* se representa el proceso de verificación. Inicialmente, el diseñador especifica el requerimiento por medio un escenario condicional (EC) en un *front-end*. Luego, utiliza el *verificador* con este EC y el modelo del sistema para *generar* los antiescenarios que se *traducen* en autómatas temporizados y posteriormente efectuar el *modelchecking* de todos estos en un *modelchecker*, como Kronos. Finalmente, si y sólo si ninguno de los antiescenarios se satisface el diseñador determina que el requerimiento, expresado como un EC, se satisface. En caso contrario el *modelchecker* revela un contraejemplo que le describe al diseñador cómo deberían ocurrir las cosas para que el sistema viole al requerimiento.

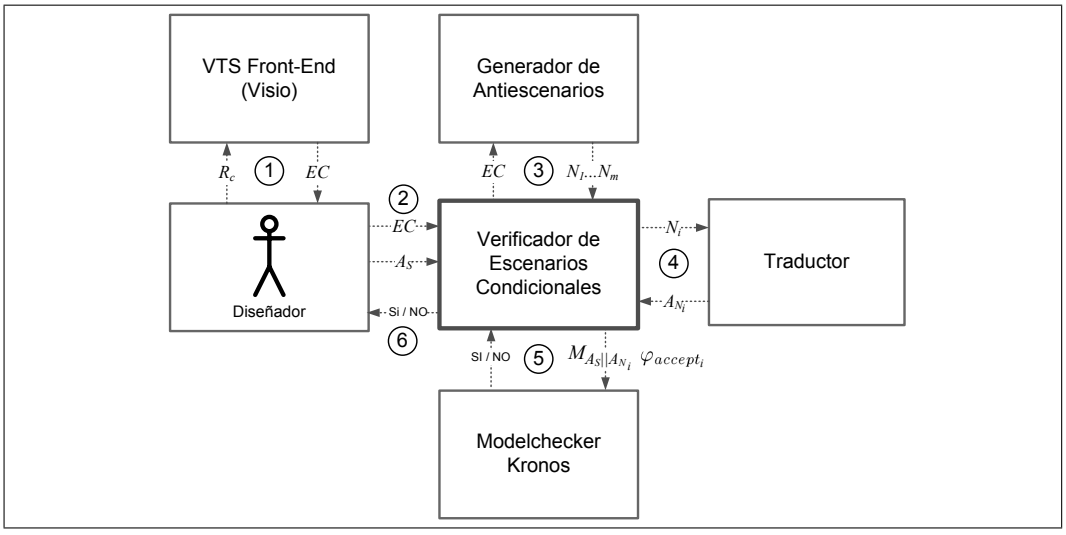


Figura 1.5: Proceso para la verificación de escenarios condicionales

Este trabajo consiste en la formalización de un conjunto de reglas para la generación de los antiescenarios: definición de las reglas, el algoritmo que las implementa y las pruebas formales que establecen la corrección de reglas; proporcionando ejemplos aplicados a casos de estudio. También se extiende la herramienta visual para soportar escenarios condicionales, y se construye una herramienta, desarrollada en Java, que soporta el proceso de verificación de escenarios condicionales que se presenta en la *Figura 1.5*.

1.5 Estructura del trabajo

El capítulo 2 introduce el lenguaje de los escenarios VTS existenciales que se utilizará en el resto del trabajo. Se describen los conceptos y las definiciones respecto a la sintaxis y la semántica, se presentan ejemplos y se detalla la técnica de verificación de los escenarios VTS.

En el capítulo 3 se define la extensión del VTS para soportar los escenarios condicionales. Se presenta la notación visual, la sintaxis y la semántica asociada a los escenarios condicionales.

En el capítulo 4 se desarrolla la técnica de verificación de escenarios condicionales. Se introduce primero la idea del algoritmo mediante ejemplos parciales, y luego se definen el algoritmo y las reglas para la construcción de los antiescenarios.

En el capítulo 5 se presenta un conjunto de optimizaciones relacionadas con el algoritmo.

En el capítulo 6 se incluyen diversos casos de estudio para ilustrar la utilización del algoritmo.

El capítulo 7 describe la implementación de la herramienta construida para la verificación de escenarios condicionales.

El capítulo 8 presenta las conclusiones, los trabajos relacionados, y las actividades de investigación futuras relacionadas con este trabajo.

Finalmente, en el Apéndice A se desarrolla el aspecto formal de la técnica de verificación de los escenarios condicionales y en el en el Apéndice B se incluye la definición del Tableau para el modelchecking de escenarios VTS existenciales.

Capítulo 2

Escenarios Existenciales

En este capítulo se presentan los conceptos y definiciones del lenguaje VTS[ABKO04]. Se describe la notación gráfica acompañada de ejemplos, para luego formalizar la sintaxis, la semántica y la técnica de verificación para los escenarios VTS. En el próximo capítulo estos conceptos se utilizarán en relación a la definición de los escenarios condicionales.

2.1 Introducción

Los elementos básicos de la representación gráfica de VTS son puntos conectados por líneas y flechas. Los puntos son etiquetados por un conjunto de eventos (no vacío), para indicar que el punto corresponde a la ocurrencia de uno de estos eventos durante la ejecución del sistema. Una flecha entre dos puntos indica *precedencia* del punto inicio respecto al punto destino. El conjunto de eventos que etiquetan a las fechas corresponden a *eventos prohibidos* entre ambos puntos. A continuación se introducen distintos escenarios para ejemplificar estos conceptos.

La *Figura 2.1* muestra un patrón VTS que expresa un predicado que es verdadero sobre una ejecución dada únicamente si ésta contiene un *stimulus* e [1] seguido de dos respuestas (es decir, los siguientes eventos *r1* y *r2*), entre las cuales no ocurre otra respuesta *r3*. Mediante triángulos en la parte inferior de los puntos se asignan a estos un nombre opcional.

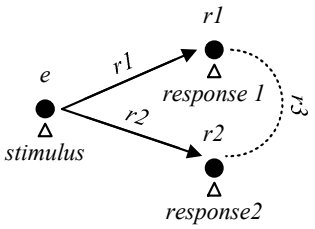


Figura 2.1: Respuestas separadas

Las flechas de *stimulus* a *response1* y *response2* establecen que e ocurre antes que las respuestas *r1* y *r2*, sin importar el orden relativo entre estas. Para indicar que el punto *response1* (*response2* respectivamente) representa la primera ocurrencia de sus eventos después de e (es decir no hay otra *r1* (*r2*) entre e y *r1* (*r2*)), la flecha *stimulus-response1* (*-response2)* se etiqueta con *r1* (*r2*). Mediante la línea punteada etiquetada con *r3* que relaciona los puntos *response1* y *response2*, se expresa la condición "... los cuales no

[1] En este trabajo, los nombres para los puntos y eventos de los escenarios se definen en inglés.

son separados por otra respuesta *r3*". En este caso se utiliza una línea en lugar de una flecha dado que no hay precedencia entre *response1* y *response2* (su orden relativo no está determinado).

Ahora, veamos cuando una ejecución tiene *matching* con el escenario de la *Figura 2.1.* Suponiendo que se tienen las siguientes secuencias de eventos:

s1: ... *a, e, b, c, r1, d, r3, r2, z...*,

s2: ... *a, e, b, c, r1, d, r2, f, r3, z...*,

s3: ... *a, e, b, r2, r1, c, r3, r2, z....*

Las secuencias *s2* y *s3* tienen matching con el escenario, porque para los primeros eventos *r1* y *r2* después del único evento *e* no hay un evento *r3* entre ellos. Entonces si este escenario fuese interpretado como un escenario *negativo*, se descartarían los sistemas que generan *s2* y *s3* por no cumplir el requerimiento. Al contrario, la secuencia *s1* no tiene matching con el escenario, y por lo tanto satisface el requerimiento.

Ahora se introduce un caso que incluye restricciones temporales. El escenario de la *Figura 2.2* presenta la situación donde un *stimulus e* no es seguido por una respuesta *r1* o *r2* dentro de 100 unidades de tiempo (u.t.), incumpliendo la propiedad de respuesta acotada. El punto *stimulus* representa un instante posterior a que ocurra el *stimulus e.* El punto no etiquetado [2] representa un instante posterior a que ocurra el *stimulus e.* La restricción > 100 en la parte inferior de la flecha indica que la distancia temporal entre ambos es mayor que 100 u.t. Se expresa la ausencia de los eventos *r1* y *r2* entre el *stimulus* y el *instant* etiquetando la flecha con *r1, r2.*

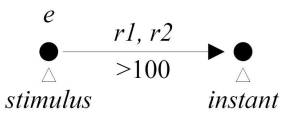


Figura 2.2: Respuesta acotada

Cuando los escenarios incluyen restricciones temporales, estos deben tener matching con secuencias temporizadas. Tomando las secuencias anteriores, pero agregándoles marcas de tiempo o *timestamp* a cada evento (por simplicidad, en este caso se utilizan números naturales como timestamp, pero VTS admite cualquier número real no negativo):

$$s4 : \ldots \begin{array}{cccccccccc} a & e & b & c & r1 & d & r3 & r2 & z & \\ 12 & 15 & 39 & 50 & 72 & 123 & 140 & 148 & 155 & \ldots, \end{array}$$

$$s5 : \ldots \begin{array}{cccccccccc} a & e & b & c & r1 & d & r2 & f & r3 & z \\ 3 & 7 & 12 & 88 & 109 & 111 & 114 & 121 & 125 & 152 \end{array} \ldots,$$

$$s6 : \ldots \begin{array}{cccccccccc} a & e & b & r2 & r1 & c & r3 & r2 & z & \\ 5 & 7 & 69 & 78 & 87 & 100 & 146 & 152 & 199 & \ldots \end{array}$$

La secuencia *s5* tiene matching con el escenario *negativo* de la *Figura 2.2*, dado que ningún evento *r1* o *r2* aparece después de *e* en 100 u.t., por lo que no cumple el requerimiento de respuesta acotada. Por el otro lado, las secuencias *s4* y *s6* no tienen matching con el escenario (negativo), por lo que satisfacen el requerimiento.

Ahora tenemos los elementos para describir un requerimiento más complejo: una respuesta correlacionada. La *Figura 2.3* presenta un patrón VTS para detectar cuando se incumple el requerimiento donde un *stimulus e* es seguido de dos respuestas (es decir los próximos eventos *r1, r2*) entre las cuales la distancia

[2] En verdad, todo punto sin eventos de la notación gráfica tiene asociado formalmente el evento distinguido λ.

temporal es de al menos 20 u.t. y a lo sumo 100 u.t. Como en la *Figura 2.1*, las flechas indican el orden relativo entre *e*, *r1* y *r2*.

En este ejemplo se introduce una abreviatura para un caso frecuente en relación a las restricciones de eventos: un punto representa la *próxima* ocurrencia de un evento después de otro. En este caso la abreviatura es una segunda flecha (abierta) cercana al extremo final (en el texto $\rightarrow\!\!\rightarrow$); es equivalente a agregar *r1* como un evento prohibido en la flecha (como se presenta en la *Figura 2.1*). Asimismo, si se requiere expresar que después del evento *e* no ocurre otro evento *e* hasta que suceda *r1* (es decir para identificar la ocurrencia *previa* del evento *e* a *r1*) se puede usar una notación simétrica: una flecha abierta cercana al extremo inicial de la flecha (en el texto $\leftarrow\!\!\rightarrow$). La restricción temporal es expresada como ¬[20,100] en la línea punteada entre ambas respuestas.

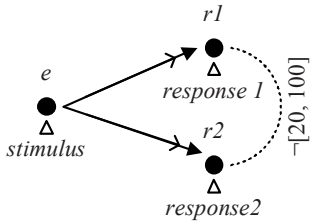


Figura 2.3: Respuestas correlacionadas

Volviendo a las secuencias *s4, s5 y s6*, se puede ver que *s4* no tiene matching con el escenario de la *Figura 2.3*, porque la distancia temporal entre los eventos *r1* y *r2* (148 - 72 = 76 u.t.) no está dentro del periodo temporal ¬[20,100]. Lo opuesto ocurre con *s5* y *s6*, donde las distancias entre las respuestas corresponden a 114 - 109 = 5 u.t. y 87 - 78 = 9 u.t. respectivamente.

Otra característica que permite representar el lenguaje es que algo sucede (o no) desde el *comienzo* o hasta el *final* de una ejecución. Para estas situaciones, *VTS* tiene dos símbolos especiales: un círculo grande lleno para el comienzo, y dos círculos concéntricos para el final. Por ejemplo, en el escenario de la *Figura 2.4* se define que la distancia temporal entre la primera *a* y la última *b* (ambas respecto a toda la ejecución) esta limitada a 100 u.t.

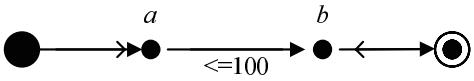


Figura 2.4: Símbolos de comienzo y fin

En *VTS* también se pueden identificar el *primero* y el *último* respecto a un conjunto de eventos. En la notación gráfica, el primer evento en un conjunto se representa por un punto que se relaciona a cada punto del conjunto por líneas punteadas que finalizan con un círculo vacío. Una notación similar es utilizada para representar el último evento, pero usando un círculo lleno. La representación combinada de relaciones de primeros y últimos también se pueden expresar, tal como se puede ver en la *Figura 2.5*. Parte de este escenario modela la situación donde, dado dos conjuntos de eventos, {*a1, a2*} y {*b1, b2, b3*}, la distancia temporal entre el último evento en cada conjunto en ocurrir es menor que 100 u.t. Este ejemplo también permite apreciar como en *VTS* se pueden expresar propiedades de gran complejidad para un sistema. Este escenario representa el caso donde un sensor de movimiento (*watchdog*) es activado (*wd-on*) al menos 50 u.t. antes que cualquiera de los eventos en ambos conjuntos ocurra y no se desactiva hasta tanto todos los eventos monitoreados sucedan. Como este es un escenario negativo, el matching es de ejecuciones en las cuales todas estas condiciones son detectadas y sin embargo el sistema falla en activar la alarma (*alarm*).

Nótese que todo punto representativo (p.ej. en la *Figura 2.5* el punto *start*) debe etiquetarse con todos los eventos de los puntos a los que representa dado que es utilizado para representar un evento de estos

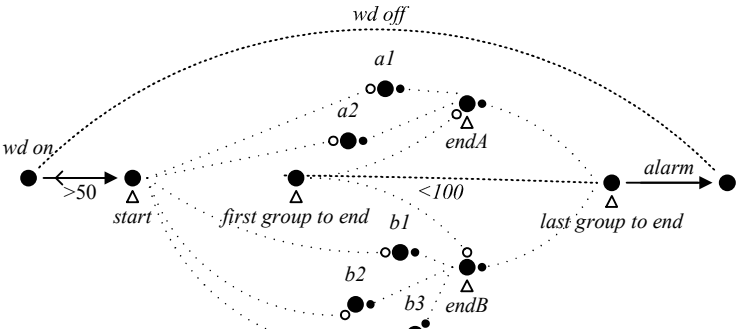


Figura 2.5: Representativos VTS

conjuntos. No obstante, en la notación gráfica, los representativos no presentan etiquetas con el objetivo de mantener la claridad del diagrama.

La siguiente tabla de la *Figura 2.6* resume en forma completa la notación gráfica para los escenarios VTS.

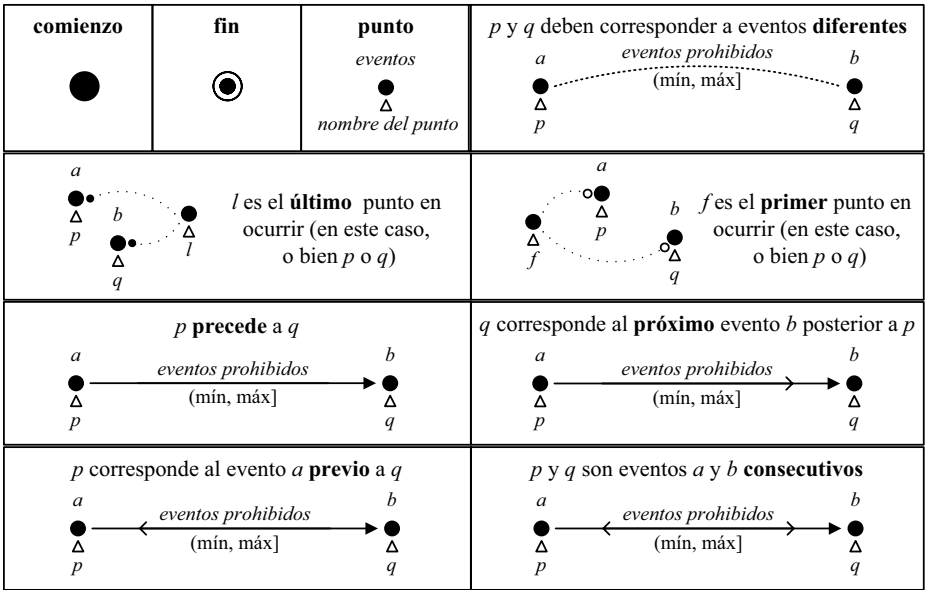


Figura 2.6: Notación gráfica VTS

2.2 Sintaxis Formal

Definición 2.2.1 (Escenario)**.** Un *escenario* es una tupla $\langle \Sigma, P, \ell, \not\equiv, <, <_F, <_L, \gamma, \delta \rangle$, donde:

- Σ es un conjunto finito de eventos.
- P es un conjunto finito de puntos.
- $\ell : P \to 2^{\Sigma \cup \{\lambda\}}$ es una función que etiqueta cada punto con un conjunto de eventos no vacío donde λ es un evento distinguido que representa un momento de la ejecución sin asignación de eventos.
- $\not\equiv \ \subseteq P \times P$ es una relación asimétrica entre puntos para representar la desigualdad entre estos.

- $< \subseteq (P \uplus \{\mathbf{0}\} \times P \uplus \{\infty\}) \setminus \{\langle \mathbf{0}, \infty \rangle\}$ es una relación de precedencia – asimétrica y no reflexiva – entre puntos ($\mathbf{0}$ y ∞ representan el comienzo y el final de la ejecución, respectivamente).
- $<_F \subseteq P \times P$ asocia a un punto para representar el primero de un conjunto en ocurrir.
- $<_L \subseteq P \times P$ asocia cada punto en un conjunto a otro punto que representa el último en ocurrir.
- $\gamma : (\not\equiv \cup <) \to 2^{\Sigma}$ asigna a cada par de puntos, relacionados por una relación de desigualdad o de precedencia, el conjunto de eventos prohibidos entre ellos.
- $\delta : (\not\equiv \cup < \setminus (P \uplus \{\infty\})) \to \Phi$ (que se define a continuación) asigna para cada relación de desigualdad o de precedencia una restricción respecto del tiempo transcurrido entre ambos puntos.

Se define $\mathfrak{I}_{\mathbf{N}}$ al conjunto de intervalos de números reales positivo determinados por extremos enteros (se admite el símbolo distinguido ∞ para definir que el intervalo no tiene cota superior). Una *restricción temporal* es una fórmula de la forma θ o $\neg\theta$, donde θ es un intervalo en $\mathfrak{I}_{\mathbf{N}}$. Φ es el conjunto de todas las restricciones de tiempo. Dado un número real no negativo t y un intervalo θ, se establece que $t \vDash \theta$ sii $t \in \theta$ y $t \vDash \neg\theta$ sii $t \notin \theta$.

Ejemplo. La sintaxis formal para el escenario de la *Figura 2.3* se define por la tupla: $\langle \Sigma, P, \ell, \not\equiv, <, <_F, <_L, \gamma, \delta \rangle$ donde:

$$\Sigma = \{e, r1, r2\},$$
$$P = \{stimulus, response1, response2\},$$
$$\ell = \{stimulus \to \{e\}, response1 \to \{r1\}, response2 \to \{r2\}\},$$
$$\not\equiv = \{(response1, response2)\},$$
$$< = \{(stimulus, response1), (stimulus, response2)\},$$
$$<_F = \emptyset,$$
$$<_L = \emptyset,$$
$$\gamma = \{(stimulus, response1) \to \{r2\}, (stimulus, response2) \to \{r2\}, (response1, response2) \to \emptyset\},$$
$$\delta = \{(stimulus, response1) \to \{[0, \infty)\}, (stimulus, response2) \to \{[0, \infty)\}, (response1, response2) \to \neg[20, 100]\}$$

2.3 Semántica

Como vimos anteriormente, la semántica de VTS se define utilizando el concepto de matching en función de una ejecución. A continuación se formalizan las nociones de *secuencia*, *secuencia temporal* y *ejecución*.

Definición 2.3.1 (Secuencia)**.** Dado un conjunto C, una *secuencia* sobre C es una sucesión (posiblemente infinita) de elementos de C. Dada una secuencia s:

- $|s|$ es su longitud (se establece que $|s| \stackrel{def}{=} \infty$ cuando s es infinita).
- $\Pi(s) \stackrel{def}{=} \{i \in \mathbf{N} \ / \ 0 \leq i < |s|\}$ es el conjunto de posiciones de s.
- Dados $i, j \in \Pi(s)$, s_i denotará el i^{esimo} elemento de s.
- $s_{i]}$ es el prefijo de s que termina con el i^{esimo} elemento inclusive.
  $s_{[i}$ es el sufijo que comienza en la posición i.
  y $s_{[i,j]}$ es la subsecuencia de la posición i a la posición j (si $i > j$, $s_{[i,j]} \stackrel{def}{=} s_{[j,i]}$).
  Usando '(' o ')' en lugar de '[' o ']' la subsecuencia correspondiente no incluye sus extremos.

- Se define $first(s)$ al primer elemento de s. Si s es finita, $last(s)$ es su último elemento.

Definición 2.3.2 (Secuencia Temporal)**.** Una *secuencia temporal* es una secuencia incremental de *timestamps* (es decir de números reales no negativos).
Dada una secuencia temporal finita τ se define el *tiempo transcurrido* durante τ como
$\Delta(\tau) = last(\tau) - first(\tau)$ o 0 si $|\tau| = 0$.
Una secuencia temporal τ puede ser *desplazada* por un número real ϵ produciendo una secuencia temporal denominada $\tau + \epsilon$, tal que $\forall i \in \Pi(\tau); (\tau + \epsilon)_i = \tau_i + \epsilon$.
Se define la *concatenación temporal* de dos secuencias temporales τ y τ' como $\tau \triangleright \tau' = \tau(\tau' + last(\tau))$. Por ejemplo $(0\ 2\ 3\ 5.5) \triangleright (1\ 5.3) = (0\ 2\ 3\ 5.5\ 6.5\ 10.8)$.

Definición 2.3.3 (Ejecución)**.** Una *ejecución* sobre un conjunto C es un par $\langle s, \tau \rangle$ donde s es una secuencia sobre $C \cup \{\lambda\}$ y τ es una secuencia temporal de la misma longitud, donde λ representa un instante en la ejecución sin asignación de eventos.

La semántica de VTS está dada por la asignación a cada escenario de un conjunto de ejecuciones que lo satisfacen. Los puntos representan eventos en las ejecuciones, estos corresponden a una posición particular de la ejecución si el evento en dicha posición aparece entre los eventos asociados al punto por la función ℓ de etiquetado. Los puntos etiquetados con el evento distinguido λ se denominan *instantes*, que representan momentos en la ejecución donde no ocurren eventos.

Se define el conjunto de *representativos-primeros* (*-últimos*) como $FirstReps = Dom(<_F)$ (resp. $LastReps = Ran(<_L)$). Un punto p en estos conjuntos es denominado *representativo*, dado que éste representa el primer (resp. último) punto del conjunto $R = \{\mathsf{p}'/\mathsf{p}' <_F \mathsf{p}\}$ (resp. $\{\mathsf{p}'/\mathsf{p} <_L \mathsf{p}'\}$) que tiene matching en una ejecución. El conjunto R para un *representativo-primero* (resp. *-último*) p se denota como $FirstOf(\mathsf{p})$ (resp. $LastOf(\mathsf{p})$). Los puntos restantes son llamados *concretos*.

Intuitivamente, un *matching* representa una forma de asociar los puntos de un escenario a posiciones de una ejecución, demostrando como la ejecución satisface el escenario.

Definición 2.3.4 (Matching)**.** Dado un escenario $\mathcal{S} = \langle \Sigma, P, \ell, \not\equiv, <, <_F, <_L, \gamma, \delta \rangle$, una ejecución $\sigma = \langle s, \tau \rangle$ sobre Σ y una función de mapping $\hat{\cdot} : P \to \Pi(\sigma)$; se dice que $\hat{\cdot}$ es un *matching* entre $\mathcal{S}$ y σ sii para todos los puntos $\mathsf{p}, \mathsf{q} \in P$:

- **M1** $s_{\hat{\mathsf{p}}} \in \ell(\mathsf{p})$;
- **M2** si $\mathsf{p} \not\equiv \mathsf{q}$ entonces $\hat{\mathsf{p}} \neq \hat{\mathsf{q}}$;
- **M3** si $\mathsf{p} < \mathsf{q}$ entonces $\hat{\mathsf{p}} < \hat{\mathsf{q}}$;
- **M4** $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{q}})} \cap \gamma(\mathsf{p}, \mathsf{q}) = \emptyset$;
- **M5** $s_{\hat{\mathsf{p}})} \cap \gamma(\mathbf{0}, \mathsf{p}) = s_{(\hat{\mathsf{p}}} \cap \gamma(\mathsf{p}, \infty) = \emptyset$;
- **M6** $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \vDash \delta(\mathsf{p}, \mathsf{q})$;
- **M7** $\Delta(\tau_{\hat{\mathsf{p}}]}) \vDash \delta(\mathbf{0}, \mathsf{p})$;
- **M8** si $\mathsf{p} \in FirstRep$ (resp. $LastRep$) entonces $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in FirstOf(\mathsf{p})\}$ (resp. max y $LastOf$).

Definición 2.3.5 (Semántica VTS Existencial)**.** Se define que una ejecución σ *satisface* un escenario $\mathcal{S}$ (denotado $\sigma \vDash \mathcal{S}$) sii existe al menos un matching entre ambos. Se define que un autómata temporal [3] $\mathcal{A}$ satisface un escenario $\mathcal{S}$ ($\mathcal{A} \vDash \mathcal{S}$) sii éste presenta al menos una ejecución divergente en el tiempo[4] que satisfaga al escenario.

2.4 Model Checking para la Semántica Existencial VTS

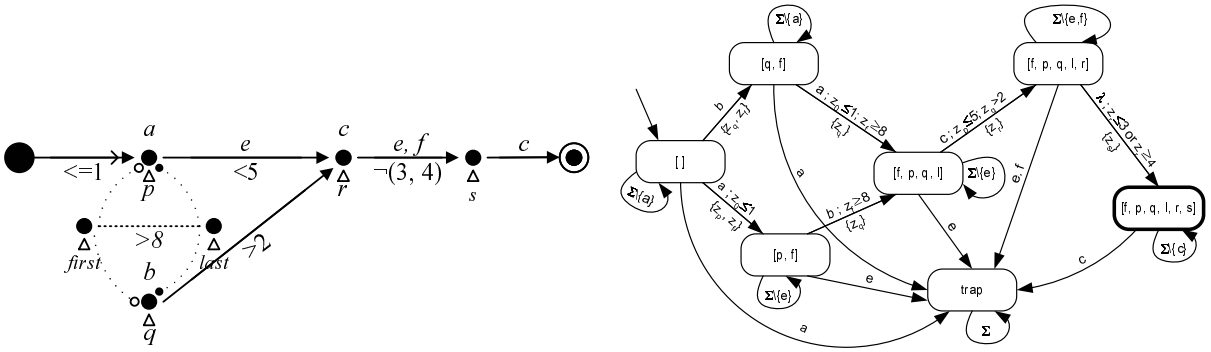
Como ya mencionamos, la verificación o *model-checking* de un escenario VTS se realiza por medio de autómatas temporizados. El mapping desde un escenario a un autómata temporizado se define por un algoritmo de *Tableau* cuya definición se encuentra en el *Apéndice B*. Este algoritmo genera un *Tableau* $\mathcal{T}_\mathcal{S}$ que reconoce todas las ejecuciones que tienen matching con el escenario $\mathcal{S}$.

Teorema 2.4.1 (Model checking VTS)**.** Dado un autómata temporizado $\mathcal{A}$ y un escenario $\mathcal{S}$, con $\Sigma \subseteq label(\mathcal{A})$, $Pr(accept) = locs(\mathcal{A}) \times \{\mathsf{s}_{\mathsf{accept}}\}$ y $Pr(init) = \{(init(\mathcal{A}), init(\mathcal{T}_\mathcal{S}))\}$:

$$\mathcal{A} \vDash \mathcal{S} \qquad \text{sii} \qquad \mathcal{A}||\mathcal{T}_\mathcal{S} \vDash init \Rightarrow \exists\Diamond(\exists\Box\, accept)$$

Dado que la proposición *accept* esta asociada a P (el conjunto completo de puntos), la satisfacción de la fórmula TCTL $init \Rightarrow \exists\Diamond\, accept$ ("*accept* es alcanzable desde *init*") podría significar que existe matching con el escenario. Sin embargo, se tiene que verificar que la ejecución puede permanecer en el nodo de aceptación sin recibir ningún evento prohibido (es decir eventos que no deberían ocurrir hasta ∞), esto se expresa como $init \Rightarrow \exists\Diamond(\exists\Box\, accept)$ ("Es posible evolucionar desde *init* a *accept* y permanecer ahí para siempre"). Este concepto resultará importante más adelante cuando hablemos de las optimizaciones relacionadas con las reglas del algoritmo de verificación de escenarios condicionales. La demostración completa de que el problema es decidible y los detalles relacionados con la construcción del tableau pueden encontrarse en [BKO05b].

En la *Figura 2.7(b)* se presenta un ejemplo con el *tableau* para el escenario VTS de la *Figura 2.7(a)*



(a) Escenario (b) Tableau

Figura 2.7: Ejemplo de un escenario VTS con su correspondiente tableau

[3] La definición de *autómata temporal* se encuentra en el *Apéndice B*.

[4] τ es una ejecución divergente en el tiempo sii para cualquier número real T entonces existe una posición k tal que $\Delta(\tau_{k]}) > T$.

Capítulo 3

Escenarios Condicionales

3.1 Conceptos generales

En esta sección se presenta una extensión al lenguaje VTS que brinda un nuevo tipo de escenarios que son interpretados como predicados condicionales[BKO05a] sobre los matchings de escenarios.

Los escenarios condicionales son útiles para expresar que si para una ejecución dada se determina un matching para un sub-escenario (el antecedente), este mismo matching debe ser extensible para cubrir alguno de los escenarios consecuentes. Este noción es más flexible que otras aproximaciones conocidas basadas en *trigger* (p.ej., [UKM02, HM02, AEN99, SC02]) dado que las extensiones no tienen que ser necesariamente eventos subsecuentes (futuros); sino que estas pueden relacionarse sobre ocurrencias de eventos previos (p.ej., "si *ack* es detectado, entonces el *request* debe haber sucedido al menos 10 u.t. antes"), y puede incluso incorporar restricciones de *patrones de eventos* al antecedente (p.ej., "si dos eventos a y b son encontrados en una ejecución, entonces o bien no puede encontrarse un evento c entre ellos, o la distancia es mayor que 5 u.t.").

A continuación se presentan las definiciones para formalizar el concepto de escenarios condicionales:

Definición 3.1.1 (Especialización de Escenarios)**.** Dados dos escenarios $\mathcal{S}_1$, $\mathcal{S}_2$, se dice que $\mathcal{S}_2$ *especializa* $\mathcal{S}_1$ (denotado $\mathcal{S}_2 <: \mathcal{S}_1$) sii

- **Sp1** $P_1 \subseteq P_2$ [1];
- **Sp2** $\Sigma_1 \subseteq \Sigma_2$;
- **Sp3** $\ell_2(\mathsf{p}) \subseteq \ell_1(\mathsf{p})$ para todo $\mathsf{p} \in P_1$;
- **Sp4** $<_1 \subseteq <_2$;
- **Sp5** $<_{F1} \subseteq <_{F2}$;
- **Sp6** $<_{L1} \subseteq <_{L2}$;
- **Sp7** $\gamma_1(\mathsf{p},\mathsf{q}) \subseteq \gamma_2(\mathsf{p},\mathsf{q})$ para todo $\mathsf{p},\mathsf{q} \in P_1$;
- **Sp8** $\delta_2(\mathsf{p},\mathsf{q}) \subseteq \delta_1(\mathsf{p},\mathsf{q})$ para todo $\mathsf{p},\mathsf{q} \in P_1$;

[1] Notar que, para evitar las trabas de formalismos innecesarios, se define esta notación basada en la inclusión de conjuntos; un tratamiento más general requiere una función de relación entre puntos de los escenarios.

- **Sp9** $\not\equiv_1 \subseteq \not\equiv_2$.

Es fácil ver que la siguiente propiedad se verifica:

Propiedad 3.1.2. Dados $\mathcal{S}_1$ y $\mathcal{S}_2$ dos escenarios tales que $\mathcal{S}_2 <: \mathcal{S}_1$, σ una ejecución sobre Σ_2, y $\hat{\cdot}$ un matching entre $\mathcal{S}_2$ y σ entonces $\hat{\cdot}|_{P_1}$ ($\hat{\cdot}$ restringido a P_1) es un matching entre $\mathcal{S}_1$ y σ.

Definición 3.1.3 (Escenario Condicional)**.** Dado un escenario $\mathcal{S}_0$ (*antecedente*) y un conjunto ordenado de escenarios $\mathcal{S}_1$, $\mathcal{S}_2$, ..., $\mathcal{S}_k$ (*consecuentes*), tal que $\mathcal{S}_i <: \mathcal{S}_0$ para $i \in \{1...k\}$, y $P_i \cap P_j = P_0$ para $i \neq j \in \{1...k\}$, se denomina a $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_i\}_{i=1...k} \rangle$ un *Escenario Condicional (EC).*

Definición 3.1.4 (Semántica de ECs)**.** Una ejecución $\sigma = \langle s, \tau \rangle$ satisface un EC $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_i\}_{i=1...k} \rangle$ ($\sigma \vDash \mathcal{C}$) sii para cada matching $\hat{\cdot}$ entre $\mathcal{S}_0$ y σ existe $\hat{\hat{\cdot}}$ un matching entre $\mathcal{S}_i$ y σ, **para algún** $i \in \{1...k\}$, tal que $\forall \mathsf{p} \in P_0.\ \hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$ (es decir que $\hat{\hat{\cdot}}$ *extiende* $\hat{\cdot}$).

La *Figura 3.1* presenta un ejemplo gráfico de escenario condicional. En la notación gráfica, todos los consecuentes se presentan en un único esquema donde comparten el mismo antecedente común. Los elementos en el escenario antecedente (el cual también forma parte de todos los consecuentes) se identifican en color negro, mientras que en los consecuentes (escenarios alternativos) éstos se dibujan en gris, acompañados de un número indicador del consecuente al que pertenecen. La interpretación del escenario en este ejemplo es que, cuando un evento de solicitud de acceso (*Access request*) es seguido por un evento de acceso otorgado (*Access granted*) sin un *logoff* entre ellos, una de las otras dos secuencias de eventos también tiene que ser observada. Una de éstas (consecuente 1) requiere que, posterior a la solicitud de acceso (*Access request*), se ingrese una *password* válida. La otra (consecuente 2) es el caso donde una *password* válida ha sido ingresada antes de la solicitud de acceso al recurso (*Access request*) y el tiempo entre ambos eventos no supera el umbral de *timeout*. Es importante apreciar la flexibilidad de esta notación de *trigger*, donde el antecedente no requiere preceder al consecuente en el tiempo.

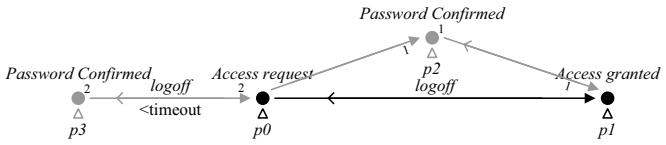


Figura 3.1: Autorización

En la *Figura 3.2* se presenta otro escenario condicional de ejemplo para un caso de estudio de un robot. En este caso, cuando el robot recibe una orden de operación (*Command*) o bien se observa un mensaje de error (consecuente 2) o, ambos motores arrancan en cualquier orden (consecuente 1), siendo el tiempo entre el primero en arrancar y el último en detenerse menor a 100 u.t. y no se observa un error.

En los escenarios condicionales, cada consecuente es una *especialización* del antecedente. Por lo tanto, tenemos que cada punto del escenario antecedente también se presenta en el consecuente. No obstante, éste puede estar etiquetado con un subconjunto de los eventos con los cuales esta definido en el antecedente. Con el motivo de visualizar esta situación, en los escenarios condicionales se incorpora la notación gráfica de *alias* representada por una doble línea entre dos puntos. La *Figura 3.3* presenta un ejemplo de un escenario utilizando la notación de *alias*. En primer lugar, el antecedente describe la siguiente situación:

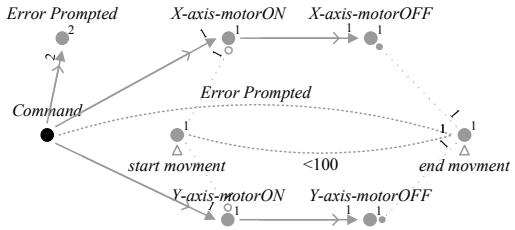


Figura 3.2: Movimiento de robot

al venir un evento *send1* o *send2* ocurre un evento *recive1* o *recive2*. Pero en el consecuente 1 se establece que si este primer evento es *send1* entonces el segundo evento corresponde a *recive1*. Para el consecuente 2 el comportamiento es análogo pero con los eventos *send2* y *recive2*.

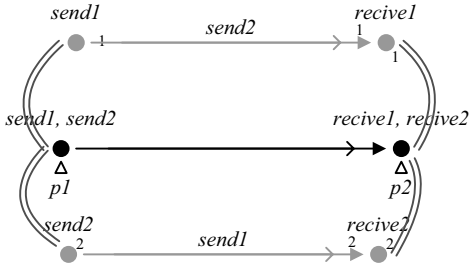


Figura 3.3: Uso de alias

En la *Figura 3.4* se presenta otro escenario condicional de ejemplo. En este caso, el escenario condicional tiene un único consecuente que incorpora el siguiente conjunto de restricciones respecto del antecedente:

- Desde el comienzo hasta p1 no ocurre ningún evento *c*.
- Desde el comienzo hasta p1 ocurren menos de 10 unidades de tiempo.
- Entre p2 y p3 hay una precedencia.
- Para p4 el evento que ocurre es necesariamente el evento *c*.
- p4 y p5 corresponden a distintos eventos de la ejecución.
- Entre p4 y p5 transcurren entre más de 10 y menos de 20 unidades de tiempo.
- Luego de p5 no ocurre ningún evento *c*.

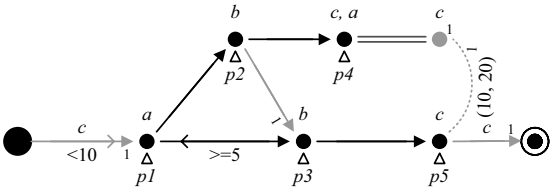


Figura 3.4: Escenario con restricciones de comienzo y fin

Ahora bien, los lenguajes definidos por ECs no son necesariamente regulares en el tiempo. Por ejemplo, el lenguaje que contiene todas las palabras temporizadas donde cada a es seguida por una b en una u.t.

no es expresable como un autómata temporizado (ver [AM04]) mientras que si es expresable como el lenguaje que acepta el simple escenario condicional de la *Figura 3.5(a)* o el complemento del lenguaje por el escenario de la *Figura 3.5(b)*. Esto implica que para los ECs una estrategia de *model-checking* basada en una construcción de *tableau* para su lenguaje o su complemento no es factible.

(a) NTR (b) Complemento NTR

Figura 3.5: Escenarios que aceptan un lenguaje no regular en el tiempo (NTR) y su complemento

Dado que el problema de verificación de autómatas temporizados para los ECs no es decidible[BKO05a], se determina trabajar con una subclase de estos denominada ECs *determinísticos*, que es decidible pero también relevante para la utilización de escenarios condicionales.

3.2 Escenarios Condicionales Determinísticos (ECDs)

En los ECs determinísticos (ECDs) los puntos en los consecuentes tienen a lo sumo una única forma de hacer matching. Lo interesante es que no sólo son útiles en la práctica, sino también adecuados para el model-checking por medio de la construcción de un conjunto de *escenarios VTS existenciales* que detectan los posibles contraejemplos que no satisfacen el ECD[2]. Desde un punto de vista sintáctico los puntos en los consecuentes de un ECD son alcanzables mediante precedencias de tipo "próximas" ($\rightarrow\!\!\!\!\rightarrow$) y "previas" ($\longleftarrow$) desde algún punto en el antecedente. En cuanto a la estructura del antecedente no hay restricciones. Los ejemplos de autorización y del robot de las *Figura 3.1* y *Figura 3.2* son actualmente ECDs dado que los puntos en sus consecuentes son alcanzables desde el antecedente siguiendo caminos donde las conexiones son precedencias de tipo próximas o previas.

Para formalizar estos conceptos, es necesario introducir las siguientes definiciones.

Definición 3.2.1 (Determinación entre pares de puntos)**.** Dados dos puntos p_1 y p_2 (donde p_2 es un punto concreto) en un escenario, se dice que p_1 *determina* p_2 (denotado $\mathsf{p}_1 \hookrightarrow \mathsf{p}_2$) sii:

1. o bien $\ell(\mathsf{p}_2) \subseteq \gamma(\mathsf{p}_1, \mathsf{p}_2)$ y $\mathsf{p}_1 < \mathsf{p}_2$,
2. o $\ell(\mathsf{p}_2) \subseteq \gamma(\mathsf{p}_2, \mathsf{p}_1)$ y $\mathsf{p}_2 < \mathsf{p}_1$.

En la notación gráfica esta propiedad puede ser expresada como $\mathsf{p}_1 \rightarrow\!\!\!\!\rightarrow \mathsf{p}_2$ o $\mathsf{p}_2 \longleftarrow \mathsf{p}_1$, respectivamente [3].

Esta propiedad se extiende a un conjunto de puntos de la siguiente manera:

[2]Notar que la estrategia clásica para verificar *triggered* escenarios consiste en construir un autómata que reconozca el complemento del lenguaje del escenario condicional. Pero este método no se puede aplicar para el lenguaje VTS dado que los autómata temporizados no son, en general, determinizables (ver [AD94]).

[3]Un operador para expresar el evento más reciente de un determinado tipo, similar a la flecha $\rightarrow\!\!\!\!\rightarrow$ (ocurrencia *próxima*), aparece en la lógica de estados-temporales de [RS97].

Definición 3.2.2 (Determinación general)**.** Dado un punto p y un conjunto de puntos P, se dice que P *determina* p (denotado como $P \hookrightarrow \mathsf{p}$) sii

1. o bien $\mathsf{p} \in P$,
2. o existe un punto $\mathsf{p}' \in P$ tal que $\mathsf{p}' \hookrightarrow \mathsf{p}$ (cuando p es un punto concreto),
3. o $FirstOf(\mathsf{p}) \subseteq P$ (cuando $\mathsf{p} \in FirstRep$),
4. o $LastOf(\mathsf{p}) \subseteq P$ (cuando $\mathsf{p} \in LastRep$).

Definición 3.2.3 (Ranking de Puntos)**.** Dado un conjunto de puntos P, un *ranking* $\lessdot$ en P es un orden total sobre P.

Definición 3.2.4 (Especialización Determinística de Escenario)**.** Dados $\mathcal{S}_1$ y $\mathcal{S}_2$ dos escenarios tal que $\mathcal{S}_2 <: \mathcal{S}_1$, se dice que $\mathcal{S}_2$ *especializa determinísticamente* $\mathcal{S}_1$ (denotado $\mathcal{S}_2 <:: \mathcal{S}_1$), sii existe un ranking $\lessdot$ sobre P_2 tal que: $\forall \mathsf{p}_1 \in P_1, \mathsf{p}_2 \in P_2 \setminus P_1$. $\mathsf{p}_1 \lessdot \mathsf{p}_2$ y $\lessdot \mathsf{p}_2 \hookrightarrow \mathsf{p}_2$ (donde $\lessdot \mathsf{p}_2 = \{\mathsf{p} \in P_2 \mid \mathsf{p} \lessdot \mathsf{p}_2\}$).

En otras palabras, los puntos concretos en $\mathcal{S}_2$ son alcanzables a través de flechas $\longrightarrow$ o $\longleftrightarrow$ desde puntos en $\mathcal{S}_1$.

Definición 3.2.5 (EC Determinístico (ECD))**.** Un escenario condicional $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_i\}_{i=1\ldots k} \rangle$ es un *Escenario Condicional Determinístico* sii para todo $i \in \{1..k\}$, $\mathcal{S}_i <:: \mathcal{S}_0$.

Capítulo 4

Verificación de Escenarios Condicionales Determinísticos

La estrategia propuesta para la verificación de los ECDs consiste en la construcción de escenarios negativos, es decir antiescenarios (expresados como escenarios de tipo VTS existenciales) que definen todas las alternativas posibles por las que el ECD podría no cumplirse. Son estos antiescenarios los que se verifican contra el modelo del sistema para determinar si el ECD se puede satisfacer.
Los antiescenarios no deben ser considerados únicamente como elementos exclusivamente técnicos y de carácter exclusivo a la verificación, son también construcciones valiosas por si mismas para los ingenieros. Por ejemplo, los antiescenarios generados a verificar pueden determinarse que se satisfacen en una simple inspección realizada por un diseñador, por lo que se evita el paso de model-checking. En resumen, los antiescenarios generados a partir de los escenarios condicionales son una descripción entendible por ingenieros y diseñadores de cómo podrían darse situaciones que no se quiere que ocurran en el sistema.

4.1 Introducción al algoritmo

A continuación se introduce el algoritmo de construcción de antiescenarios por medio de los ejemplos de autorización (*Figura 3.1*), del robot (*Figura 3.2*), del escenario con restricciones de comienzo y fin (*Figura 3.4*), y un nuevo escenario con restricciones de representativos (*Figura 4.1*) que es incorporado para cubrir la mayor parte de los casos tratado por este procedimiento.

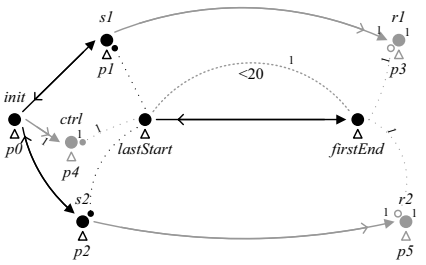
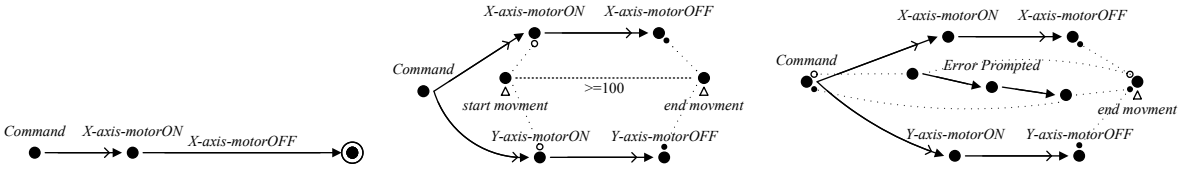


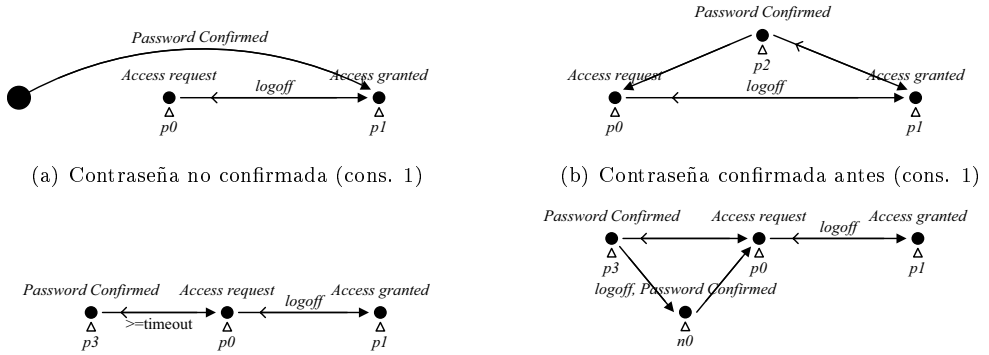
Figura 4.1: ECD con representativos para ilustrar la construcción de antiescenarios

Para cada consecuente, el algoritmo procede con la construcción de un conjunto de escenarios negativos. Si alguno de estos se satisface, revela que hay una manera de hacer un matching del antecedente en una ejecución, donde alguna parte del patrón propia al consecuente no se satisface. Inicialmente genera los escenarios negativos resultantes donde los puntos del consecuente no son encontrado en la posición relativa donde se esperan (puntos sin matching, p.ej., ver *Figuras 4.2(a), 4.3(a), 4.4(a)*. Para construir estos antiescenarios el algoritmo usa el orden definido por el *ranking* con el fin de establecer el camino de precedencia que conduce al punto no detectado.



(a) El motor-X no se detiene (b) Demasiado lento (c) Mensaje de error mientras hay movimiento

Figura 4.2: Algunos de los antiescenarios intermedios para el ejemplo del robot



(a) Contraseña no confirmada (cons. 1)

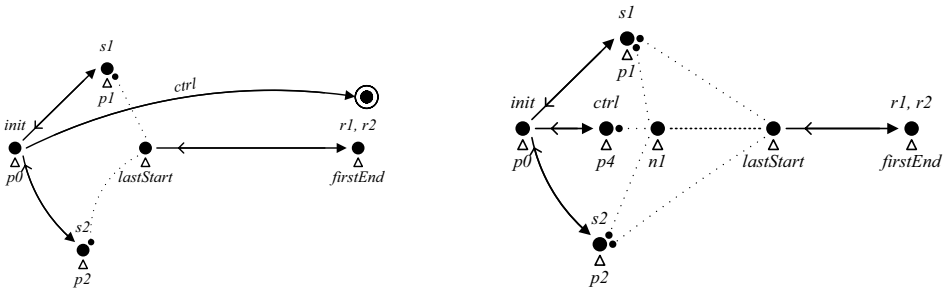
(b) Contraseña confirmada antes (cons. 1)

(c) Violación del plazo al confirmarse la contraseña (cons. 2)

(d) Logoff luego de la última confirmación (cons. 2)

Figura 4.3: Algunos antiescenarios intermedios para el ejemplo de autorización

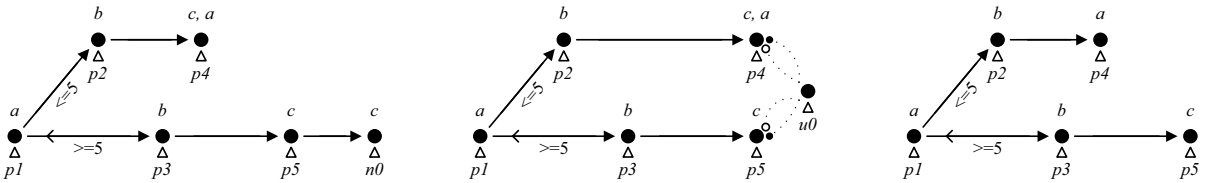
Luego, para cada par de puntos en el consecuente, los antiescenarios son construidos expresando el caso donde aunque estos puntos en el consecuente se encuentran siguiendo el camino de precedencias desde el antecedente, una restricción entre pares de puntos que debería respetarse según el escenario condicional es incumplida. Las restricciones incluyen relaciones de precedencias (p.ej., *Figura 4.3(b)*), eventos prohibidos (p.ej., *Figuras 4.2(c), 4.3(d),4.5(a)*), restricciones temporales (p.ej., *Figuras 4.2(b), 4.3(c)*), y restricciones de desigualdad (p.ej., *Figura 4.5(b)*). En verdad, los patrones de los consecuentes corresponden a una conjunción de estas restricciones atómicas entre pares de puntos junto con restricciones representativas (las cuales pueden de hecho involucrar varios puntos). En relación a éstas últimas, los elementos de construcción *primero* y *último* permiten al diseñador expresar situaciones donde un punto representativo del antecedente es también representativo de puntos del consecuente, puntos del consecuente que son al mismo tiempo representativos-primeros y representativos-últimos, etc. Entonces los constructores *primero* y *último* pueden establecer restricciones de aliasing ("el primero en este conjunto es también el último en este otro conjunto"). Por lo tanto, para estos casos también es necesario construir escenarios que violan las restricciones de aliasing subyacentes (p.ej.,*Figura 4.4(b)*).



(a) Luego del evento *init* no ocurre *ctrl*

(b) El último evento en ocurrir de (*s1, s2, ctrl*) no coincide con el último de (*s1, s2*)

Figura 4.4: Algunos antiescenarios generados para el ejemplo de representativos



(a) Luego de *p5* ocurre un evento *c*

(b) Los puntos p4 y p5 no satisfacen la restricción de desigualdad

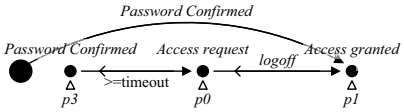
(c) El evento que ocurren en *p4* puede ser distinto de *c*

Figura 4.5: Algunos antiescenarios generados para el ejemplo de restricciones de comienzo y fin

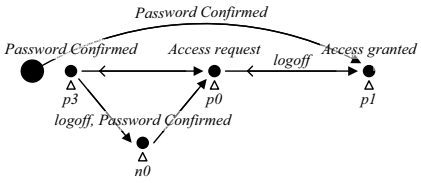
Por último, se construye un antiescenario para cada punto del antecedente donde en el consecuente éstos puntos no tiene asociados todos los eventos, es decir cuando el consecuente limita los eventos que pueden ocurrir en puntos del antecedente (p.ej.,*Figura 4.5(c)*).

Se puede ver que, si hay un único escenario consecuente, estos contraejemplos genéricos podrían comprender todas las formas en que el ECD podría ser violado. Ahora, si existen n consecuentes alternativos, los contraejemplos deben obtenerse generando todas las n-fusiones (ver a continuación) que resultan de tomar un antiescenario para cada uno de los n consecuentes. Es decir, los contraejemplos de los consecuentes se combinan, dado que se está negando una disjunción de conjunciones (ver *Figura 4.6*). De todas formas, muchas de estas combinaciones de escenario en la práctica son imposibles por construcción, y por lo tanto no requieren pasar a la fase de model-checking (*Figura 4.6(a)* y *Figura 4.6(b)*) son los dos antiescenarios finales imposibles). Por lo tanto, los escenarios que requieren verificarse con el Modelchecker son sólo los últimos dos en la *Figura 4.6*.

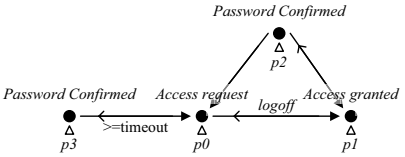
A continuación se introduce la operación de *fusión*, para obtener un escenario por la combinación de dos o más escenarios. Esta operación esta basada en la unión de conjuntos; entonces, si dos escenarios combinados comparten puntos y relaciones, en la construcción final estos elementos aparecerán unificados. Por lo tanto, la fusión es más restrictiva que la intersección de los lenguajes (esta última, conformada por la unión disjunta de patrones); la fusión requiere no sólo la existencia de los matchings para ambos patrones sino que verifica que estos coincidan en el sub-patrón en común (en nuestro caso, el antecedente).



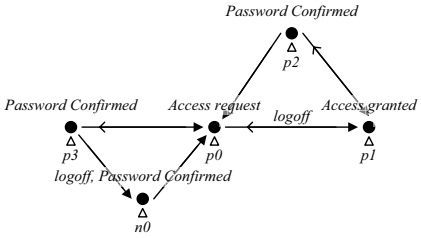
(a) Antiescenario final (a ⊕ c) -imposible-



(b) Antiescenario final (a ⊕ d) -imposible-



(c) Antiescenario final (b ⊕ c)



(d) Antiescenario final (b ⊕ d)

Figura 4.6: Algunos de los antiescenarios finales para el ejemplo de autorización

Definición 4.1.1 (Fusión)**.** Dados $\mathcal{S}_1$ y $\mathcal{S}_2$ dos escenarios [1] se define la *fusión* entre $\mathcal{S}_1$ y $\mathcal{S}_2$ como el escenario $\mathcal{S}_1 \oplus \mathcal{S}_2 = \langle \Sigma_{12}, P_{12}, \ell_{12}, \not\equiv_{12}, <_{12}, <_{F12}, <_{L12}, \gamma_{12}, \delta_{12} \rangle$ donde:

- $\Sigma_{12} = \Sigma_1 \cup \Sigma_2$;
- $P_{12} = P_1 \cup P_2$;
- $\ell_{12}(\mathsf{p}) = \ell_1(\mathsf{p})$ si $\mathsf{p} \in P_1 \setminus P_2$,
  $\ell_{12}(\mathsf{p}) = \ell_2(\mathsf{p})$ si $\mathsf{p} \in P_2 \setminus P_1$, o
  $\ell_{12}(\mathsf{p}) = \ell_1(\mathsf{p}) \cap \ell_2(\mathsf{p})$ [2] si $\mathsf{p} \in P_1 \cap P_2$;
- $\not\equiv_{12} = (\not\equiv_1 \cup \not\equiv_2)$;
- $<_{12} = (<_1 \cup <_2)$;
- $<_{F12} = (<_{F1} \cup <_{F2})$;
- $<_{L12} = (<_{L1} \cup <_{L2})$;
- $\gamma_{12} = \gamma_1 \cup \gamma_2$;
- $\delta_{12}(\mathbf{0}, \mathsf{p}) = \delta_1(\mathbf{0}, \mathsf{p})$ si $(\mathbf{0}, \mathsf{p}) \in \delta_1 \setminus \delta_2$,
  $\delta_{12}(\mathbf{0}, \mathsf{p}) = \delta_2(\mathbf{0}, \mathsf{p})$ si $(\mathbf{0}, \mathsf{p}) \in \delta_2 \setminus \delta_1$,
  $\delta_{12}(\mathbf{0}, \mathsf{p}) = \delta_1(\mathbf{0}, \mathsf{p}) \cap \delta_2(\mathbf{0}, \mathsf{p})$ si $(\mathbf{0}, \mathsf{p}) \in \delta_1 \cap \delta_2$,
  $\delta_{12}(\mathsf{p}, \mathsf{q}) = \delta_1(\mathsf{p}, \mathsf{q})$ si $(\mathsf{p}, \mathsf{q}) \in \delta_1 \setminus \delta_2$,
  $\delta_{12}(\mathsf{p}, \mathsf{q}) = \delta_2(\mathsf{p}, \mathsf{q})$ si $(\mathsf{p}, \mathsf{q}) \in \delta_2 \setminus \delta_1$, o
  $\delta_{12}(\mathsf{p}, \mathsf{q}) = \delta_1(\mathsf{p}, \mathsf{q}) \cap \delta_2(\mathsf{p}, \mathsf{q})$ si $(\mathsf{p}, \mathsf{q}) \in \delta_1 \cap \delta_2$.

Propiedad 4.1.2 (Especialización minimal)**.** Dados $\mathcal{S}_1$ y $\mathcal{S}_2$ dos escenarios, entonces $\mathcal{S}_1 \oplus \mathcal{S}_2 <: \mathcal{S}_1$ y $\mathcal{S}_1 \oplus \mathcal{S}_2 <: \mathcal{S}_2$. La fusión $\mathcal{S}_1 \oplus \mathcal{S}_2$ es una *especialización minimal* de $\mathcal{S}_1$ y $\mathcal{S}_2$; lo que significa, que para cualquier otro escenario $\mathcal{S}$, donde $\mathcal{S} <: \mathcal{S}_1$, y $\mathcal{S} <: \mathcal{S}_2$; entonces $\mathcal{S} <: \mathcal{S}_1 \oplus \mathcal{S}_2$.

[1] En caso que $\not\equiv_1$ y $\not\equiv_2$ contengan un par y su transmutado, los escenarios deben compatibilizarse antes de realizar la fusión eligiendo una dirección y consecuentemente invirtiendo los pares en las definiciones correspondientes de γ y δ, con el objetivo de mantener la asimetría.

[2] observar que cuando el etiquetado de la intersección es vació la fusión genera un escenario imposible dado que no puede existir un matching que cumpla la condición **M1**.

4.2 Construcción de antiescenarios

Definición 4.2.1 (Mínimos Determinantes)**.** Dados $\mathcal{S}_1$ y $\mathcal{S}_2$ dos escenarios tal que $\mathcal{S}_2 <:: \mathcal{S}_1$ por un ranking $\lessdot$ sobre P_2, y un punto $\mathsf{p} \in P_2 \setminus P_1$, se definen los *Mínimos Determinantes* del punto p como el conjunto de puntos $ld_{\lessdot}(\mathsf{p}) = \{\text{mín}_{\lessdot}\{\mathsf{q} \in P_2 \mid \mathsf{q} \hookrightarrow \mathsf{p}\}\}$ –esto es, el conjunto unitario que contiene el primer elemento (con respecto al ranking $\lessdot$) del conjunto de puntos que determinan p– cuando p es un punto concreto, y cuando es representativo $ld_{\lessdot}(\mathsf{p}) = FirstOf_2(\mathsf{p})$ cuando $FirstOf_2(\mathsf{p}) \subseteq \lessdot\mathsf{p}$, o $LastOf_2(\mathsf{p})$, en otro caso.

Ahora se define como generar un escenario que representa como un punto en el consecuente es alcanzable desde el antecedente mediante un camino de precedencias con restricciones de tipo *próximas* o *previas*.

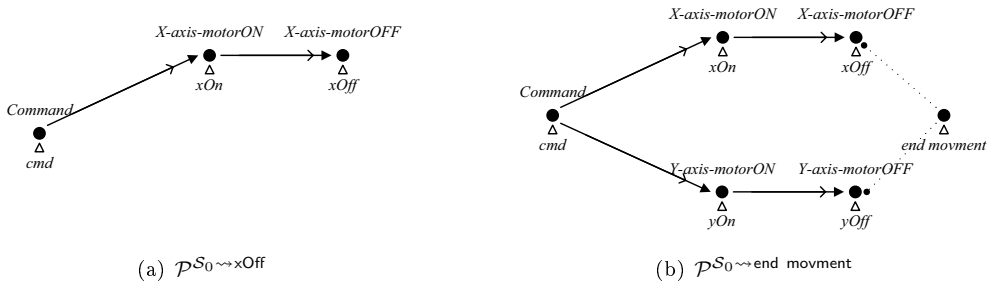
Definición 4.2.2 (Escenario del Camino)**.** Dados $\mathcal{S}_1$ y $\mathcal{S}_2$ dos escenarios tal que $\mathcal{S}_2 <:: \mathcal{S}_1$ por un ranking $\lessdot$ sobre P_2, y un punto $\mathsf{p} \in P_2$, se construye el *escenario del camino* a p desde $\mathcal{S}_1$ (denominado $\mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}}$) como el siguiente escenario:

- $\mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} = \mathcal{S}_1$ cuando $\mathsf{p} \in P_1$.
- $\mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \hookleftarrow \mathsf{p}}$ cuando p es un punto concreto en $P_2 \setminus P_1$ y $\mathsf{q} \in ld_{\lessdot}(\mathsf{p})$, donde
  $\mathcal{P}^{\mathsf{q} \hookleftarrow \mathsf{p}} = \langle \ell_2(\mathsf{p}) \cup \ell_q(\mathsf{q}), \{\mathsf{p}, \mathsf{q}\}, \{(\mathsf{p}, \ell_2(\mathsf{p})), (\mathsf{q}, \ell_q(\mathsf{q}))\}, \emptyset, \{(\mathsf{p}, \mathsf{q})\}, \emptyset, \emptyset, \{(\mathsf{p}, \mathsf{q}, \ell_2(\mathsf{p}))\}, \emptyset \rangle$ cuando $\mathsf{p} <_2 \mathsf{q}$ o
  $\mathcal{P}^{\mathsf{q} \hookleftarrow \mathsf{p}} = \langle \ell_2(\mathsf{p}) \cup \ell_q(\mathsf{q}), \{\mathsf{p}, \mathsf{q}\}, \{(\mathsf{p}, \ell_2(\mathsf{p})), (\mathsf{q}, \ell_p(\mathsf{q}))\}, \emptyset, \{(\mathsf{q}, \mathsf{p})\}, \emptyset, \emptyset, \{(\mathsf{q}, \mathsf{p}, \ell_2(\mathsf{p}))\}, \emptyset \rangle$ cuando $\mathsf{q} <_2 \mathsf{p}$.
- $\mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} = \bigoplus_{\mathsf{q} \in ld_{\lessdot}(\mathsf{p})} (\mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \hookleftarrow \mathsf{p}})$ cuando p es un punto representativo en $P_2 \setminus P_1$, donde
  $\mathcal{P}^{\mathsf{q} \hookleftarrow \mathsf{p}} = \langle \Sigma_1, \{\mathsf{p}, \mathsf{q}\}, \{(\mathsf{p}, \Sigma_1 \cup \{\lambda\}), (\mathsf{q}, \ell_q(\mathsf{q}))\}, \emptyset, \emptyset, \{(\mathsf{p}, \mathsf{q})\}, \emptyset, \emptyset, \emptyset \rangle$ cuando $\mathsf{p} <_{F2} \mathsf{q}$ o
  $\mathcal{P}^{\mathsf{q} \hookleftarrow \mathsf{p}} = \langle \Sigma_1, \{\mathsf{p}, \mathsf{q}\}, \{(\mathsf{p}, \Sigma_1 \cup \{\lambda\}), (\mathsf{q}, \ell_q(\mathsf{q}))\}, \emptyset, \emptyset, \emptyset, \{(\mathsf{q}, \mathsf{p})\}, \emptyset, \emptyset \rangle$ cuando $\mathsf{q} <_{L2} \mathsf{p}$.

  siendo $\Sigma_q, P_q, \ell_q, \not\equiv_q, <_q, <_{Fq}, <_{Lq}, \gamma_q, \delta_q$ la notación que refiere a los elementos de $\mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}}$.

Observar que $\mathcal{P}^{\mathsf{q} \hookleftarrow \mathsf{p}}$ representa la manera en que q determina a p en $\mathcal{S}_2$ y $\mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}}$ (un **escenario del camino** desde $\mathcal{S}_1$ hacia p) explica como el punto p es determinísticamente alcanzado en $\mathcal{S}_2$ desde $\mathcal{S}_1$.

Las *Figuras 4.7(a)* y *4.7(b)* presentan dos escenarios del camino para el ejemplo del Robot $\langle \mathcal{S}_0, \{\mathcal{S}_1, \mathcal{S}_2\} \rangle$. Ambos escenarios del camino son hacia puntos del consecuente $\mathcal{S}_1$ utilizando el ranking de puntos definido por el siguiente orden: cmd, xOn, xOff, yOn, yOff, start movment, end movment.



(a) $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{xOff}}$ (b) $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{end\ movment}}$

Figura 4.7: Algunos escenarios del camino para el ejemplo del Robot

Definición 4.2.3 (Antiescenarios para puntos sin matching)**.** Dados $\mathcal{S}_1$ y $\mathcal{S}_2$ dos escenarios tal que $\mathcal{S}_2 <:: \mathcal{S}_1$ por un ranking $\lessdot$ sobre P_2, y un punto concreto $\mathsf{p} \in P_2 \setminus P_1$, se construye el escenario "p *sin matching*" (denominado $\mathcal{N}^{\mathcal{S}_1 \not\rightsquigarrow \mathsf{p}}$) como:

- $\mathcal{N}^{\mathcal{S}_1 \not\rightsquigarrow \mathsf{p}} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}}$, donde $\{\mathsf{q}\} = ld_{\lessdot}(\mathsf{p})$ y
  $\mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}} = \langle \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{q}\}, \{(\mathsf{q}, \ell_2(\mathsf{q}))\}, \emptyset, \{(\mathbf{0}, \mathsf{q})\}, \emptyset, \emptyset, \{(\mathbf{0}, \mathsf{q}, \ell_2(\mathsf{p}))\}, \emptyset \rangle$ cuando $\mathsf{p} <_2 \mathsf{q}$ o
  $\mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}} = \langle \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{q}\}, \{(\mathsf{q}, \ell_2(\mathsf{q}))\}, \emptyset, \{(\mathsf{q}, \infty)\}, \emptyset, \emptyset, \{(\mathsf{q}, \infty, \ell_2(\mathsf{p}))\}, \emptyset \rangle$ cuando $\mathsf{q} <_2 \mathsf{p}$.

El escenario $\mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}}$ representa como infringir la existencia de p.

Utilizando una aproximación similar, se presenta la construcción del resto de los antiescenarios, es decir, aquellos que modelan la violación de restricciones entre dos puntos, y antiescenarios que modelan puntos representativos que en el escenario condicional deberían hacer matching en diferentes posiciones.

A continuación se utiliza la notación de $\Sigma_p, P_p, \ell_p, \not\equiv_p, <_p, <_{Fp}, <_{Lp}, \gamma_p, \delta_p$ para los elementos de $\mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}}$. Se denota $\mathsf{p} \neq \mathsf{q}$ el hecho que o bien $\mathsf{p} \not\equiv \mathsf{q}$, o $\mathsf{p} < \mathsf{q}$, o $\mathsf{q} < \mathsf{p}$.

Definición 4.2.4 (Antiescenarios que modelan la violación de restricciones)**.** Dados dos escenarios, $\mathcal{S}_1$ y $\mathcal{S}_2$, tal que $\mathcal{S}_2 \ <:: \ \mathcal{S}_1$, por un ranking $\lessdot$ sobre P_2, para cada punto $\mathsf{p} \in P_2$, se construye el siguiente conjunto de escenarios $\mathcal{N}$ que corresponden a escenarios que violan restricciones que involucran a p. A continuación n, f, l, i, u son nuevos puntos:

- $(\mathsf{p}, \infty) \in <_2$
  - $\mathcal{N}^{\neg\gamma_2(\mathsf{p},\infty)} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\gamma_2(\mathsf{p},\infty)}$ cuando $(\mathsf{p}, \infty) \in \gamma_2 \smallsetminus \gamma_p$, donde
    $\mathcal{P}^{\neg\gamma_2(\mathsf{p},\infty)} = \langle \ell_2(\mathsf{p}) \cup \gamma_2(\mathsf{p}, \infty), \{\mathsf{p}, \mathsf{n}\}, \{(\mathsf{p}, \ell_2(\mathsf{p})), (\mathsf{n}, \gamma_2(\mathsf{p}, \infty) \smallsetminus \gamma_p(\mathsf{p}, \infty))\}, \emptyset, \{(\mathsf{p}, \mathsf{n})\}, \emptyset, \emptyset, \emptyset, \emptyset \rangle$.
- $(\mathbf{0}, \mathsf{p}) \in <_2$
  - $\mathcal{N}^{\neg\gamma_2(\mathbf{0},\mathsf{p})} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\gamma_2(\mathbf{0},\mathsf{p})}$ cuando $(\mathbf{0}, \mathsf{p}) \in \gamma_2 \smallsetminus \gamma_p$, donde
    $\mathcal{P}^{\neg\gamma_2(\mathbf{0},\mathsf{p})} = \langle \ell_2(\mathsf{p}) \cup \gamma_2(\mathbf{0}, \mathsf{p}), \{\mathsf{p}, \mathsf{n}\}, \{(\mathsf{p}, \ell_2(\mathsf{p}), (\mathsf{n}, \gamma_2(\mathbf{0}, \mathsf{p}) \smallsetminus \gamma_p(\mathbf{0}, \mathsf{p}))\}, \emptyset, \{(\mathsf{n}, \mathsf{p})\}, \emptyset, \emptyset, \emptyset, \emptyset \rangle$.
  - $\mathcal{N}^{\neg\delta_2(\mathbf{0},\mathsf{p})} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\delta_2(\mathbf{0},\mathsf{p})}$ cuando $(\mathbf{0}, \mathsf{p}) \in \delta_2 \subsetneq \delta_p$, donde
    $\mathcal{P}^{\neg\delta_2(\mathbf{0},\mathsf{p})} = \langle \ell_2(\mathsf{p}), \{\mathsf{p}\}, \{(\mathsf{p}, \ell_2(\mathsf{p})\}, \emptyset, \{(\mathbf{0}, \mathsf{p})\}, \emptyset, \emptyset, \emptyset, \{(\mathbf{0}, \mathsf{p}, \neg\delta_2(\mathbf{0}, \mathsf{p}))\}\rangle$.

Para todo $\mathsf{q} \in P_2$ se construyen los escenarios que violan la restricción que involucra a p y q:

- $(\mathsf{p}, \mathsf{q}) \in <_2$
  - $\mathcal{N}^{\mathsf{p} \not<_2 \mathsf{q}} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not<_2 \mathsf{q}}$, cuando $(\mathsf{p}, \mathsf{q}) \notin (<_p \cup <_q)$ donde
    $\mathcal{P}^{\mathsf{p} \not<_2 \mathsf{q}} = \langle \{\ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q})\}, \{\mathsf{p}, \mathsf{q}\}, \{(\mathsf{p}, \ell_2(\mathsf{p})), (\mathsf{q}, \ell_2(\mathsf{q}))\}, \emptyset, \{(\mathsf{q}, \mathsf{p})\}, \emptyset, \emptyset, \emptyset, \emptyset \rangle$.
  - $\mathcal{N}^{\mathsf{p} \not\equiv_2 \mathsf{q}} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not\equiv_2 \mathsf{q}}$ cuando $(\mathsf{p}, \mathsf{q}) \notin (\neq_p \cup \neq_q)$ donde
    $\mathcal{P}^{\mathsf{p} \not\equiv_2 \mathsf{q}} = \langle \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{p}, \mathsf{q}, \mathsf{u}\}, \{(\mathsf{p}, \ell_2(\mathsf{p})), (\mathsf{q}, \ell_2(\mathsf{q})), (\mathsf{u}, \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}))\} \emptyset, \emptyset, \{(\mathsf{u}, \mathsf{p}), (\mathsf{u}, \mathsf{q})\}, \{(\mathsf{p}, \mathsf{u}), (\mathsf{q}, \mathsf{u})\}, \emptyset, \emptyset \rangle$.
  - $\mathcal{N}^{\neg\gamma_2(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\gamma_2(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p}, \mathsf{q}) \in \gamma_2 \smallsetminus (\gamma_p \cup \gamma_q)$, donde
    $\mathcal{P}^{\neg\gamma_2(\mathsf{p},\mathsf{q})} = \langle \gamma_2(\mathsf{p}, \mathsf{q}) \cup \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{p}, \mathsf{q}, \mathsf{n}\}, \{(\mathsf{p}, \ell_2(\mathsf{p}), (\mathsf{q}, \ell_2(\mathsf{q}), (\mathsf{n}, \gamma_2(\mathsf{p}, \mathsf{q}))\}, \emptyset, \{(\mathsf{p}, \mathsf{n}), (\mathsf{n}, \mathsf{q})\}, \emptyset, \emptyset, \emptyset, \emptyset \rangle$.
  - $\mathcal{N}^{\neg\delta_2(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\delta_2(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p}, \mathsf{q}) \in \delta_2$, donde
    $\mathcal{P}^{\neg\delta_2(\mathsf{p},\mathsf{q})} = \langle \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{p}, \mathsf{q}\}, \{(\mathsf{p}, \ell_2(\mathsf{p})), (\mathsf{q}, \ell_2(\mathsf{q}))\}, \emptyset, \{(\mathsf{p}, \mathsf{q})\}, \emptyset, \emptyset, \emptyset, \{(\mathsf{p}, \mathsf{q}, \neg\delta_2(\mathsf{p}, \mathsf{q}))\}\rangle$.
- $(\mathsf{p}, \mathsf{q}) \in (\not\equiv_2)$
  - $\mathcal{N}^{\mathsf{p} \not\equiv_2 \mathsf{q}} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not\equiv_2 \mathsf{q}}$ cuando $(\mathsf{p}, \mathsf{q}) \notin (\neq_p \cup \neq_q)$ donde
    $\mathcal{P}^{\mathsf{p} \not\equiv_2 \mathsf{q}} = \langle \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{p}, \mathsf{q}, \mathsf{u}\}, \{(\mathsf{p}, \ell_2(\mathsf{p})), (\mathsf{q}, \ell_2(\mathsf{q})), (\mathsf{u}, \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}))\}, \emptyset, \emptyset, \{(\mathsf{u}, \mathsf{p}), (\mathsf{u}, \mathsf{q})\}, \{(\mathsf{p}, \mathsf{u}), (\mathsf{q}, \mathsf{u})\}, \emptyset, \emptyset \rangle$.

– $\mathcal{N}^{\neg\gamma_2(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\gamma_2(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p},\mathsf{q}) \in \gamma_2 \setminus (\gamma_p \cup \gamma_q)$, donde
$\mathcal{P}^{\neg\gamma_2(\mathsf{p},\mathsf{q})} = \langle \gamma_2(\mathsf{p},\mathsf{q}) \cup \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{p},\mathsf{q},\mathsf{f},\mathsf{i},\mathsf{l}\}, \{(\mathsf{p},\ell_2(\mathsf{p})),(\mathsf{q},\ell_2(\mathsf{q})),(\mathsf{f},\ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q})),(\mathsf{i},\gamma_2(\mathsf{p},\mathsf{q})),$
$(\mathsf{l},\ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}))\}, \emptyset, \{(\mathsf{f},\mathsf{i}),(\mathsf{i},\mathsf{l})\}, \{(\mathsf{f},\mathsf{p}),(\mathsf{f},\mathsf{q})\}, \{(\mathsf{p},\mathsf{l}),(\mathsf{q},\mathsf{l})\}, \emptyset, \emptyset\rangle$.
– $\mathcal{N}^{\neg\delta_2(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\delta_2(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p},\mathsf{q}) \in \delta_2$, donde
$\mathcal{P}^{\neg\delta_2(\mathsf{p},\mathsf{q})} = \langle \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{p},\mathsf{q}\}, \{(\mathsf{p},\ell_2(\mathsf{p})),(\mathsf{q},\ell_2(\mathsf{q}))\}, \{(\mathsf{p},\mathsf{q})\}, \emptyset, \emptyset, \emptyset, \{(\mathsf{p},\mathsf{q},\neg\delta_2(\mathsf{p},\mathsf{q}))\}\rangle$.

Definición 4.2.5 (Antiescenarios que modelan la violación de Aliasing)**.** Datos dos escenarios, $\mathcal{S}_1$ y $\mathcal{S}_2$, tal que $\mathcal{S}_2 <:: \mathcal{S}_1$ por un ranking $\lessdot$ sobre P_2, y un punto $\mathsf{p} \in P_2$ se construye el escenario "p *viola el aliasing representativo*" (denotado $\mathcal{N}^{\mathsf{p}\leftrightarrow\mathsf{p}}$) como:

- $\mathcal{N}^{\mathsf{p}\leftrightarrow\mathsf{p}} = \bigoplus_{\mathsf{q}\in FirstOf_2(\mathsf{p})}(\mathcal{P}^{\mathcal{S}_1\rightsquigarrow\mathsf{q}} \oplus \mathcal{P}^{\mathsf{n}<_F\mathsf{q}}) \oplus \mathcal{P}^{\mathcal{S}_1\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathsf{p}\not\equiv\mathsf{n}}$ cuando $\exists\mathsf{q}(\mathsf{p},\mathsf{q}) \in <_{F2} \setminus <_{Fp}$, donde
$\mathcal{P}^{\mathsf{n}<_F\mathsf{q}} = \langle \Sigma_1, \{\mathsf{n},\mathsf{q}\}, \{(\mathsf{n},\Sigma_1 \cup \{\lambda\}),(\mathsf{q},\ell_2(\mathsf{q}))\}, \emptyset, \emptyset, \{(\mathsf{n},\mathsf{q})\}, \emptyset, \emptyset, \emptyset\rangle$ y
$\mathcal{P}^{\mathsf{p}\not\equiv\mathsf{n}} = \langle \Sigma_1, \{\mathsf{p},\mathsf{n}\}, \{(\mathsf{n},\Sigma_1 \cup \{\lambda\}),(\mathsf{p},\ell_p(\mathsf{p}))\}, \{(\mathsf{n},\mathsf{p})\}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\rangle$.
- $\mathcal{N}^{\mathsf{p}\leftrightarrow\mathsf{p}} = \bigoplus_{\mathsf{q}\in LastOf_2(\mathsf{p})}(\mathcal{P}^{\mathcal{S}_1\rightsquigarrow\mathsf{q}} \oplus \mathcal{P}^{\mathsf{q}<_L\mathsf{n}}) \oplus \mathcal{P}^{\mathcal{S}_1\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathsf{p}\not\equiv\mathsf{n}}$ cuando $\exists\mathsf{q}(\mathsf{q},\mathsf{p}) \in <_{L2} \setminus <_{Lp}$, donde
$\mathcal{P}^{\mathsf{q}<_L\mathsf{n}} = \langle \Sigma_1, \{\mathsf{n},\mathsf{q}\}, \{(\mathsf{n},\Sigma_1 \cup \{\lambda\}),(\mathsf{q},\ell_2(\mathsf{q}))\}, \emptyset, \emptyset, \emptyset, \{(\mathsf{q},\mathsf{n})\}, \emptyset, \emptyset\rangle$ y
$\mathcal{P}^{\mathsf{p}\not\equiv\mathsf{n}} = \langle \Sigma_1, \{\mathsf{p},\mathsf{n}\}, \{(\mathsf{n},\Sigma_1 \cup \{\lambda\}),(\mathsf{p},\ell_p(\mathsf{p}))\}, \{(\mathsf{n},\mathsf{p})\}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\rangle$.

Finalmente se construyen los antiescenarios los casos donde el consecuente limita los eventos que pueden ocurrir en puntos del antecedente.

Definición 4.2.6 (Antiescenarios que modelan la violación de eventos limitados)**.** Dados $\mathcal{S}_1$ y $\mathcal{S}_2$ dos escenarios tal que $\mathcal{S}_2 <:: \mathcal{S}_1$ por un ranking $\lessdot$ sobre P_2, para cada punto $\mathsf{p} \in P_1$, se construye el escenario "p *viola la limitación de eventos*" (denotado $\mathcal{N}^{\mathcal{S}_1\not\Rightarrow\mathsf{p}}$) como:

- $\mathcal{N}^{\mathcal{S}_1\not\Rightarrow\mathsf{p}} = \mathcal{P}^{\mathcal{S}_1\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\not\Rightarrow\mathsf{p}}$, cuando $\ell_2(\mathsf{p}) \subsetneq \ell_1(\mathsf{p})$, donde
$\mathcal{P}^{\not\Rightarrow\mathsf{p}} = \langle \ell_1(\mathsf{p}) \setminus \ell_2(\mathsf{p}), \{\mathsf{p}\}, \{(\mathsf{p},\ell_1(\mathsf{p}) \setminus \ell_2(\mathsf{p}))\}, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset\rangle$.

De esta forma, dado un ECD $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_i\}_{i=1...k}\rangle$ y un conjunto de rankings $\lessdot_i$ sobre P_i para $i \in \{1...k\}$, se construye, para cada punto $\mathsf{p} \in P_i$, el conjunto de escenarios de contraejemplos $\mathcal{N}$ detallados anteriormente. Cada escenario en este conjunto representa una manera de violar un consecuente $\mathcal{S}_i$ en el ECD $\mathcal{C}$.

Finalmente, se construyen todos los escenarios que resultan de las fusiones obtenidas de seleccionar un contraejemplo para cada consecuente (con el fin de reconocer todas las situaciones donde no hay posibilidad de extensión entre las distintos consecuentes alternativos). El resultado es que el ECD se satisface si y sólo si ninguno de los contraejemplos finales generados se satisface. En consecuencia, se tiene un mecanismo efectivo para comprobar los ECDs por medio de la verificación de la intersección del lenguaje del modelo respecto a su lenguaje complementario, el cual es un lenguaje expresable usando los escenarios VTS existenciales.

Teorema 4.2.7 (Verificación de ECD)**.** Sea $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_i\}_{i=1...k}\rangle$ un ECD, σ una ejecución, entonces:

$$\sigma \vDash \mathcal{C} \iff \text{todo } \mathcal{N}_\mathcal{C} = \langle \mathcal{N}_{\mathcal{S}_1} \oplus \mathcal{N}_{\mathcal{S}_2} \oplus ... \oplus \mathcal{N}_{\mathcal{S}_k}\rangle \text{ un antiescenario final, donde } \mathcal{N}_{\mathcal{S}_j\, j=1...k}$$

es un antiescenario generado por las reglas de $\mathcal{S}_j$, verifica $\sigma \nvDash \mathcal{N}_\mathcal{C}$.

Notar que en caso que un consecuente sea exactamente el mismo al antecedente (donde no se incorporan nuevos puntos o restricciones) el conjunto resultante de antiescenarios del consecuente será vacío. Por lo tanto, como no hay antiescenarios para elegir de este consecuente, no se puede construir ningún antiescenario final, y entonces se satisface el escenario condicional como es de esperar.

4.3 Complejidad del algoritmo

El número de antiescenarios generados para cada consecuente es, aproximadamente, tres veces la cantidad de puntos pertenecientes exclusivamente al consecuente. Notar que estas es una mejor situación que la aproximación tradicional de complementar el tableau para que reconozca el lenguaje de las ejecuciones que se satisfacen (exponencial en el tamaño del tableau).

Respecto a los antiescenarios finales, la cantidad total esta determinada por el producto de la cantidad de antiescenarios para cada consecuente. Sin embargo, en la práctica el número de consecuentes es usualmente pequeño, y asimismo, antes del modelchecking, es posible descartar todos los escenarios imposibles resultantes de la fusión de los antiescenarios intermedios.

Finalmente, el tamaño del traductor de tableau, para un único antiescenario final depende del número de *configuraciones* que, según el número de puntos, varía de lineal a exponencial, dependiendo del grado de concurrencia. El peor caso de complejidad exponencial no es raro al trabajar con formalismos de especificación de propiedades "lineales en el tiempo". Afortunadamente, los escenarios que los ingenieros y diseñadores generan para verificar las propiedades relacionadas con los requerimientos tienden a ser de tamaño reducido.

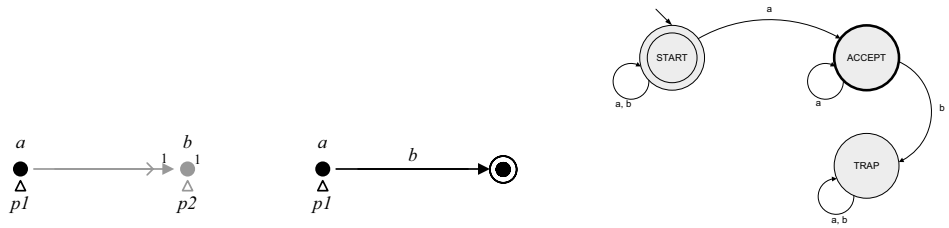
Capítulo 5

Optimizaciones

5.1 Regla de puntos sin matching

Cuando se aplica la regla de "puntos sin matching" el antiescenario que se construye incorpora una restricción de eventos prohibidos que se relaciona con el elemento *comienzo* o bien con el elemento *fin*. En este último caso, la restricción respecto al elemento *fin* le ocasiona al modelchecker un elevado costo de verificación. Esto ocurre porque el modelchecker tiene que verificar una condición de aceptación de "Liveness". Es decir, no sólo tiene que verificar que, en el autómata del antiescenario, la ejecución alcance el estado de aceptación sino que también debe comprobar que posteriormente no ocurra un evento prohibido hasta el final de la ejecución.

En la *Figura 5.1* se presenta un escenario a modo de ejemplo donde se aplica la definición de esta regla. La *Figura 5.1(a)* corresponde al escenario condicional que, al aplicarle la regla de puntos sin matching, genera el antiescenario de la *Figura 5.1(b)*. Sobre este último el traductor definido por el algoritmo de tableau genera el autómata temporizado de la *Figura 5.1(c)*. En el autómata se puede apreciar que no basta con determinar que el estado de aceptación ($ACCEPT$) es alcanzable, sino que se debe comprobar también que no ocurra ningún evento b que termine en el estado trampa ($TRAP$).



(a) Escenario Condicional (b) Antiescenario $\mathcal{N}^{\mathcal{S}_1 \not\rightarrow \mathsf{p2}}$ (c) Autómata observador

Figura 5.1: Ejemplo de verificación de escenario con regla de "puntos sin matching"

Esta regla, en gran parte de los casos, se puede mejorar para que evitar la necesidad de verificación por "Liveness". La estrategia consiste en determinar si el punto que se quiere negar es precedente (no necesariamente inmediato) de algún punto que forme parte del escenario antecedente. En este caso se construye un antiescenario que en lugar de emplear el elemento *fin* utiliza este otro punto. A continuación se define la regla para "puntos sin matching" que contempla esta optimización.

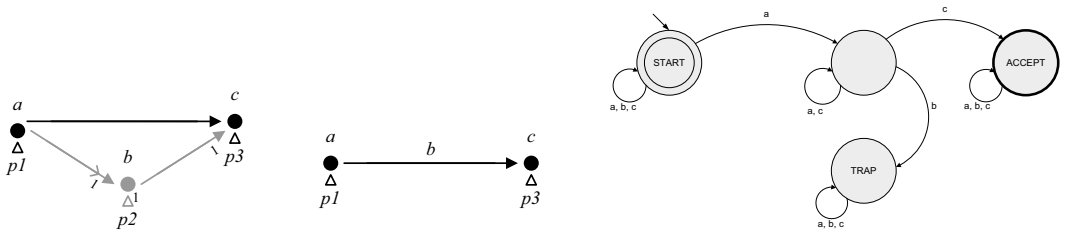
Definición 5.1.1 (Clausura transitiva de Precedencia)**.** Dados dos puntos p y h en un escenario se define que $\mathsf{p} <^{+} \mathsf{h}$ sii:

1. o bien $(\mathsf{p}, \mathsf{h}) \in <$
2. o existe un punto r tal que $(\mathsf{p}, \mathsf{r}) \in <$ y $\mathsf{r} <^{*} \mathsf{h}$.

Definición 5.1.2 (Antiescenarios para puntos sin matching –optimizada–)**.** Dados dos escenarios $\mathcal{S}_1$ y $\mathcal{S}_2$ tal que $\mathcal{S}_2 <:: \mathcal{S}_1$ por un ranking $\lessdot$ sobre P_2, y un punto concreto $\mathsf{p} \in P_2 \setminus P_1$, se construye el escenario "p *sin matching*" (denominado $\mathcal{N}^{\mathcal{S}_1 \not\rightsquigarrow \mathsf{p}}$) como:

- $\mathcal{N}^{\mathcal{S}_1 \not\rightsquigarrow \mathsf{p}} = \mathcal{P}^{\mathcal{S}_1 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}}$, donde $\{\mathsf{q}\} = ld_{\lessdot}(\mathsf{p})$ y
  - cuando $\mathsf{p} <_2 \mathsf{q}$:
    $\mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}} = \langle \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{q}\}, \{(\mathsf{q}, \ell_2(\mathsf{q}))\}, \emptyset, \{(\mathbf{0}, \mathsf{q})\}, \emptyset, \emptyset, \{(\mathbf{0}, \mathsf{q}, \ell_2(\mathsf{p}))\}, \emptyset \rangle$
  - cuando $\mathsf{q} <_2 \mathsf{p}$
    - si $\nexists$ un punto h donde $\mathsf{h} \in \mathcal{S}_1$ y $\mathsf{p} <_2^{+} \mathsf{h}$:
      $\mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}} = \langle \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}), \{\mathsf{q}\}, \{(\mathsf{q}, \ell_2(\mathsf{q}))\}, \emptyset, \{(\mathsf{q}, \infty)\}, \emptyset, \emptyset, \{(\mathsf{q}, \infty, \ell_2(\mathsf{p}))\}, \emptyset \rangle$
    - si $\exists$ un punto h donde $\mathsf{h} \in \mathcal{S}_1$ y $\mathsf{p} <_2^{+} \mathsf{h}$:
      $\mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}} = \langle \ell_2(\mathsf{p}) \cup \ell_2(\mathsf{q}) \cup \ell_2(\mathsf{h}), \{\mathsf{q}, \mathsf{h}\}, \{(\mathsf{q}, \ell_2(\mathsf{q})), (\mathsf{h}, \ell_2(\mathsf{h}))\}, \emptyset, \{(\mathsf{q}, \mathsf{h})\}, \emptyset, \emptyset, \{(\mathsf{q}, \mathsf{h}, \ell_2(\mathsf{p}))\}, \emptyset \rangle$
      Es este último caso el que se incorpora en la definición para contemplar la optimización.

En la *Figura 5.2* se presenta un escenario a modo de ejemplo donde se aplica la regla optimizada. La *Figura 5.2(a)* corresponde al escenario condicional que, al aplicarle la nueva regla de puntos sin matching, genera el antiescenario de la *Figura 5.2(b)* donde en lugar de utilizar el elemento *fin* se emplea el punto p3 el cual es posterior a p2. La *Figura 5.2(c)* es el autómata temporizado para el antiescenario donde se puede apreciar que a partir del estado de aceptación ($ACCEPT$) ya no se puede avanzar a otro estado simplificando la condición de verificación para el modelchecker.

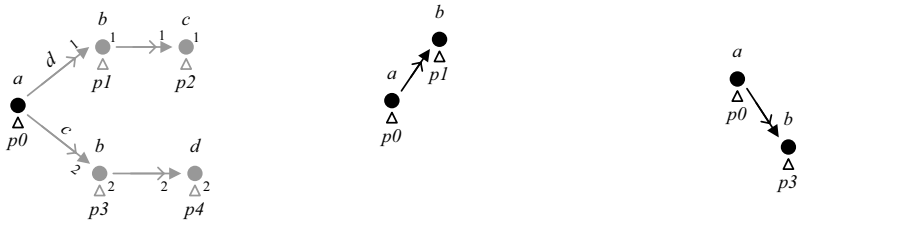


(a) Escenario Condicional (b) Antiescenario $\mathcal{N}^{\mathcal{S}_1 \not\rightsquigarrow \mathsf{p2}}$ (c) Autómata observador

Figura 5.2: Ejemplo de verificación de escenario con regla de "puntos sin matching" optimizada

5.2 Intersección entre múltiples consecuentes

Cuando se trabaja con escenarios condicionales es posible que entre los consecuentes –no necesariamente todos– exista un *subpatrón* en común que extienda al antecedente. La *Figura 5.3* presenta un ejemplo de un escenario condicional con un subpatrón común entre ambos consecuentes. En este caso, el escenario que representa el camino a p1 (del consecuente 1), *Figura 5.3(b)*, y el escenario que representa el camino a p2 (del consecuente 2), *Figura 5.3(c)*, corresponden al mismo subpatrón común.

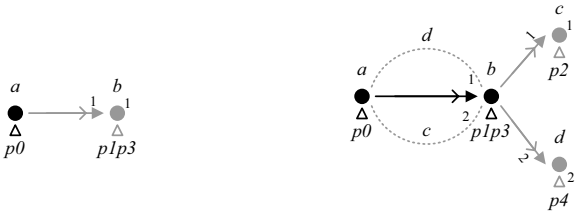


(a) Escenario Condicional (b) Escenario que representa el camino a p1 (c) Escenario que representa el camino a p3

Figura 5.3: ECD con subpatrón común entre los consecuentes

Al aplicar para este ejemplo el algoritmo de construcción de antiescenarios, se generan –entre otros– el antiescenario p1 *sin matching* (para el consecuente 1), y el antiescenario p3 *sin matching* (para el consecuente 2), resultando ambos antiescenarios equivalentes. Luego, cada uno de estos dos antiescenarios se combina con los antiescenarios del consecuente alternativo para la construcción de los antiescenarios finales. Sin embargo, ambos antiescenarios, que son equivalentes, ya representan un único antiescenario final, y en consecuencia en estos casos es posible reducir la cantidad de antiescenarios finales.

Para tratar estos casos, nuestra propuesta consiste en identificar manualmente el subpatrón común que extiende al antecedente (*Figura 5.4(a)*). A continuación, se procede con la verificación de este escenario condicional. Si no se verifica entonces se puede determinar que el escenario condicional original no se satisface, de lo contrario se procede con la verificación del escenario condicional resultante de llevar al subpatrón común al antecedente (*Figura 5.4(b)*). De esta manera, en este ejemplo, donde para el escenario original el algoritmo construye 9 antiescenarios finales, con la mejora se reducen a 5 antiescenarios finales, siendo 1 para el escenario de *Figura 5.4(a)* y 4 para el escenario de la *Figura 5.4(b)*.



(a) Escenario Condicional para el subpatrón común (b) Escenario Condicional con el subpatrón común en el antecedente

Figura 5.4: Escenarios condicionales a verificar para el ejemplo de la *Figura 5.3*

5.3 Antiescenarios imposibles

Para los escenarios condicionales, el número de consecuentes en la práctica es usualmente pequeño (frecuentemente un consecuente para operaciones regulares y otro consecuente para el error o representación de circunstancias excepcionales). Sin embargo, como el número de escenarios finales es el producto de la cantidad de antiescenarios de cada consecuente, en la medida que crece la cantidad de consecuentes del escenario condicional, el número de antiescenarios finales que se generan crece rápidamente. Afortunada-

mente, es frecuente el caso donde la fusión de antiescenarios genera escenarios imposibles, los cuales pueden ser descartados antes de alcanzar la fase de model-checking por medio de un análisis estático eficiente.

Los escenarios imposibles son escenarios correctos sintácticamente pero también contradictorios y por esto no se pueden satisfacer (independientemente del modelo del sistema); por ejemplo *Figura 4.6(a)* y *Figura 4.6(b)*. Cuando el algoritmo de tableau es alimentado con un escenario de este tipo, que presenta una relación de precedencia cíclica o inconsistencias entre precedencias y restricciones de eventos prohibidos, éste genera autómatas observadores donde el estado de aceptación es inalcanzable. Utilizando esta propiedad, los antiescenarios (no necesariamente finales) que son imposibles pueden ser detectados y descartados por el algoritmo de construcción de antiescenarios. Si bien sería posible realizar el análisis sobre el mismo escenario, el mecanismo propuesto es eficiente. Esta mejora resulta considerablemente significativa para aquellos casos donde los escenarios condicionales constan de múltiples consecuentes. Por ejemplo, para un escenario condicional con 3 consecuentes, donde se generan 4 antiescenarios para cada consecuente, se construyen 64 antiescenarios finales. Si se detectase tempranamente que la mitad de las fusiones realizas resultan en escenarios imposibles, entonces esta cantidad se reduciría a sólo 8 antiescenarios. Es por este motivo que se decidió incluir esta mejora en la herramienta que implementa el algoritmo de construcción de antiescenarios.

Capítulo 6

Casos de estudio

En esta sección se presentan una serie de ejemplos, con el objetivo de mostrar en detalle como es el mecanismo de verificación de los escenarios condiciones. Cabe aclarar que estos casos abarcan la totalidad de las reglas del algoritmo de generación de antiescenarios.

6.1 Autorización

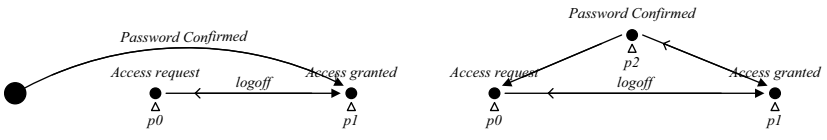
Este caso corresponde al escenario de la *Figura 3.1*. Si bien en la introducción al algoritmo ya adelantamos parte de los antiescenarios de este ejemplo, ahora se expone el conjunto completo de los antiescenarios indicando para cada uno de estos la regla del algoritmo que le da origen.

El ranking de puntos empleado para cada consecuente es:

$\mathcal{S}_1$: {p0, p1, p2}

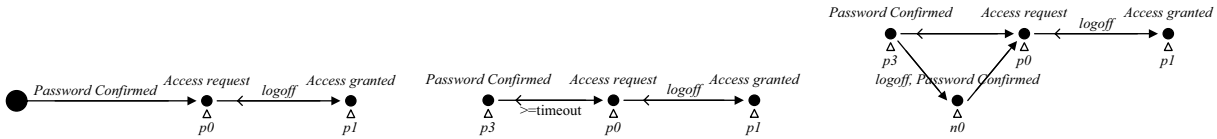
$\mathcal{S}_2$: {p0, p1, p3}

Aplicando las reglas del algoritmo se generan los antiescenarios para los consecuentes $\mathcal{S}_1$ y $\mathcal{S}_2$ que se detallan en la *Figura 6.1* y la *Figura 6.2* respectivamente. Luego, para construir de los antiescenarios finales se fusionan los antiescenarios de ambos consecuentes. La *Figura 6.3* presenta el resultado, donde los antiescenarios finales *6.3(b)*, *6.3(c)* y *6.3(d)* son imposibles. Por lo tanto, en la etapa del modelchecker sólo hace falta verificar los antiescenarios *6.3(a)*, *6.3(e)* y *6.3(f)*.



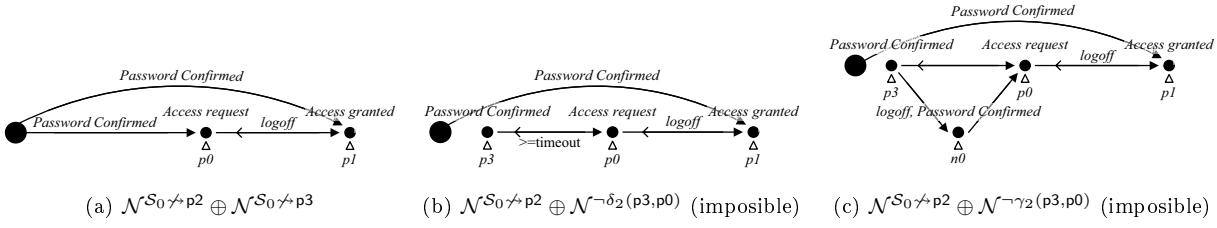
(a) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p2}}$ (b) $\mathcal{N}^{\mathsf{p0} \not\prec_1 \mathsf{p2}}$

Figura 6.1: Antiescenarios del consecuente $\mathcal{S}_1$.

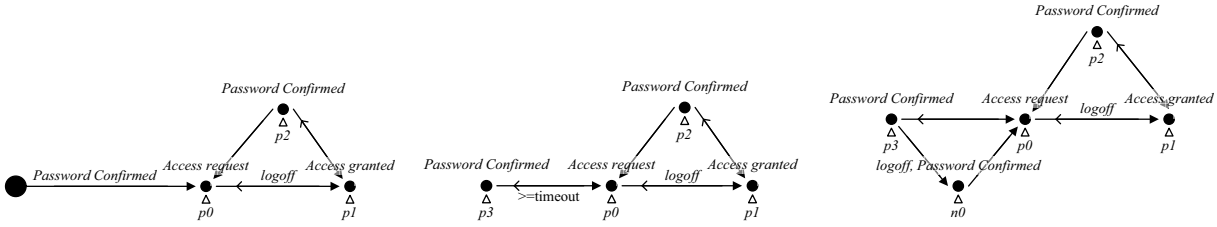


(a) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p3}}$ (b) $\mathcal{N}^{\neg\delta_2(\mathsf{p3},\mathsf{p0})}$ (c) $\mathcal{N}^{\neg\gamma_2(\mathsf{p3},\mathsf{p0})}$

Figura 6.2: Antiescenarios del consecuente $\mathcal{S}_2$.



(a) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p2}} \oplus \mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p3}}$ (b) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p2}} \oplus \mathcal{N}^{\neg\delta_2(\mathsf{p3},\mathsf{p0})}$ (imposible) (c) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p2}} \oplus \mathcal{N}^{\neg\gamma_2(\mathsf{p3},\mathsf{p0})}$ (imposible)



(d) $\mathcal{N}^{\mathsf{p0} \not<_1 \mathsf{p2}} \oplus \mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p3}}$ (imposible) (e) $\mathcal{N}^{\mathsf{p0} \not<_1 \mathsf{p2}} \oplus \mathcal{N}^{\neg\delta_2(\mathsf{p3},\mathsf{p0})}$ (f) $\mathcal{N}^{\mathsf{p0} \not<_1 \mathsf{p2}} \oplus \mathcal{N}^{\neg\gamma_2(\mathsf{p3},\mathsf{p0})}$

Figura 6.3: Antiescenarios finales

6.2 Comienzo y fin

Este caso corresponde al escenario de la *Figura 3.4* para el cual hemos adelantamos parte de los antiescenarios en la introducción del algoritmo. En la *Figura 6.4* se presentan todos los antiescenario para este ejemplo, empleando el ranking de puntos definido por {p2, p5, p3, p4, p1}. A continuación se relaciona cada una de las restricciones que incorpora el consecuente con las reglas que definen los antiescenarios.

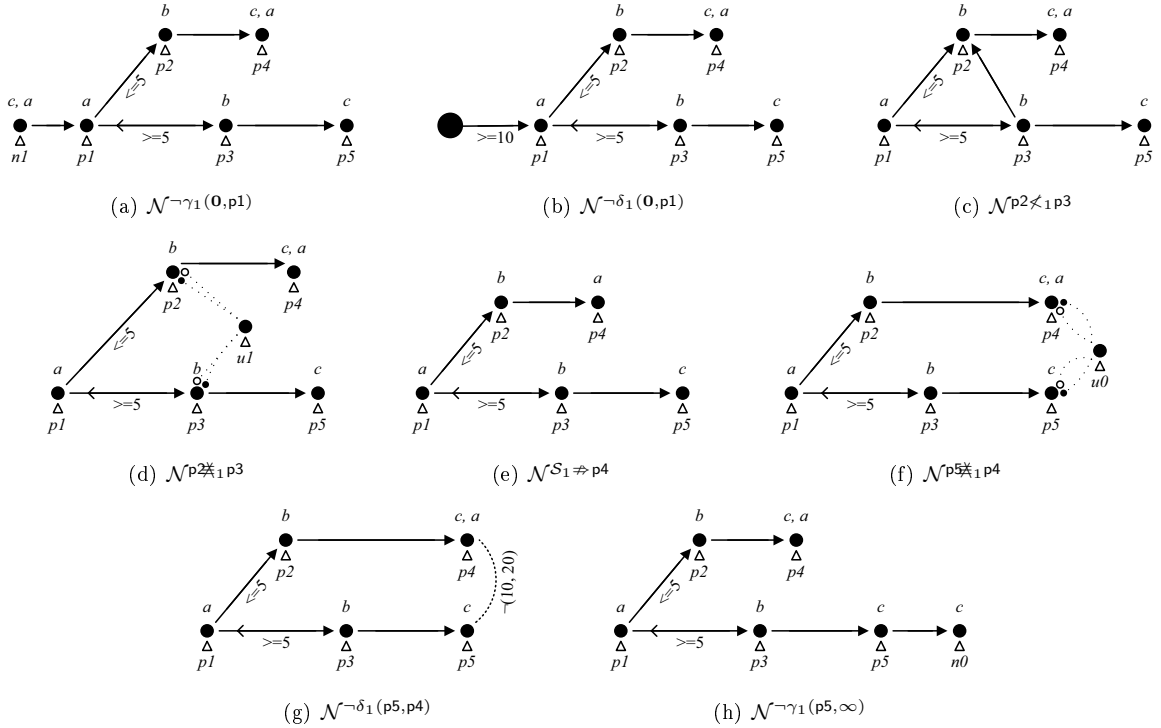
- desde el comienzo hasta p1 no ocurre ningún evento c y a. La negación de esta restricción es el antiescenario donde antes de p1 ocurre una evento c y a. El antiescenario que representa esta situación se construye por la regla $\mathcal{N}^{\neg\gamma_1(\mathbf{0},\mathsf{p1})}$ de la *Figura 6.4(a)*.

- desde el comienzo hasta p1 transcurren menos de 10 unidades de tiempo. La negación de esta restricción es el antiescenario donde hasta p1 transcurren 10 o más unidades de tiempo. El antiescenario que representa esta situación se construye por la regla $\mathcal{N}^{\neg\delta_1(\mathbf{0},\mathsf{p1})}$ de la *Figura 6.4(b)*.

- entre p2 y p3 hay una precedencia. La negación de esta restricción es el antiescenario donde o bien p2 y p3 corresponden al mismo evento de la ejecución, o y p2 ocurre luego de p2. Para el primer caso el antiescenario se construye por la regla $\mathcal{N}^{\mathsf{p2} \not\leftrightarrows_1 \mathsf{p3}}$ correspondiente a la *Figura 6.4(c)*, mientras que para el segundo caso el antiescenario se construye por la regla $\mathcal{N}^{\mathsf{p2} \not\leftrightarrows_1 \mathsf{p3}}$ correspondiente a la *Figura 6.4(d)*.

- para p4 el evento que ocurre es necesariamente el evento c. La negación de esta restricción es el antiescenario donde el evento que ocurre para p4 es un evento que permite el antecedente pero que esta restringuido en el consecuente, en este caso el evento a. El antiescenario que representa esta situación se construye por la regla $\mathcal{N}^{\mathcal{S}_1 \not\Rightarrow \mathsf{p4}}$ de la *Figura 6.4(e)*.

- p4 y p5 corresponde a distintos eventos de la ejecución. La negación de esta restricción es el antiescenario donde p4 y p5 corresponden al mismo evento de la ejecución. El antiescenario se construye por la regla $\mathcal{N}^{\mathsf{p5} \not\asymp_1 \mathsf{p4}}$ correspondiente a la *Figura 6.4(f)*.
- entre p4 y p5 transcurren entre más de 10 y menos de 20 unidades de tiempo. La negación de esta restricción es el antiescenario donde entre p4 y p5 transcurren o hasta 10 o más de 20 unidades de tiempo. El antiescenario se construye por la regla $\mathcal{N}^{\neg\delta_1(\mathsf{p5},\mathsf{p4})}$ correspondiente a la *Figura 6.4(g)*.
- luego de p5 no ocurre ningún evento *c*. La negación de esta restricción es el antiescenario donde luego de p5 ocurre un evento *c*. El antiescenario se construye por la regla $\mathcal{N}^{\neg\gamma_1(\mathsf{p5},\infty)}$ correspondiente a la *Figura 6.4(h)*.



(a) $\mathcal{N}^{\neg\gamma_1(\mathbf{0},\mathsf{p1})}$ (b) $\mathcal{N}^{\neg\delta_1(\mathbf{0},\mathsf{p1})}$ (c) $\mathcal{N}^{\mathsf{p2} \not\preceq_1 \mathsf{p3}}$

(d) $\mathcal{N}^{\mathsf{p2} \not\asymp_1 \mathsf{p3}}$ (e) $\mathcal{N}^{\mathcal{S}_1 \not\Rightarrow \mathsf{p4}}$ (f) $\mathcal{N}^{\mathsf{p5} \not\asymp_1 \mathsf{p4}}$

(g) $\mathcal{N}^{\neg\delta_1(\mathsf{p5},\mathsf{p4})}$ (h) $\mathcal{N}^{\neg\gamma_1(\mathsf{p5},\infty)}$

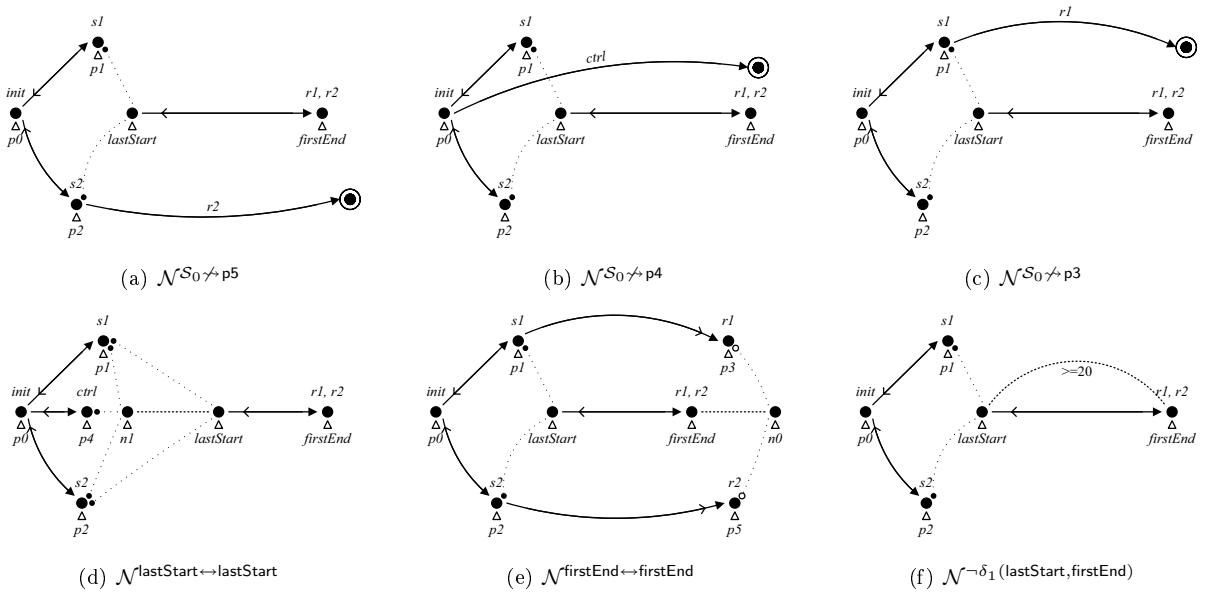
Figura 6.4: Antiescenarios generados

6.3 Representativos

Este caso corresponde al escenario que hemos presentado en la *Figura 4.1*. La *Figura 6.5* exhibe todos los antiescenario relacionados con este ejemplo por el algoritmo, que utilizó el ranking de puntos definido por {p0, firstEnd, lastStart, p2, p1, p5, p4, p3}.

El objetivo de este ejemplo es mostrar como funcionan las reglas de violación de aliasing en relación a los puntos representativo del consecuente. Para el punto representativo lastStart el consecuente incorpora la relación de representativo-último del punto p4. En este caso el antiescenario (*Figura 6.5(d)*) representa la situación donde el matching del lastStart actual difiera del matching del nuevo lastStart (punto n1) que incorpora la restricciones de representativo-último con p4. De esta manera, existe una violación de aliasing del punto lastStart si este antiescenario tiene un matching.

Respecto al punto representativo firstEnd – que en el antecedente es un punto concreto – el consecuente establece que es representativo-primero de los puntos p3 y p4. El antiescenario de la *Figura 6.5(e)* representa la posibilidad de que el matching del firstEnd actual difiera del matching del nuevo firstEnd (punto n0) que incorpora las restricciones de representativos del consecuente. Por lo tanto, existe una violación de aliasing del punto firstEnd si este antiescenario tiene un matching.



(a) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p5}}$ (b) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p4}}$ (c) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p3}}$

(d) $\mathcal{N}^{\mathsf{lastStart} \leftrightarrow \mathsf{lastStart}}$ (e) $\mathcal{N}^{\mathsf{firstEnd} \leftrightarrow \mathsf{firstEnd}}$ (f) $\mathcal{N}^{\neg \delta_1(\mathsf{lastStart},\mathsf{firstEnd})}$

Figura 6.5: Antiescenarios generados

6.4 Tautología

El escenario condicional de la *Figura 6.6* expresa que, si luego de un evento a, ocurre el evento c, donde entre ambos hay un único evento b, entonces o bien entre estos dos eventos a y c no ocurre otro evento a (consecuente 1), o entre a y b ocurre algún evento a (consecuente 2), o bien entre b y c ocurre algún evento a (consecuente 3).

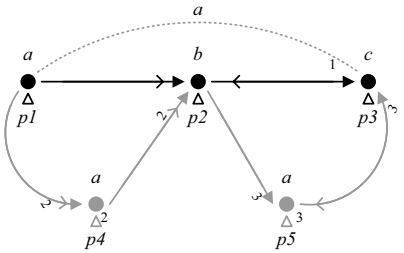


Figura 6.6: ECD $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3\} \rangle$

Este escenario se satisface si para cada matching del antecedente $\mathcal{S}_0$, entonces este matching es extensible para alguno de los consecuentes $\mathcal{S}_1$, $\mathcal{S}_2$ o $\mathcal{S}_3$. La particularidad de este escenario es que se trata de una tautología, es decir, que se satisface independientemente del modelo del sistema. Por lo tanto, luego de aplicar el algoritmo, no será necesaria la etapa posterior de verificación con modelchecker de los antiescenarios generados.

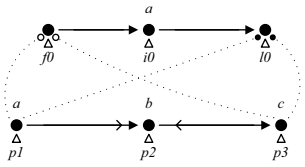
Las Figuras 6.7, 6.8 y 6.9 presentan los antiescenarios que generan las reglas para cada consecuente donde el ranking de puntos para la generación de los antiescenarios corresponde a:

$\mathcal{S}_1$: {p2, p3, p1}

$\mathcal{S}_2$: {p2, p3, p1, p4}

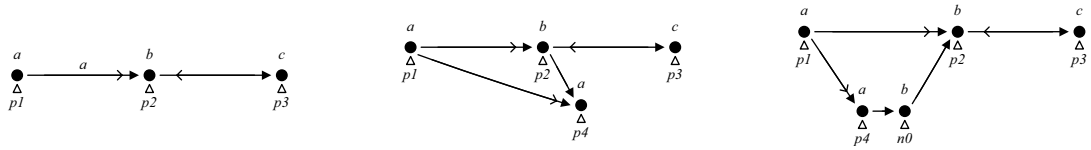
$\mathcal{S}_3$: {p2, p3, p1, p5}

Los antiescenarios de la *Figura 6.8(c)* y la *Figura 6.9(b)* son escenarios imposibles, por lo tanto, son descartados para la construcción de los antiescenarios finales.



(a) $\mathcal{N}^{\neg\gamma_1(\mathsf{p1},\mathsf{p3})}$

Figura 6.7: Antiescenarios del consecuente $\mathcal{S}_1$.



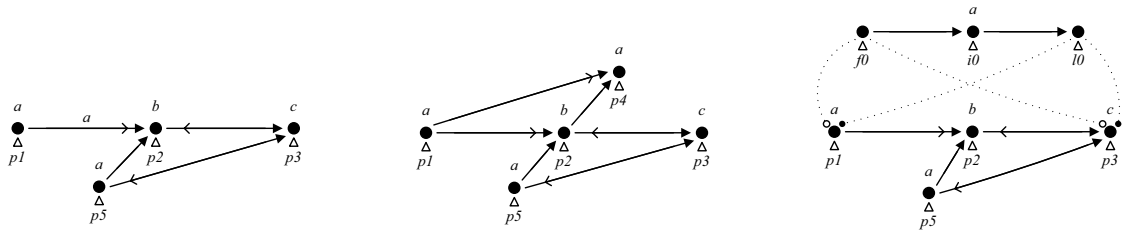
(a) $\mathcal{N}^{\mathcal{S}_0 \not\sim \mathsf{p4}}$ (optimizada a p2) (b) $\mathcal{N}^{\mathsf{p4} \not\lesssim_2 \mathsf{p2}}$ (c) $\mathcal{N}^{\neg\gamma_2(\mathsf{p4},\mathsf{p2})}$ (imposible)

Figura 6.8: Antiescenarios del consecuente $\mathcal{S}_2$.

(a) $\mathcal{N}^{\mathsf{p2} \not\lesssim_3 \mathsf{p5}}$ (b) $\mathcal{N}^{\mathcal{S}_0 \not\sim \mathsf{p5}}$ (imposible)

Figura 6.9: Antiescenarios del consecuente $\mathcal{S}_3$.

Para generar de los antiescenarios finales, se procede con la fusiones entre los antiescenarios de los distintos consecuentes. Las fusiones intermedias para los antiescenarios (no imposibles) de $\mathcal{S}_2$ y $\mathcal{S}_3$, es decir, *6.8(a)* $\oplus$ *6.9(a)* y *6.8(b)* $\oplus$ *6.9(a)*, se presentan en la *Figura 6.10(a)* y la *Figura 6.10(b)*. Como este último escenario es imposible, para construir los antiescenarios finales se fusiona *6.10(a)* con *6.7(a)*, el cual es el único antiescenario de $\mathcal{S}_1$. La *Figura 6.10(c)* presenta el antiescenario resultante que es imposible. Por lo tanto, según el *Teorema 4.2.7*, el escenario condicional de la tautología se satisface.



(a) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p4}} \oplus \mathcal{N}^{\mathsf{p2} \not<_3 \mathsf{p5}}$ (b) $\mathcal{N}^{\mathsf{p4} \not<_2 \mathsf{p2}} \oplus \mathcal{N}^{\mathsf{p2} \not<_3 \mathsf{p5}}$ (imposible) (c) $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p4}} \oplus \mathcal{N}^{\mathsf{p2} \not<_3 \mathsf{p5}} \oplus \mathcal{N}^{\neg\gamma_1(\mathsf{p1},\mathsf{p3})}$ (imposible)

Figura 6.10: Construcción de antiescenarios finales

6.5 Sensor remoto

El caso de estudio del Sensor Remoto[ABKO04] es una abstracción de un sistema de control industrial de tiempo real distribuido. Básicamente el sistema consiste de una componente central y dos sensores remotos ($Sensorx$ con x=1,2). Los sensores periódicamente miden un conjunto de variables ambientales (evento $SampleVx$) y almacenan los valores en la memoria compartida (eventos $StoreVx$). Cuando la componente central necesita estos valores, esta envía una señal a ambos sensores (evento $RqstSent$). Cuando la señal se recibe (evento $RqstArrivesAtSx$) esta tarea es lanzada (evento $UpRx$), se lee el último valor almacenado de la memoria compartida (evento $ReadVx$) y lo envía devuelta a la componente central (evento $VxSent$). Al arribo de ambos datos (los eventos $V1Arrives$ y $V2Arrives$) una tarea esporádica es lanzada (evento $UpPair$). Esta tarea aparea las lecturas (evento $V1V2Paired$) para que otro proceso pueda utilizar esta información para llevar a cabo ciertas acciones.

La *Figura 6.11* presenta la arquitectura del modelo. En este caso el modelo comprende 12 autómatas temporales (uno para cada componente del diseño) que interactúan asincrónicamente.

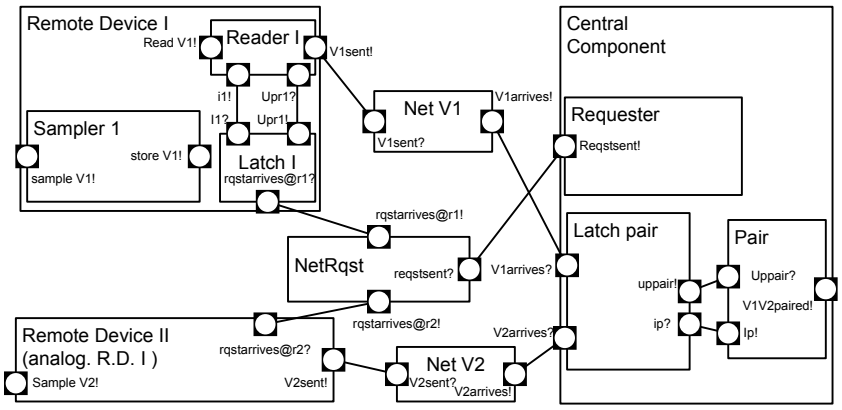


Figura 6.11: Sensor Remoto: arquitectura del modelo

El escenario condicional de la *Figura 6.12* es un patrón con el objetivo de determinar que todo par de muestras recibidas por la componente central, se originan por el mismo pedido de solicitud. En este caso, el escenario antecedente es el patrón que modela el arribo de ambas muestras a la componente central, pasando por las distintas tareas de medición, almacenamiento, lectura y envío hasta llegar a la componente central. Mientras que el escenario condicional exige adicionalmente que las lecturas de ambas variables sea en respuesta a un mismo pedido del procesador central (eventos $RqstSent$, $RqstArriveAtSx$ y $Uprx$).

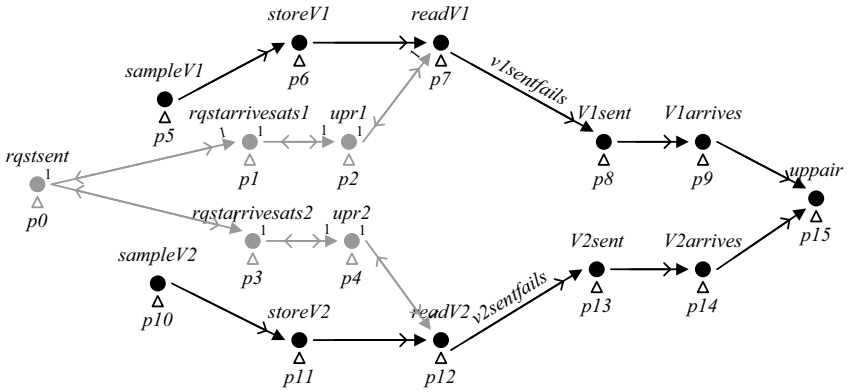
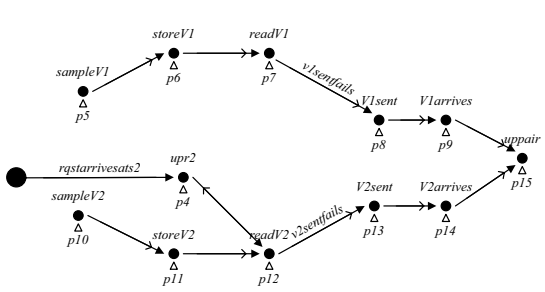
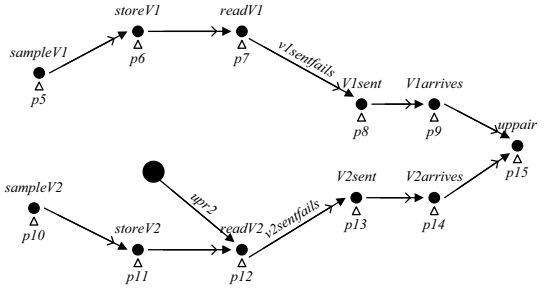


Figura 6.12: Escenario para el requerimiento de sincronización de respuestas

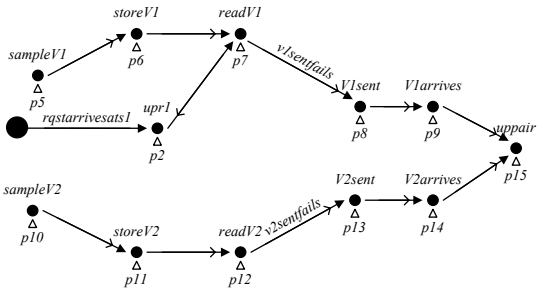
Las *Figuras 6.13, 6.14 y 6.15* presentan todos los antiescenarios generados por la herramienta que implementa el algoritmo con las reglas. Cada uno de estos antiescenarios presenta una manera en la cual el escenario condicional podría no cumplirse en relación al escenario antecedente. El ranking de puntos utilizado para la generación de los antiescenarios es: {p15, p14, p13, p12, p11, p10, p9, p8, p7, p6, p5, p4, p3, p2, p1, p0}.



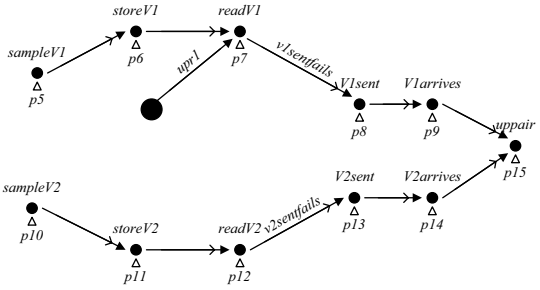
(a) AE1 $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p3}}$



(b) AE2 $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p4}}$

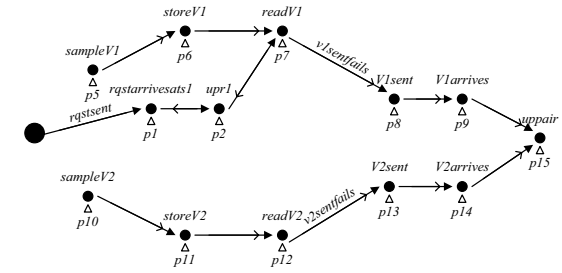


(c) AE3 $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p1}}$

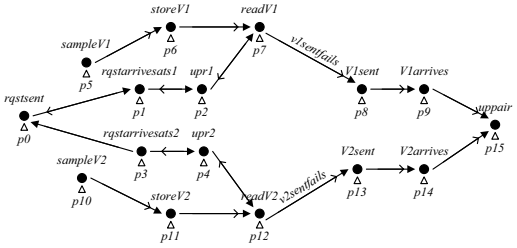


(d) AE4 $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p2}}$

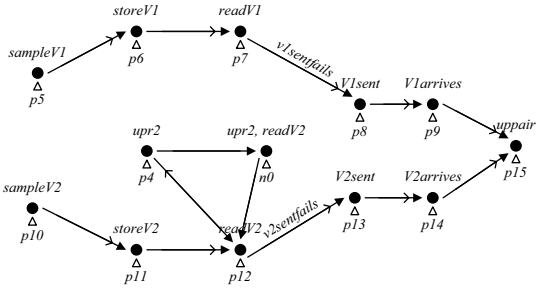
Figura 6.13: Antiescenarios del Escenario Condicional (parte 1)



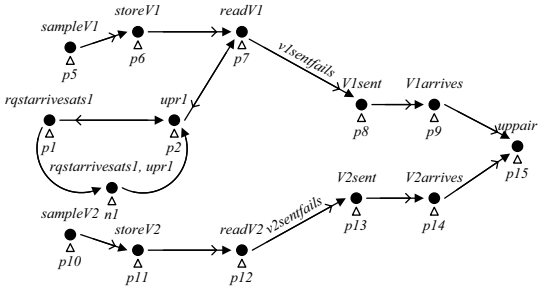
(a) AE5 $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p0}}$



(b) AE6 $\mathcal{N}^{\mathsf{p0} \not\leq_2 \mathsf{p3}}$

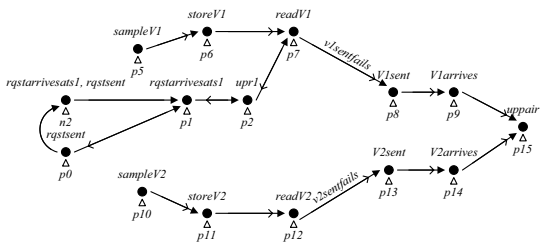


(c) AE7 $\mathcal{N}^{\neg\gamma_1(\mathsf{p4},\mathsf{p12})}$

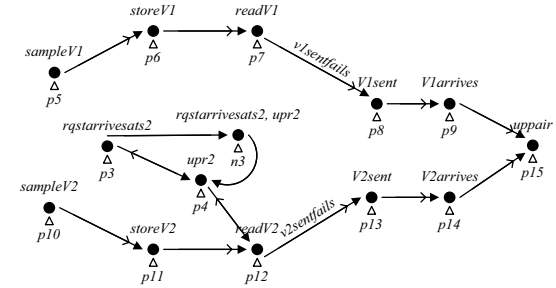


(d) AE8 $\mathcal{N}^{\neg\gamma_1(\mathsf{p1},\mathsf{p2})}$

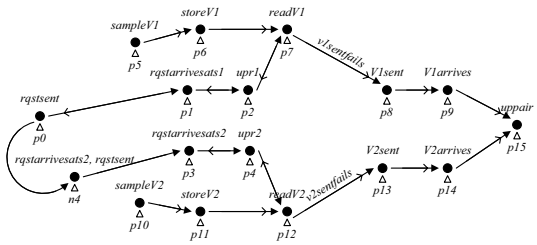
Figura 6.14: Antiescenarios del Escenario Condicional (parte 2)



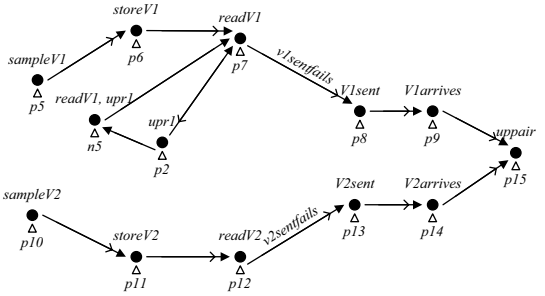
(a) AE9 $\mathcal{N}^{\neg\gamma_1(\mathsf{p0},\mathsf{p1})}$



(b) AE10 $\mathcal{N}^{\neg\gamma_1(\mathsf{p3},\mathsf{p4})}$



(c) AE11 $\mathcal{N}^{\neg\gamma_1(\mathsf{p0},\mathsf{p3})}$



(d) AE12 $\mathcal{N}^{\neg\gamma_1(\mathsf{p2},\mathsf{p7})}$

Figura 6.15: Antiescenarios del Escenario Condicional (parte 3)

La *Tabla 6.1* resume los resultados de la generación de los antiescenarios y de la verificación de los mismo usando una versión de la herramienta Uppaal 4.0.2 sobre Windows en una PC AMD Athlon XP +2000 1.26 Mhz con 256 MB. Se detallan los tiempos empleados del generador de antiescenarios, del traductor de *Tableau* y el tamaño del autómata observador, como así también los tiempos y resultados de la verificación con el modelchecker.

Generador de antiescenarios: #antiescenarios = 12 y tiempo = 3 seg.					
Antiescenario	traductor *Tableau*			Verificación modelchecker	
generado	Tiempo (seg.)	#Locaciones	#Aristas	Tiempo (seg.)	se satisface?
1: $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p3}}$	1,75	147	2.302	11,60	No
2: $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p4}}$	0,89	48	686	9,46	No
3: $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p1}}$	1,67	147	2.302	10,13	No
4: $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p2}}$	0,78	48	686	8,88	No
5: $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p0}}$	2,17	194	3.262	9,14	No
6: $\mathcal{N}^{\mathsf{p0} \not<_2 \mathsf{p3}}$	3,76	792	15.271	63,76	Sí
7: $\mathcal{N}^{\neg\gamma_1(\mathsf{p4},\mathsf{p12})}$	1,91	246	3.730	31,41	Sí
8: $\mathcal{N}^{\neg\gamma_1(\mathsf{p1},\mathsf{p2})}$	3,11	340	5.553	26,55	Sí
9: $\mathcal{N}^{\neg\gamma_1(\mathsf{p0},\mathsf{p1})}$	3,23	434	7.606	36,09	Sí
10: $\mathcal{N}^{\neg\gamma_1(\mathsf{p3},\mathsf{p4})}$	2,12	340	5.552	47,08	Sí
11: $\mathcal{N}^{\neg\gamma_1(\mathsf{p0},\mathsf{p3})}$	5,31	1.032	20.239	53,12	Sí
12: $\mathcal{N}^{\neg\gamma_1(\mathsf{p2},\mathsf{p7})}$	1,48	246	3.721	64,53	Sí

Tabla 6.1: Resultados de la verificación

El resultado demostró que el requerimiento no se cumple porque existen antiescenarios (del 7 al 12) que podrían satisfacerse para alguna ejecución del sistema. Uno de estos antiescenarios es el de la *Figura 6.15(a)* que representa la situación donde el pedido es realizado (evento *rqstSent*) pero entre que esto sucede y éste llega al sensor 1 (evento *ReqArrivesAtS*1 del punto p_1) el sensor puede disparar el evento *ReqArrivesAtS*1 (punto n_2) al recibir un pedido de una solicitud enviada previamente.

Capítulo 7

Implementación

En esta sección se describen las herramientas que se implementaron para soportar el proceso de verificación de escenarios condicionales (ver la *Figura 1.5*). Como mencionamos anteriormente, la implementación tiene dos partes:

- la extensión de la herramienta visual VTS para soportar escenarios condicionales
- y la herramienta que implementa el algoritmo de generación de antiescenarios.

7.1 Front-end VTS

Esta herramienta, desarrollada con un plug-in de `Microsoft Visio`, se extendió para soportar la notación gráfica y definición de los escenarios VTS condicionales. Mediante el front-end VTS el diseñador puede especificar gráficamente los escenarios condicionales relacionados con los requerimientos, contando al mismo tiempo con la ayuda de las validaciones sintácticas que proporciona la herramienta. La *Figura 7.1* presenta una pantalla de ejemplo en la cual se modela un escenario condicional.

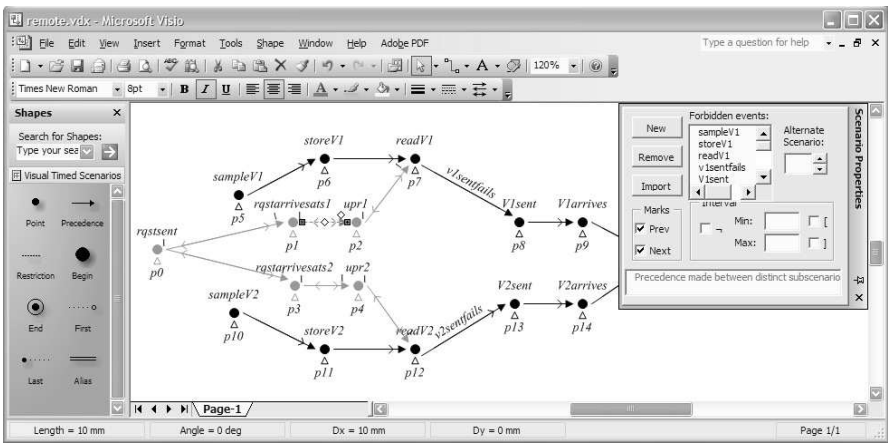


Figura 7.1: Front-end VTS para el diseño de escenarios

El escenario que se modela con la herramienta se persiste en un archivo de tipo `XML Drawing (*.vdx)` en el que se almacenan en formato XML, tanto la representación gráfica, como la definición formal del escenario. Esta definición cumple con la especificación del siguiente esquema DTD.

7.1.1 DTD del documento para definición de ECD

A continuación se presenta la estructura del documento XML para los escenarios condicionales (*ECD*). Esta definición es utilizada por el front-end para representar los escenarios condicionales, y por el algoritmo generador de antiescenarios para interpretarlos.

```
<?xml version="1.0" encoding="UTF-8"?>
```

<!-- *Un ECD tiene un identificador y esta conformado por un conjuntos de eventos, un conjunto de puntos, un conjunto de relaciones de precedencia (opcional), un conjunto de restricciones de desigualdad (opcional), un conjunto de relaciones de representativos (opcional). y un conjunto de consecuentes.*-->

```
<!ELEMENT dcscenario (events, points, precedence?, restrictions?, representatives?,
                      consequents)>
<!ATTLIST dcscenario id ID #IMPLIED>
```

<!-- *El conjunto de* ***consecuentes*** *es una secuencia no vacía de consecuentes donde cada uno de estos tiene un identificador y esta conformado por un conjunto de puntos, un conjunto de relaciones de precedencia (opcional), un conjunto de restricciones de desigualdad (opcional) y un conjunto de relaciones de representativos (opcional).*-->

```
<!ELEMENT consequents (consequent)+>
<!ELEMENT consequent (points, precedence?, restrictions?, representatives?)+>
<!ATTLIST consequent id ID #IMPLIED>
```

<!-- *El conjunto de* ***eventos*** *es una secuencia de eventos.*-->

```
<!ELEMENT events (event*)>
<!ELEMENT event (#PCDATA)>
```

<!-- *El conjunto de* ***puntos*** *es una secuencia de puntos. Cada uno de estos puntos tiene asociado un identificador y esta etiquetado con una secuencia de eventos (no vacía) donde* lambda *es un evento distinguido utilizado para representar un momento de la ejecución sin asignación de eventos.*-->

```
<!ELEMENT points (point*)>
<!ELEMENT point ((lambda | event)+)>
<!ELEMENT lambda EMPTY>
<!ATTLIST point id ID #REQUIRED>
```

<!-- *El conjunto de* ***precedencias*** *es una secuencia de precedencias donde cada una de éstas se define por un punto inicial o* **0** *(el comienzo de la ejecución), un punto destino o* ∞ *(el final de la ejecución), y opcionalmente una secuencia de eventos prohibidos y/o una restricción temporal. Adicionalmente se puede indicar si el tipo de flecha es* $\longleftrightarrow$, $\longrightarrow$ *o de ambos tipos, es decir,* `prev`, `next` *o* `both` *respectivamente.*-->

```
<!ELEMENT precedence (precedes*)>
<!ELEMENT precedes ((point-ref | before-start),
                    (point-ref | after-end),
                    (event)*, interval?)>
<!ATTLIST precedes marks (prev | next | both) #IMPLIED>
<!ELEMENT point-ref EMPTY>
<!ATTLIST point-ref id IDREF #IMPLIED>
<!ELEMENT before-start EMPTY>
<!ELEMENT after-end EMPTY>
```

<!-- *Cada* ***intervalo*** *tiene dos limites: uno inferior y otro inferior, donde cada uno de estos límites puede ser estricto o no. Adicionalmente el intervalo puede o no estar negado.*-->

```
<!ELEMENT interval (lower-bound?, upper-bound?)>
<!ATTLIST interval negated (true | false) #IMPLIED>
<!ELEMENT lower-bound EMPTY>
<!ATTLIST lower-bound value CDATA #REQUIRED included (true | false) #REQUIRED>
<!ELEMENT upper-bound EMPTY>
<!ATTLIST upper-bound value CDATA #REQUIRED included (true | false) #REQUIRED>
```

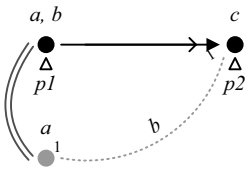
```
<!-- El conjunto de restricciones es una secuencia de restricciones. Una restricción se define de la
misma forma que una precedencia excepto que no tiene relevancia el orden entre el elemento inicial y
el final.-->

<!ELEMENT restrictions (restriction*)>
<!ELEMENT restriction ((point-ref | before-start | after-end),
                       (point-ref | before-start | after-end),
                       (event)*, interval?)>
<!ATTLIST restriction marks (prev | next | both) #IMPLIED>

<!-- El conjunto de relaciones de representativos es una secuencia de representativos-primeros
y/o representativos-últimos. Cada una de estos elementos se conforma por dos puntos, donde para
representa-tivos-primeros el representante se indica en el primer punto, y para representativos-últimos
se indica en el segundo punto.-->

<!ELEMENT representatives (first | last)*>
<!ELEMENT first (point-ref, point-ref)>
<!ELEMENT last (point-ref, point-ref)>
```

A continuación se presenta el documento XML para le escenario condicional de la *Figura 7.1.*



(a) Escenario condicional

```
<scenario id=''>
   <events>
      <event>a</event>
      <event>b</event>
      <event>c</event>
   </events>
   <points>
      <point id='p1'>
         <event>a</event>
         <event>b</event>
      </point>
      <point id='p2'>
         <event>c</event>
      </point>
   </points>
   <precedence>
      <precedes marks='next'>
         <point-ref id='p1'/>
         <point-ref id='p2'/>
      </precedes>
   </precedence>
   <consequents>
      <consequent id='1'>
         <points>
            <point id='p1'>
               <event>a</event>
            </point>
            <point id='p2'>
               <event>c</event>
            </point>
         </points>
         <restrictions>
            <restriction>
               <point-ref id='p1'/>
               <point-ref id='p2'/>
               <event>b</event>
            </restriction>
         </restrictions>
      </consequent>
   </consequents>
</scenario>
```

(b) Representación XML del escenario condicional

Figura 7.2: Ejemplo de un escenario condicional con su correspondiente definición XML

7.2 Generador de Antiescenarios

La implementación del algoritmo generador de antiescenarios se realizó en el lenguaje de programación Java, y está estrechamente relacionada con las definiciones empleadas en la especificación del algoritmo. La *Figura 7.3* presenta el diagrama de clases asociado con la representación de un escenario.

En la *Figura 7.4* se presenta el diagrama de las clases relacionado con el algoritmo. En particular, la clase `BuilderNegativeCScenario` se encarga de implementar el procedimiento para el algoritmo.

Los principales pasos de este procedimiento comprenden:

1. Inicialmente, para cada consecuente se verifica que esté es una especialización del antecedente y se determina un ranking $\preccurlyeq$ para la especialización determinística.
2. A continuación, para cada punto, relación de precedencia y de desigualdad del consecuente se determina que reglas aplican, y se construyen los antiescenarios correspondientes.

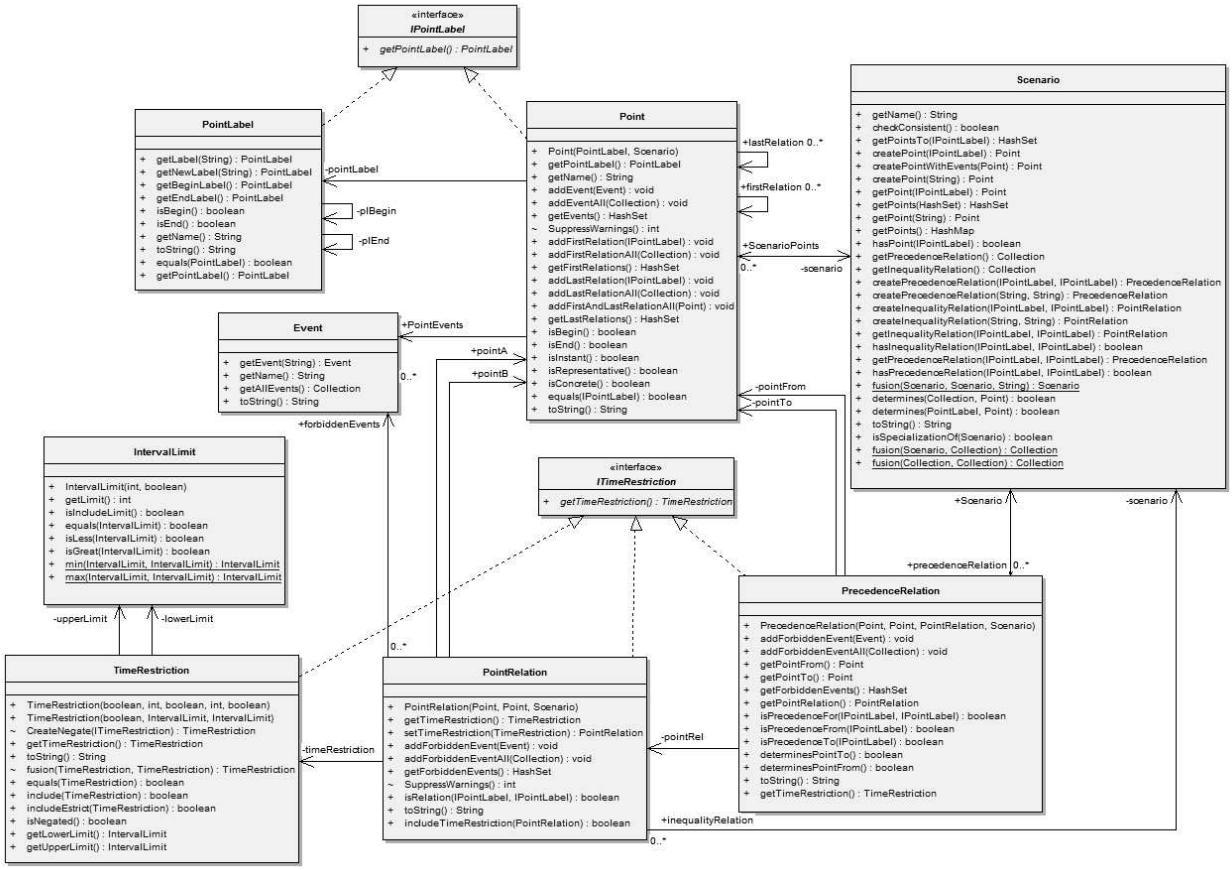


Figura 7.3: Diagrama de clases para la representación de escenarios

3. Luego, se generan los antiescenarios finales resultantes de las fusiones entre los antiescenarios de todos los consecuentes, descartando en este proceso todos los antiescenarios que generan antiescenarios imposibles (determinados por el algoritmo de *Tableau*).

4. Finalmente, para cada uno de estos antiescenarios finales, invocando al algoritmo de *Tableau*, se construye el autómata verificador.

De esta forma, una vez generados los autómatas temporizados de estos antiescenarios, el diseñador utilizará un Modelchecker para sistemas temporizados como Kronos y Uppaal con el objetivo de verificar si cada uno de estos antiescenarios se puede o no satisfacer, y concluir si el escenario condicional se verifica.

Para verificar la especialización determinística entre un consecuente $\mathcal{S}_1$ y el antecedente $\mathcal{S}_0$ ($\mathcal{S}_1 <:: \mathcal{S}_0$), y determinar un ranking $\lessdot$ que verifique la especialización se implementó el siguiente procedimiento:

1. Se verifican las condiciones de especialización entre ambos escenarios, es decir, para $\mathcal{S}_1 <: \mathcal{S}_0$.

2. sea $\lessdot$ una lista ordenada, se define:
   $\lessdot = P_0$ (no tiene relevancia el orden en que se agregan estos puntos).

3. Mientras se puedan agregar puntos a $\lessdot$.

   (a) para cada $\mathsf{p} \in P_1$ donde $(\mathsf{p} \notin \lessdot)$ y $(\lessdot \hookrightarrow \mathsf{p})$:
   $\lessdot = \lessdot + \mathsf{p}$

4. Si $P_1 \nsubseteq \lessdot$ no existe un ranking válido, y por lo tanto $\mathcal{S}_1$ no es una especialización determinística de $\mathcal{S}_0$, de lo contrario $\lessdot$ es un ranking que verifica $\mathcal{S}_1 <:: \mathcal{S}_0$.

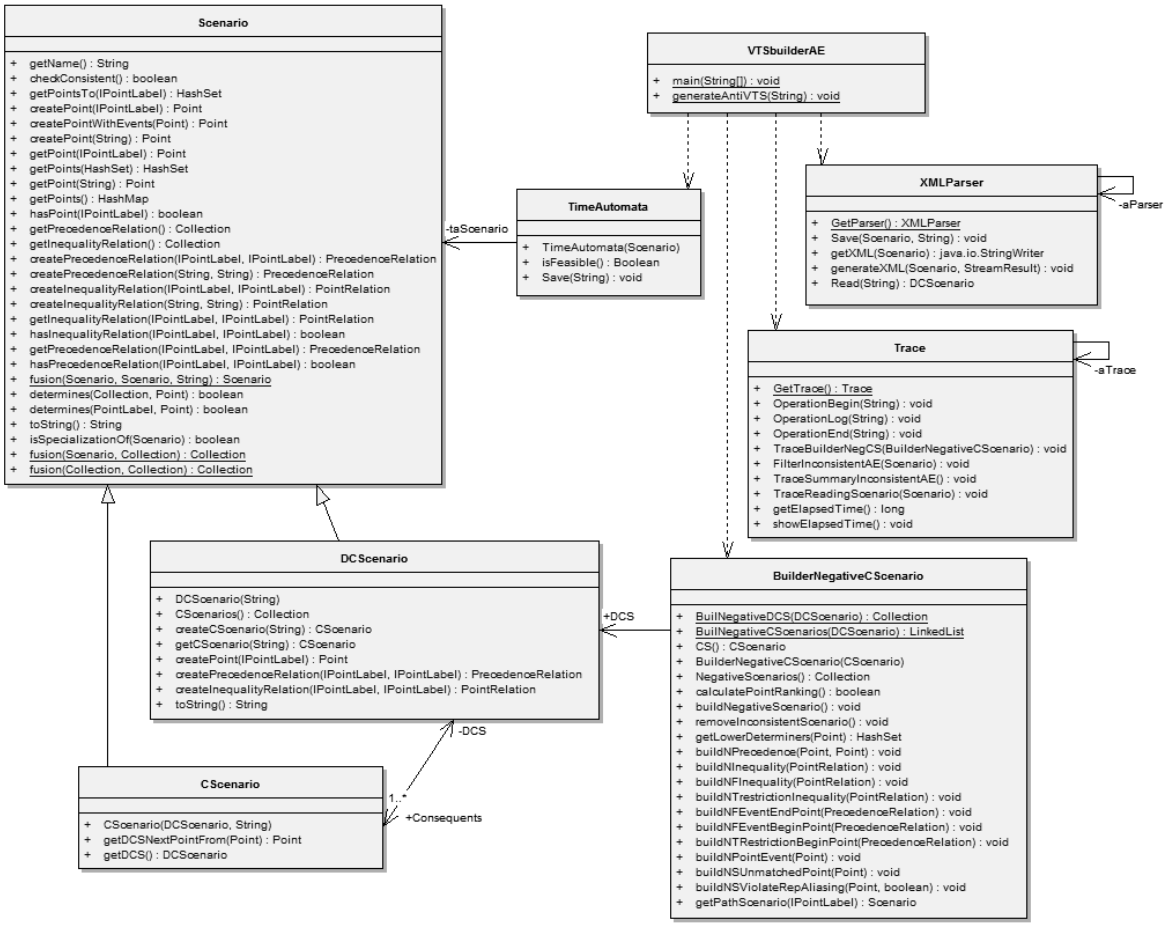


Figura 7.4: Diagrama de clases del algoritmo

7.2.1 Caso de estudio

El programa principal se denomina `VTSBuilderAE` y tiene los siguientes parámetros de invocación:

```
VTSbuilderAE (v1.0.1 - 20070401)
--------------------------------

Sintax: VTSbuilderAE <inputFile> [<outputSpec>] [options]

 <inputFile> is a Conditional VTS Escenario (XML-DCS generated with VTS Tool in MicrosoftVisio)
 <outputSpec> path and prefix for output files. If missing, is same as the input without extension

 options
   -r=t/(f)  : track Reading DCScenario
   -o=(t)/f  : track Operation
   -c=t/(f)  : generate CS AntiE (suffix _csN_ai.vts)
   -f=(t)/f  : generate Final AE (suffix _ae_i.vts)
   -uc=(t)/f : filter UnfeasibleCS AE (suffix _UnfeasibleCS_AE_i.vts)
   -ud=(t)/f : filter UnfeasibleDCS AE (suffix _UnfeasibleDCS_AE_i.vts)
   -a=t/(f)  : generate TimeAutomata (suffix _ae_i.tg)

Sample: VTSbuilderAE demo.vdx -r=t -o=f
```

A continuación, a modo de ejemplo, se presenta el resultado de la invocación de la herramienta para el caso de estudio de autorización (*Figura 3.1*).

```
Scenario Reading[...\VTS_Tool\JavaProject\VTSRun\pwd.vdx]
```

```
    File Containt: [scenario: null]
    Identify consequent[1]
    Identify consequent[2]
Scenario Readed[]
Generate NegativeCScenarios for DCS[] Begin

    Build NegativeCScenarios for CS[2] Begin
        calculatePointRanking for CS[2] Begin
            PointRanking: [00, p0, 0, p1, p3]
        calculatePointRanking for CS[2] End
        NegativeScenarios for CS[2] generate: 3 anti-e.
        Scenario Saving[2:UnmatchedPoint p3 (from Begin)]: ..._cs2_a1.vts
        Scenario Saved
        Scenario Saving[2:buildNFInequality p3-p0 (optimized by Prec.Rel.)]: ..._cs2_a2.vts
        Scenario Saved
        Scenario Saving[2:buildNTrestrictionInequality p3-p0]: ..._cs2_a3.vts
        Scenario Saved
    Build NegativeCScenarios for CS[2] End

    Build NegativeCScenarios for CS[1] Begin
        calculatePointRanking for CS[1] Begin
            PointRanking: [00, p0, 0, p1, p2]
        calculatePointRanking for CS[1] End
        NegativeScenarios for CS[1] generate: 2 anti-e.
        Scenario Saving[1:UnmatchedPoint p2 (from Begin)]: ..._cs1_a1.vts
        Scenario Saved
        Scenario Saving[1:buildNPrecedence p0-p2]: ..._cs1_a2.vts
        Scenario Saved
    Build NegativeCScenarios for CS[1] End
Generate NegativeCScenarios for [] End

Generate NegativeDCS (Merge Anti-e) Begin
Generate NegativeDCS End. Total Anti-e=6

Save[[2:UnmatchedPoint p3 (from Begin)] [1:UnmatchedPoint p2 (from Begin)]]: ..._ae_1.vts
Save[[2:UnmatchedPoint p3 (from Begin)] [1:buildNPrecedence p0-p2]]: ..._ae_2.vts
Save[[2:buildNFInequality p3-p0 (optimized by Prec.Rel.)] [1:UnmatchedPoint p2 (from Begin)]]: ..._ae_3.vts
Save[[2:buildNFInequality p3-p0 (optimized by Prec.Rel.)] [1:buildNPrecedence p0-p2]]: ..._ae_4.vts
Save[[2:buildNTrestrictionInequality p3-p0] [1:UnmatchedPoint p2 (from Begin)]]: ..._ae_5.vts
Save[[2:buildNTrestrictionInequality p3-p0] [1:buildNPrecedence p0-p2]]: ..._ae_6.vts
ElapsedTime:4172 miliseconds
```

Capítulo 8

Conclusiones, trabajos relacionados y trabajo futuro

8.1 Conclusiones

En este trabajo se presenta de forma completa el lenguaje VTS, el cual es una herramienta poderosa para la verificación de aplicaciones de tiempo real: tiene una sintaxis formal simple basada en ordenes parciales, una semántica subyacente declarativa y compacta basada en la idea intuitiva de matching sobre ejecuciones, y una semántica operacional basada en Autómatas Temporizados. Además los eventos no están limitados a ser eventos de comunicación o consecutivos; y la negación de la ocurrencia de un evento en un intervalo permite identificar los eventos próximos y previos de un determinado tipo. El concepto de eventos representativos y de instantes son características innovadoras, como así también los puntos de comienzo y fin (especialmente adecuado para negar el progreso). Asimismo, VTS trabaja con restricciones de tiempo real en un dominio de tiempo denso sin la utilización de relojes.

El objetivo de este trabajo se centra en la extensión de VTS para soportar escenarios condicionales. Se desarrolló el mecanismo de verificación para este tipo de escenarios el cual se vale de la construcción de escenarios negativos, expresados con escenarios VTS existenciales, para describir todos los casos en los cuales el escenario condicional podría no cumplirse. Estos antiescenarios, además de ser empleados para el proceso de verificación, de por si son un recurso útil para un ingeniero o diseñador al exhibir las alternativas posibles por las que el escenario condicional podría no cumplirse.

Se formalizó un conjunto de reglas para la generación de los antiescenarios y se propusieron variantes con el propósitos de optimizar casos específicos. Se presentaron las pruebas formales que establecen la corrección de las reglas y se definió el algoritmo que construye todos los antiescenarios, proporcionando ejemplos aplicados a casos de estudio.

También se extendió la notación visual para soportar la definición de los escenarios condicionales, se extendió el editor "front-end", desarrollado en Microsoft Visio, para permitir el diseño visual de escenarios condicionales. Finalmente se construyó una componente "back-end", desarrollada en Java, que implementa el algoritmo que construye los antiescenarios.

8.2 Trabajos relacionados

Existen varios trabajos cuyo objetivo es proveer herramientas y notaciones que faciliten y extiendan la verificación de sistemas, pero no demasiados de estos permiten expresar *triggered scenarios* como en VTS.

La herramienta TimeEdit[SHE01] es una herramienta que tiene el objetivo, al igual que VTS, de simplificar la tarea de expresar escenarios que nunca deberían ocurrir (en este caso, para las herramientas de verificación Spin y FeaVer). Algo similar se puede decir sobre una variante de GIL (*Graphical Interval Logic*) [MRK+97, ACD97] basada en eventos y tiempo real. Ambas herramientas TimeEdit y RT-GIL están basadas en diagramas de líneas temporizadas y no tienen la característica de ordenamiento parcial de eventos. TimeEdit no soporta restricciones temporales. RT-GIL comparte algunas características comunes con VTS. Provee operadores de búsqueda para localizar puntos finales de intervalos (similar a la noción de *próximo* y *previo* en VTS). Sin embargo el operador *previo* no puede aplicarse libremente como en VTS: el reconocimiento de intervalos comienza siempre desde un punto genérico (o el primero) en el intervalo abarcado. Por eso, por ejemplo, no es posible expresar restricciones de tipo *freshness*[1], de correlación o detectar la existencia de eventos que sucedieron en general, los cuales son fácilmente expresables en VTS. Para obtener *triggered scenarios* en RT-GIL, la implicación lógica puede utilizarse mediante subfórmulas que se apilan. A pesar de esto, el antecedente necesita ser reiterado en cada consecuente y este debe ser determinístico para asegurar que se detecte el intervalo correcto. GIL puede expresar restricción de duración de intervalos pero éstos están limitados a eventos consecutivos.

Quizás el ejemplo más distinguido y difundido de un formalismo visual para especificaciones basadas en escenarios es *Message Sequence Charts* (MSCs) [IT00]. A diferencia de VTS, los MSCs no son utilizados para describir escenarios negativos o escenarios condicionales sino para modelar parte o todos los comportamientos posibles de los protocolos del sistema.

Las herramientas TMCS[SC02], *Negative Scenarios*[UKM02] y LCS[HM02] son tres extensiones representativas de MSC que proveen cierto tipo de mecanismo de escenarios condicionales.

TMCS(*Triggered Message Sequence Charts*) extiende los MSCs con escenarios parciales y condicionales, conjuntamente con operadores de proceso de tipo algebraicos para integrar estas componentes. Sin embargo, esta herramienta no permite restricciones de tiempo real. La semántica formal esta basada en árboles de aceptación y un pre-ordenamiento "obligatorio" (para el refinamiento) (ver [SC02]). En este caso, los escenarios condicionales tienen el objetivo primario de refinar la especificación basada en un nodeterminísmo con el fin de restringir y conservar las secuencias deseadas (es decir, a diferencia de VTS, el objetivo no es describir la propiedad a verificar). En los escenarios condicionales cada secuencia de acciones de una instancia es subdividida en una parte inicial del *trigger* y una parte de la acción. Los eventos en la parte de la acción deben ocurrir si la parte del *trigger* es realizada por la instancia.

En [UKM02] se presenta un lenguaje para describir *Negative Scenarios* no temporizados basado en MSCs para extraer requerimientos. Aquí se utilizan precondiciones de tipo *trigger* ("after"/"until").

Los LCS(*Live Sequence Charts*)[HM02] incluyen notación de tipo *trigger* mediante *charts* y *precharts*. Estos buscan distinguir entre comportamiento obligatorio (universal) y comportamiento posible (existencial). La activación de los *charts* se utiliza para expresar cuando la parte principal del *chart* ocurrirá durante una ejecución del sistema.

[1] Requerimientos del tipo, por ejemplo, "la antigüedad de la medición generada por el sensor debe ser como máximo de 30 u.t. al momento que el controlador determina la acción."

Si bien se comparte con estas técnicas la idea de utilizar ordenes parciales para describir escenarios, VTS difiere en varios aspectos. Por un lado, VTS esta concebido para expresar propiedades para la verificación contra un modelo o una implementación bajo análisis; el objetivo no es construir un lenguaje del modelo ejecutable para las diferentes fases del proceso de desarrollo, o para la detección de requerimientos. Por el otro lado, no está limitado a describir intercambio de mensajes, o a definir la instancia que genera los eventos. Incluso, la visibilidad de eventos es tratada de forma diferente en el caso de VTS: dos eventos continuos de un escenario no requieren corresponder en la ejecución a eventos continuos, excepto cuando es explícitamente definido como una restricción en el escenario. La característica de *representativos-primeros/últimos* de VTS no está presente en los trabajos citados. Una diferencia notable es que, en los escenarios condicionales VTS, la condición de *trigger* (el antecedente) no esta limitada a que refiera a un comportamiento previo. Los consecuentes pueden corresponder a eventos que suceden antes de la condición de *trigger* o estar combinados con estos eventos. Adicionalmente, la notación de VTS y su herramienta declarativa opera con restricciones de tiempo real en un dominio de tiempo denso sin utilizar relojes, como es el caso de [HM02].

8.3 Trabajo futuro

Como parte del trabajo futuro se prevé incluir extensiones al lenguaje, como soportar modularidad, parametrización y extender las capacidades del matching de nombres de eventos como también mejorar el poder expresivo de los escenarios VTS.

También estamos explorando la definición de eventos compuestos (en el contexto del escenario global). Permitir recursión en estas definiciones extenderá el poder expresivo de la notación, dado que los eventos compuestos permitirán referir a características de un ilimitado historial de eventos (por ejemplo, "Si se recibe un número inusual de solicitudes, entonces la respuesta deber ser atendida en menos de 10 u.t."). En relación a esta extensión nosotros pensamos incorporar la definición de estado (a la "fluents", ver [GM03]) y sus utilización para expresar propiedades sobre intervalos.

Con el objetivo de evidenciar la relevancia de VTS, pensamos estudiar como algunos patrones (por ejemplo [DAC99]) se traducen en escenarios para orientar a los diseñadores en la construcción de requerimientos (algunos ejemplos pueden encontrarse en [Alf03]). También se proyecta comparar el poder expresivo de VTS respeto a lógicas de ejecuciones finitas lineales en el tiempo (por ejemplo [HR02]) para identificar fragmentos igualmente expresivos.

Por otro lado, estamos experimentando para la optimización del model-checking mediante la combinación las notaciones de escenarios con la tecnología de *slicing* como la herramienta *ObsSlice* [BGO02, BGO04]. La noción es que al especializar el escenario original a verificar, valiéndose de la notación de los escenarios condicionales, se incrementan los casos por los cuales la herramienta de *slicing* puede optimizar el modelo; logrando reducir el esfuerzo de verificación del modelchecker.

Respecto al front-end VTS, como los antiescenarios además de ser útiles para el proceso de verificación, de por si son un recurso valioso para los arquitectos (dado que éstos exhiben cómo podrían darse situaciones que no se quiere que ocurran en el sistema) una extensión que consideramos sumamente útil es la integración en la herramienta gráfica con los antiescenarios que construye el algoritmo.

Apéndice A

Demostraciones

A.1 Verificación de ECD

Teorema A.1.1 (Verificación de ECD)**.** Sea $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_i\}_{i=1...k} \rangle$ un ECD, σ una ejecución, entonces:

$\sigma \vDash \mathcal{C} \iff$ todo $\mathcal{N}_\mathcal{C} = \langle \mathcal{N}_{\mathcal{S}_1} \oplus \mathcal{N}_{\mathcal{S}_2} \oplus ... \oplus \mathcal{N}_{\mathcal{S}_k} \rangle$ un antiescenario final, donde $\mathcal{N}_{\mathcal{S}_j\, j=1...k}$ es un antiescenario generado por las reglas de $\mathcal{S}_j$, verifica $\sigma \nvDash \mathcal{N}_\mathcal{C}$.

Demostración.

$\Longrightarrow$) Supongamos que la afirmación no se cumple. Entonces existe un $\mathcal{N}_\mathcal{C}$ que verifica $\sigma \vDash \mathcal{N}_\mathcal{C}$. Luego, por *Lema A.2.1*, tenemos $\sigma \nvDash \mathcal{C}$, que es contradictorio con la hipótesis, por lo tanto la afirmación se verifica.

$\Longleftarrow$) Queremos probar que, si todo antiescenario final $\mathcal{N}_\mathcal{C}$ verifica $\sigma \nvDash \mathcal{N}_\mathcal{C} \implies \sigma \vDash \mathcal{C}$.
Esto es equivalente a, si $\sigma \nvDash \mathcal{C} \implies$ no todo antiescenario final $\mathcal{N}_\mathcal{C}$ verifica $\sigma \nvDash \mathcal{N}_\mathcal{C}$. Es decir, que existe un antiescenario final $\mathcal{N}_\mathcal{C}$ que verifica $\sigma \vDash \mathcal{N}_\mathcal{C}$.

Como $\mathcal{C}$ es ECD donde $\sigma \nvDash \mathcal{C}$, por la *Definición 3.1.4*, tenemos que necesariamente existe un matching $\hat{\cdot}$ entre $\mathcal{S}_0$ y σ, donde no existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{S}_i$ y σ, para ningún $i \in \{1..k\}$ tal que $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$. Luego, por *Lema A.9.1*, existe un antiescenario final $\mathcal{N}_\mathcal{C}$ que verifica $\sigma \vDash \mathcal{N}_\mathcal{C}$; que es lo que queríamos demostrar.

□

A.2 Correctitud de reglas para un ECD

Lema A.2.1 (Correctitud de reglas para un ECD)**.** Sea $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_i\}_{i=1...k} \rangle$ un ECD,
y sea $\mathcal{N}_\mathcal{C} = \langle \mathcal{N}_{\mathcal{S}_1} \oplus \mathcal{N}_{\mathcal{S}_2} \oplus ... \oplus \mathcal{N}_{\mathcal{S}_k} \rangle$ un antiescenario final, donde cada $\mathcal{N}_{\mathcal{S}_j\, j=1...k}$ es un antiescenario generado por las reglas de $\mathcal{S}_j$, entonces:

si existe una ejecución σ tal que $\sigma \vDash \mathcal{N}_\mathcal{C}$, entonces $\sigma \nvDash \mathcal{C}$.

Demostración.

Dado que $\sigma \vDash \mathcal{N}_\mathcal{C}$ entonces existe un matching $\hat{\cdot}$ entre σ y $\mathcal{N}_\mathcal{C}$.

Como $\mathcal{N}_\mathcal{C} = \langle \mathcal{N}_{\mathcal{S}_1} \oplus \mathcal{N}_{\mathcal{S}_2} \oplus ... \oplus \mathcal{N}_{\mathcal{S}_k} \rangle$, entonces, por la *Propiedad 4.1.2*, se puede afirmar que $\mathcal{N}_\mathcal{C} <: \mathcal{N}_{\mathcal{S}_j}$ para $j \in \{1...k\}$. Entonces, por la *Propiedad 3.1.2*, $\hat{\cdot}|_{\mathcal{N}_{\mathcal{S}_j}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_j}$ y σ.

Luego por la *Propiedad A.6.1* tenemos que $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{S}_0$, es decir, todo antiescenario especializa al antecedente.
Por lo tanto $\mathcal{N}_\mathcal{C} <: \mathcal{S}_0$, y por la *Propiedad 3.1.2* vale que:

(A) $\hat{\cdot}|_{\mathcal{S}_0}$ es un matching entre $\mathcal{S}_0$ y σ.

Aplicando *Lema A.3.1* para cada $\mathcal{N}_{\mathcal{S}_j}$ (con el matching $\hat{\cdot}|_{\mathcal{N}_{\mathcal{S}_j}}$ entre $\mathcal{N}_{\mathcal{S}_j}$ y σ) podemos afirmar que:

(B) No existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{S}_j$ y σ tal que $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$ **para ningún** $j \in \{1...k\}$.

Finalmente, podemos afirmar que $\sigma \not\models \mathcal{C}$ porque no se cumple la *Definición 3.1.4*, dado que encontramos un matching para el antecedente (**A**) que no se puede extender a ningún consecuente (**B**).

□

A.3 Correctitud de reglas para todo consecuente

Lema A.3.1 (Correctitud de reglas para todo consecuente)**.** Sea $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_i\}_{i=1...k} \rangle$ un ECD, $\mathcal{N}_{\mathcal{S}_j}$ un antiescenario generado por las reglas de $\mathcal{S}_{j\ j=1...k}$, sea $\sigma = \langle s, \tau \rangle$ una ejecución, $\hat{\cdot}$ un matching entre $\mathcal{N}_{\mathcal{S}_j}$ y σ, entonces se verifica:

(A) $\hat{\cdot}|_{\mathcal{S}_0}$ es un matching entre $\mathcal{S}_0$ y σ.

(B) No existe $\hat{\hat{\cdot}}$ matching entre $\mathcal{S}_j$ y σ tal que $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$

y en consecuencia $\sigma \not\models \langle \mathcal{S}_0, \{\mathcal{S}_j\} \rangle$.

Demostración.

Para (**A**), aplicando la *Propiedad A.6.1* tenemos que $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{S}_0$, es decir, todo antiescenario especializa al antecedente. Por lo tanto podemos afirmar que $\hat{\cdot}|_{\mathcal{S}_0}$ es un matching entre $\mathcal{S}_0$ y σ.

Para (**B**), vamos a suponer que existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{S}_j$ y σ donde $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$. Ahora analizaremos el antiescenario $\mathcal{N}_{\mathcal{S}_j}$ que corresponde de aplicar al consecuente $\mathcal{S}_j$ una de las siguientes reglas:

- **Puntos sin matching** (Definición 4.2.3): para un punto concreto $\mathsf{p} \in P_j \setminus P_0$.
  - $\mathcal{N}^{\mathcal{S}_0 \not\rightarrow \mathsf{p}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \not\rightarrow \mathsf{p}}$, donde $\{\mathsf{q}\} = ld_{\lessdot}(\mathsf{p})$

    Dado que $\{\mathsf{q}\} = ld_{\lessdot}(\mathsf{p})$, y como p es concreto, de acuerdo a la definición de $ld_{\lessdot}$, entonces $\mathsf{q} \hookrightarrow \mathsf{p}$. Por lo tanto, se verifica:

    1 $\ell_j(\mathsf{p}) \subseteq \gamma_j(\mathsf{q}, \mathsf{p})$ cuando $\mathsf{q} <_j \mathsf{p}$
    2 $\ell_j(\mathsf{p}) \subseteq \gamma_j(\mathsf{p}, \mathsf{q})$ cuando $\mathsf{p} <_j \mathsf{q}$

    Si es (**1**), tenemos $\mathsf{q} <_j \mathsf{p}$, y por **M3** entonces $\hat{\mathsf{q}} < \hat{\mathsf{p}}$.

    Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathsf{q} \not\rightarrow \mathsf{p}}$ tenemos $\hat{\cdot}|_{\mathcal{P}^{\mathsf{q} \not\rightarrow \mathsf{p}}}$ es matching de $\mathcal{P}^{\mathsf{q} \not\rightarrow \mathsf{p}}$, La definición de este escenario tiene $\ell_j(\mathsf{p}) \subseteq \gamma(\mathsf{q}, \infty)$, y por **M5**, sabemos que $s_{(\hat{\mathsf{q}}} \cap \ell_j(\mathsf{p}) = \emptyset$. Es decir, que no puede haber eventos de $\ell_j(\mathsf{p})$ posteriores al matching $\hat{\mathsf{q}}$, lo que implica $\hat{\hat{\mathsf{p}}} < \hat{\mathsf{q}}$. Por lo tanto, necesariamente $\hat{\mathsf{q}} < \hat{\hat{\mathsf{p}}} < \hat{\mathsf{q}}$, y entonces $\hat{\mathsf{q}} \neq \hat{\mathsf{q}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{q}} = \hat{\mathsf{q}}$.

    Si es (**2**), tenemos $\mathsf{p} <_j \mathsf{q}$, y por **M3** entonces $\hat{\mathsf{p}} < \hat{\mathsf{q}}$.

    Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathsf{q} \not\rightarrow \mathsf{p}}$ tenemos $\hat{\cdot}|_{\mathcal{P}^{\mathsf{q} \not\rightarrow \mathsf{p}}}$ es matching de $\mathcal{P}^{\mathsf{q} \not\rightarrow \mathsf{p}}$. La definición de este escenario tiene $\ell_j(\mathsf{p}) \subseteq \gamma(\mathbf{0}, \mathsf{q})$, y por **M5**, sabemos que $s_{\hat{\mathsf{q}})} \cap \ell_j(\mathsf{p}) = \emptyset$. Es decir, que no puede haber eventos de $\ell_j(\mathsf{p})$ anteriores al matching $\hat{\mathsf{q}}$, lo que implica $\hat{\mathsf{q}} < \hat{\hat{\mathsf{p}}}$. Por lo tanto, necesariamente $\hat{\mathsf{q}} < \hat{\hat{\mathsf{p}}} < \hat{\hat{\mathsf{q}}}$, y entonces $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$.

- **Violación de restricciones** (Definición 4.2.4): para un punto $\mathsf{p} \in P_j$ y un $\mathsf{q} \in P_j$

- **Caso:** $(\mathsf{p}, \infty) \in <_j$

  – $\mathcal{N}^{\neg\gamma_j(\mathsf{p},\infty)} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\gamma_j(\mathsf{p},\infty)}$ cuando $(\mathsf{p}, \infty) \in \gamma_j \smallsetminus \gamma_p$

  Dado que $(\mathsf{p}, \infty) \in \gamma_j \smallsetminus \gamma_p$ tenemos $\gamma_j(\mathsf{p}, \infty) \neq \emptyset$. Luego, por **M5**, se debe cumplir $s_{\langle\hat{\mathsf{p}}} \cap \gamma_j(\mathsf{p}, \infty) = \emptyset$. Es decir, que podemos afirmar que luego del matching $\hat{\mathsf{p}}$ no puede haber ningún matching a eventos que figuran en $\gamma_j(\mathsf{p}, \infty)$.

  Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\neg\gamma_j(\mathsf{p},\infty)}$ tenemos que $\hat{\cdot}|_{\mathcal{P}^{\neg\gamma_j(\mathsf{p},\infty)}}$ es matching de $\mathcal{P}^{\neg\gamma_j(\mathsf{p},\infty)}$. La definición de este escenario incorpora un nuevo punto n etiquetado con al menos un evento de $\gamma_j(\mathsf{p}, \infty)$. Por **M1** tenemos que $s_{\hat{\mathsf{n}}} \in \ell(\mathsf{n})$ y, como $\mathsf{p} < \mathsf{n}$, por **M3**, sabemos que $\hat{\mathsf{p}} < \hat{\mathsf{n}}$. Entonces, después $\hat{\mathsf{p}}$ necesariamente hay un matching con un evento de $\gamma_j(\mathsf{p}, \infty)$.

  Por lo tanto, por la afirmación anterior, necesariamente $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$.

- **Caso:** $(\mathbf{0}, \mathsf{p}) \in <_j$

  – $\mathcal{N}^{\neg\gamma_j(\mathbf{0},\mathsf{p})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\gamma_j(\mathbf{0},\mathsf{p}}$ cuando $(\mathbf{0}, \mathsf{p}) \in \gamma_j \smallsetminus \gamma_p$

  Dado que $(\mathbf{0}, \mathsf{p}) \in \gamma_j \smallsetminus \gamma_p$ tenemos $\gamma_j(\mathbf{0}, \mathsf{p}) \neq \emptyset$. Luego, por **M5**, se debe cumplir $s_{\langle\hat{\mathsf{p}})} \cap \gamma_j(\mathbf{0}, \mathsf{p}) = \emptyset$. Es decir, que podemos afirmar que hasta el matching $\hat{\mathsf{p}}$ no puede haber ningún matching a eventos que figuran en $\gamma_j(\mathbf{0}, \mathsf{p})$.

  Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\neg\gamma_j(\mathbf{0},\mathsf{p})}$ tenemos $\hat{\cdot}|_{\mathcal{P}^{\neg\gamma_j(\mathbf{0},\mathsf{p})}}$ es matching de $\mathcal{P}^{\neg\gamma_j(\mathbf{0},\mathsf{p})}$. La definición de este escenario incorpora un nuevo punto n etiquetado con al menos un evento de $\gamma_j(\mathbf{0}, \mathsf{p})$. Por **M1** tenemos que $s_{\hat{\mathsf{n}}} \in \ell(\mathsf{n})$ y, como $\mathsf{n} < \mathsf{p}$, por **M3**, sabemos que $\hat{\mathsf{n}} < \hat{\mathsf{p}}$. Entonces, antes de $\hat{\mathsf{p}}$ necesariamente hay un matching con un evento de $\gamma_j(\mathbf{0}, \mathsf{p})$.

  Por lo tanto, por la afirmación anterior, necesariamente $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$.

  – $\mathcal{N}^{\neg\delta_j(\mathbf{0},\mathsf{p})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\delta_j(\mathbf{0},\mathsf{p})}$ cuando $(\mathbf{0}, \mathsf{p}) \in \delta_j \subsetneq \delta_p$

  Dado que $(\mathbf{0}, \mathsf{p}) \in \delta_j \subsetneq \delta_p$ tenemos que $\delta_j(\mathbf{0}, \mathsf{p}) \neq \emptyset$. Luego, por **M7**, podemos afirmar que $\Delta(\tau_{\hat{\mathsf{p}}]}) \vDash \delta_j(\mathbf{0}, \mathsf{p})$. Es decir, el tiempo transcurrido hasta $\hat{\mathsf{p}}$ satisface la restricción temporal $\delta_j(\mathbf{0}, \mathsf{p})$.

  Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\neg\delta_j(\mathbf{0},\mathsf{p})}$ tenemos $\hat{\cdot}|_{\mathcal{P}^{\neg\delta_j(\mathbf{0},\mathsf{p})}}$ es matching de $\mathcal{P}^{\neg\delta_j(\mathbf{0},\mathsf{p})}$. La definición de este escenario tiene que $\delta(\mathbf{0}, \mathsf{p}) = \neg\delta_j(\mathbf{0}, \mathsf{p})$. Luego, por **M7** se debe cumplir que $\Delta(\tau_{\hat{\mathsf{p}}]}) \vDash \neg\delta_j(\mathbf{0}, \mathsf{p})$. Es decir, el tiempo transcurrido hasta $\hat{\mathsf{p}}$ satisface la restricción temporal $\neg\delta_j(\mathbf{0}, \mathsf{p})$.

  Por lo tanto, por la afirmación anterior, necesariamente $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$.

- **Caso:** $(\mathsf{p}, \mathsf{q}) \in <_j$

  – $\mathcal{N}^{\mathsf{p} \not<_j \mathsf{q}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not<_j \mathsf{q}}$, cuando $(\mathsf{p}, \mathsf{q}) \notin (<_p \cup <_q)$

  Dado que $\mathsf{p} <_j \mathsf{q}$, por **M3**, podemos afirmar que $\hat{\mathsf{p}} < \hat{\mathsf{q}}$.

  Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathsf{p} \not<_j \mathsf{q}}$ tenemos $\hat{\cdot}|_{\mathcal{P}^{\mathsf{p} \not<_j \mathsf{q}}}$ es matching de $\mathcal{P}^{\mathsf{p} \not<_j \mathsf{q}}$. La definición de este escenario tiene que $\mathsf{q} <_j \mathsf{p}$, por **M3**, se debe cumplir $\hat{\mathsf{q}} < \hat{\mathsf{p}}$.

Por lo tanto, por la afirmación anterior, necesariamente $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$ o bien $\hat{\mathsf{q}} \neq \hat{\hat{\mathsf{q}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$ y $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$.

– $\mathcal{N}^{\mathsf{p} \not\asymp_j \mathsf{q}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not\asymp_j \mathsf{q}}$ cuando $(\mathsf{p}, \mathsf{q}) \notin (\neq_p \cup \neq_q)$

Dado que $\mathsf{p} <_j \mathsf{q}$, por **M3**, tenemos que $\hat{\mathsf{p}} < \hat{\mathsf{q}}$ y podemos afirmar que $\hat{\hat{\mathsf{p}}} \neq \hat{\hat{\mathsf{q}}}$.

Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathsf{p} \not\asymp_j \mathsf{q}}$ tenemos $\hat{\cdot}|_{\mathcal{P}^{\mathsf{p} \not\asymp_j \mathsf{q}}}$ es matching de $\mathcal{P}^{\mathsf{p} \not\asymp_j \mathsf{q}}$. La definición de este escenario tiene un nuevo punto u, que es representativo-primero de los puntos $\{\mathsf{p},\mathsf{q}\}$ y también es representativo-último de estos mismos puntos. Entonces, por **M8**, $\hat{\mathsf{u}} = \min\{\hat{\mathsf{p}}, \hat{\mathsf{q}}\}$ y también $\hat{\mathsf{u}} = \max\{\hat{\mathsf{p}}, \hat{\mathsf{q}}\}$; condición que se cumple sólo si $\hat{\mathsf{p}} = \hat{\mathsf{q}}$.

Por lo tanto, por la afirmación anterior, necesariamente $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$ o bien $\hat{\mathsf{q}} \neq \hat{\hat{\mathsf{q}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$ y $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$.

– $\mathcal{N}^{\neg\gamma_j(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\gamma_j(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p}, \mathsf{q}) \in \gamma_j \setminus (\gamma_p \cup \gamma_q)$

Dado que $(\mathsf{p}, \mathsf{q}) \in \gamma_j \setminus (\gamma_p \cup \gamma_q)$ tenemos $\gamma_j(\mathsf{p}, \mathsf{q}) \neq \emptyset$. Luego, por **M4**, podemos afirmar que $s_{(\hat{\hat{\mathsf{p}}},\ \hat{\hat{\mathsf{q}}})} \cap \gamma(\mathsf{p}, \mathsf{q}) = \emptyset$. Es decir, que entre los matching de $\hat{\hat{\mathsf{p}}}$ y $\hat{\hat{\mathsf{q}}}$ no puede haber ningún evento que figura en $\gamma_j(\mathsf{p}, \mathsf{q})$.

Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\neg\gamma_j(\mathsf{p},\mathsf{q})}$ tenemos que $\hat{\cdot}|_{\mathcal{P}^{\neg\gamma_j(\mathsf{p},\mathsf{q})}}$ es matching de $\mathcal{P}^{\neg\gamma_j(\mathsf{p},\mathsf{q})}$. La definición de este escenario incorpora un nuevo punto n etiquetado con al menos un evento de $\gamma_j(\mathsf{p}, \mathsf{q})$. Por **M1** tenemos que $s_{\hat{\mathsf{n}}} \in \ell(\mathsf{n})$ y, como $\mathsf{p} < \mathsf{n}$ y $\mathsf{n} < \mathsf{q}$, por **M3**, sabemos que $\hat{\mathsf{p}} < \hat{\mathsf{n}} < \hat{\mathsf{q}}$. Entonces, entre $\hat{\mathsf{p}}$ y $\hat{\mathsf{q}}$ necesariamente hay un matching con un evento de $\gamma_j(\mathsf{p}, \mathsf{q})$.

Por lo tanto, por la afirmación anterior, necesariamente $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$ o bien $\hat{\mathsf{q}} \neq \hat{\hat{\mathsf{q}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$ y $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$.

– $\mathcal{N}^{\neg\delta_j(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\delta_j(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p}, \mathsf{q}) \in \delta_j$

Dado que $(\mathsf{p}, \mathsf{q}) \in \delta_j \subsetneq \delta_p$ tenemos $\delta_j(\mathsf{p}, \mathsf{q}) \neq \emptyset$. Luego, por **M6**, podemos afirmar que $\Delta(\tau_{[\hat{\hat{\mathsf{p}}},\hat{\hat{\mathsf{q}}}]}) \vDash \delta_j(\mathsf{p}, \mathsf{q})$. Es decir, que el tiempo transcurrido entre los matchings $\hat{\hat{\mathsf{p}}}$ y $\hat{\hat{\mathsf{q}}}$ satisface la restricción temporal $\delta_j(\mathsf{p}, \mathsf{q})$.

Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\neg\delta_j(\mathsf{p},\mathsf{q})}$ tenemos que $\hat{\cdot}|_{\mathcal{P}^{\neg\delta_j(\mathsf{p},\mathsf{q})}}$ es matching de $\mathcal{P}^{\neg\delta_j(\mathsf{p},\mathsf{q})}$. La definición de este escenario tiene que $\delta(\mathsf{p}, \mathsf{q}) = \neg\delta_j(\mathsf{p}, \mathsf{q})$. Luego, por **M6**, se debe cumplir que $\Delta(\tau_{[\hat{\mathsf{p}},\hat{\mathsf{q}}]}) \vDash \neg\delta_j(\mathsf{p}, \mathsf{q})$. Es decir, que el tiempo transcurrido entre $\hat{\mathsf{p}}$ y $\hat{\mathsf{q}}$ satisface la restricción temporal $\neg\delta_j(\mathsf{p}, \mathsf{q})$.

Por lo tanto, por la afirmación anterior, necesariamente $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$ o bien $\hat{\mathsf{q}} \neq \hat{\hat{\mathsf{q}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$ y $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$.

- **Caso:** $(\mathsf{p}, \mathsf{q}) \in (\not\equiv_j)$

– $\mathcal{N}^{\mathsf{p} \not\asymp_j \mathsf{q}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not\asymp_j \mathsf{q}}$ cuando $(\mathsf{p}, \mathsf{q}) \notin (\neq_p \cup \neq_q)$

Dado que $\mathsf{p} \not\equiv_j \mathsf{q}$, por **M2**, podemos afirmar que $\hat{\hat{\mathsf{p}}} \neq \hat{\hat{\mathsf{q}}}$.

Ahora podemos aplicar la misma argumentación de la regla $\mathcal{N}^{\mathsf{p} \not\asymp_j \mathsf{q}}$ del **Caso:** $(\mathsf{p}, \mathsf{q}) \in <_j$ se llega también a un absurdo.

– $\mathcal{N}^{\neg\gamma_j(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\gamma_j(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p},\mathsf{q}) \in \gamma_j \setminus (\gamma_p \cup \gamma_q)$

Dado que $(\mathsf{p},\mathsf{q}) \in \gamma_j \setminus (\gamma_p \cup \gamma_q)$, tenemos que $\gamma_j(\mathsf{p},\mathsf{q}) \neq \emptyset$. Luego, por **M4**, podemos afirmar que $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{q}})} \cap \gamma_j(\mathsf{p},\mathsf{q}) = \emptyset$.

Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\neg\gamma_j(\mathsf{p},\mathsf{q})}$ tenemos que $\hat{:}|_{\mathcal{P}^{\neg\gamma_j(\mathsf{p},\mathsf{q})}}$ es matching de $\mathcal{P}^{\neg\gamma_j(\mathsf{p},\mathsf{q})}$. La definición de este escenario tiene un nuevo punto i etiquetado con los eventos $\gamma_j(\mathsf{p},\mathsf{q})$, un nuevo punto f que es representativo-primero de los puntos {p,q}, y un nuevo punto l que es representativo-último de los puntos {p,q}, donde además $\mathsf{f} < \mathsf{i} < \mathsf{l}$. Entonces, por **M8**, $\hat{\mathsf{f}} = \min\{\hat{\mathsf{p}},\hat{\mathsf{q}}\}$ y también $\hat{\mathsf{l}} = \max\{\hat{\mathsf{p}},\hat{\mathsf{q}}\}$; y por **M3**, se debe cumplir además que $\hat{\mathsf{f}} < \hat{\mathsf{i}} < \hat{\mathsf{l}}$ donde, por **M1**, $s_{\hat{\mathsf{i}}} \in \gamma_j(\mathsf{p},\mathsf{q})$.

Por lo tanto, por la afirmación anterior, necesariamente $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$ o bien $\hat{\mathsf{q}} \neq \hat{\hat{\mathsf{q}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$ y $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$.

– $\mathcal{N}^{\neg\delta_j(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\delta_j(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p},\mathsf{q}) \in \delta_j$

Aplicando la misma argumentación de la regla $\mathcal{N}^{\neg\delta_j(\mathsf{p},\mathsf{q})}$ del Caso $(\mathsf{p},\mathsf{q}) \in <_j$ se llega también a un absurdo.

- **Violación de Aliasing** (Definición 4.2.5): para un punto $\mathsf{p} \in P_j$

– $\mathcal{N}^{\mathsf{p}\leftrightarrow\mathsf{p}} = \bigoplus_{\mathsf{q}\in FirstOf_j(\mathsf{p})}(\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{n}<_F\mathsf{q}}) \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathsf{p}\neq\mathsf{n}}$ cuando $\exists \mathsf{q}(\mathsf{p},\mathsf{q}) \in <_{Fj} \setminus <_{Fp}$.

Veamos las siguientes propiedades:

* Como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y, por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$, podemos aplicar el *Lema A.4.1* por el cual tenemos $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$.

* Llamemos $\mathcal{R}$ al escenario $\bigoplus_{\mathsf{q}\in FirstOf_j(\mathsf{p})}(\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q}<_F\mathsf{n}})$. Dado que $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{R}$ tenemos que $\hat{:}|_{\mathcal{R}}$ es matching de $\mathcal{R}$. En este escenario se define un nuevo punto n que es representativo-primero $\forall \mathsf{q} \in FirstOf_j(\mathsf{p})$. Como $\exists \mathsf{q}/(\mathsf{q},\mathsf{p}) \in <_{Fj}$ entonces $\mathsf{n} \in FirstRep_{\mathcal{R}}$. Luego, por **M8**, tenemos $\hat{\mathsf{n}} = \min\{\hat{\mathsf{q}}/\mathsf{q} \in FirstOf_j(\mathsf{p})\}$.

En $\mathcal{S}_j$, como $\exists \mathsf{q}/(\mathsf{q},\mathsf{p}) \in <_{Fj}$, sabemos que $\mathsf{p} \in FirstRep_j$. Entonces, por **M8**, $\hat{\hat{\mathsf{p}}} = \min\{\hat{\hat{\mathsf{q}}}/\mathsf{q} \in FirstOf_j(\mathsf{p})\}$. Luego, para $\forall \mathsf{q} \in FirstOf_j(\mathsf{p})$ tenemos $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, entonces, por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} <:: \mathcal{S}_0$, por lo tanto podemos aplicar el *Lema A.4.1* por el cual $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$ y podemos afirmar:

$$\hat{\hat{\mathsf{p}}} = \min\{\hat{\hat{\mathsf{q}}}/\mathsf{q} \in FirstOf_j(\mathsf{p})\} = \min\{\hat{\mathsf{q}}/\mathsf{q} \in FirstOf_j(\mathsf{p})\} = \hat{\mathsf{n}}$$

* Como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathsf{p}\neq\mathsf{n}}$ tenemos que $\hat{:}|_{\mathcal{P}^{\mathsf{p}\neq\mathsf{n}}}$ es matching de $\mathcal{P}^{\mathsf{p}\neq\mathsf{n}}$. En este escenario se define $\mathsf{n} \neq \mathsf{p}$, entonces, por **M2**, sabemos que $\hat{\mathsf{n}} \neq \hat{\mathsf{p}}$.

Pero entonces como $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$, $\hat{\hat{\mathsf{p}}} = \hat{\mathsf{n}}$, y $\hat{\mathsf{n}} \neq \hat{\mathsf{p}}$ tenemos:

$$\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}} = \hat{\mathsf{n}} \neq \hat{\mathsf{p}} \text{ , es decir, } \hat{\mathsf{p}} \neq \hat{\mathsf{p}}, \text{ que es absurdo.}$$

– $\mathcal{N}^{\mathsf{p}\leftrightarrow\mathsf{p}} = \bigoplus_{\mathsf{q}\in LastOf_j(\mathsf{p})}(\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q}<_L\mathsf{n}}) \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathsf{p}\neq\mathsf{n}}$ cuando $\exists \mathsf{q}(\mathsf{q},\mathsf{p}) \in <_{Lj} \setminus <_{Lp}$.

Veamos las siguientes propiedades:

* Como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y, por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$, podemos aplicar el *Lema A.4.1* por el cual tenemos $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$.

* Llamemos $\mathcal{R}$ al escenario $\bigoplus_{\mathsf{q}\in LastOf_j(\mathsf{p})}(\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q}<_L\mathsf{n}})$. Dado que $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{R}$ tenemos que $\hat{:}|_{\mathcal{R}}$ es matching de $\mathcal{R}$. En este escenario se define un nuevo punto n que es representativo-primero $\forall \mathsf{q} \in LastOf_j(\mathsf{p})$. Como $\exists \mathsf{q}/(\mathsf{q},\mathsf{p}) \in <_{Lj}$ entonces $\mathsf{n} \in LastRep_{\mathcal{R}}$. Luego, por **M8**, tenemos $\hat{\mathsf{n}} = \max\{\hat{\mathsf{q}}/\mathsf{q} \in LastOf_j(\mathsf{p})\}$.

En $\mathcal{S}_j$, como $\exists \mathsf{q}/(\mathsf{q},\mathsf{p}) \in <_{Lj}$, sabemos que $\mathsf{p} \in \mathit{LastRep}_j$. Entonces, por **M8**, $\hat{\mathsf{p}} = \max\{\hat{\mathsf{q}}/\mathsf{q} \in \mathit{LastOf}_j(\mathsf{p})\}$. Luego, para $\forall \mathsf{q} \in \mathit{LastOf}_j(\mathsf{p})$ tenemos $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, entonces, por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} <:: \mathcal{S}_0$, por lo tanto podemos aplicar el *Lema A.4.1* por el cual $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$ y podemos afirmar:

$$\hat{\hat{\mathsf{p}}} = \max\{\hat{\hat{\mathsf{q}}}/\mathsf{q} \in \mathit{LastOf}_j(\mathsf{p})\} = \max\{\hat{\mathsf{q}}/\mathsf{q} \in \mathit{LastOf}_j(\mathsf{p})\} = \hat{\mathsf{n}}$$

∗ Como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathsf{p} \not\equiv \mathsf{n}}$ tenemos que $\hat{\cdot}|_{\mathcal{P}^{\mathsf{p} \not\equiv \mathsf{n}}}$ es matching de $\mathcal{P}^{\mathsf{p} \not\equiv \mathsf{n}}$. En este escenario se define $\mathsf{p} \neq \mathsf{n}$, entonces, por **M2**, sabemos que $\hat{\mathsf{p}} \neq \hat{\mathsf{n}}$.

Pero entonces como $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$, $\hat{\hat{\mathsf{p}}} = \hat{\mathsf{n}}$, y $\hat{\mathsf{n}} \neq \hat{\mathsf{p}}$ tenemos:

$$\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}} = \hat{\mathsf{n}} \neq \hat{\mathsf{p}} \text{ , es decir, } \hat{\mathsf{p}} \neq \hat{\mathsf{p}}, \text{ que es absurdo.}$$

- **Violación de eventos limitados** (Definición 4.2.6): sea un punto $\mathsf{p} \in P_0$
  - $\mathcal{N}^{\mathcal{S}_0 \not\rightsquigarrow \mathsf{p}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\not\rightsquigarrow \mathsf{p}}$, cuando $\ell_j(\mathsf{p}) \subsetneq \ell_0(\mathsf{p})$

    Por **M1** tenemos que $s_{\hat{\hat{\mathsf{p}}}} \in \ell_j(\mathsf{p})$. Es decir que podemos afirmar que el matching $\hat{\hat{\mathsf{p}}}$ corresponde a un evento del conjunto $\ell_j(\mathsf{p})$.

    Ahora, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\not\rightsquigarrow \mathsf{p}}$ tenemos que $\hat{\cdot}|_{\mathcal{P}^{\not\rightsquigarrow \mathsf{p}}}$ es matching de $\mathcal{P}^{\not\rightsquigarrow \mathsf{p}}$. La definición de este escenario tiene $\ell(\mathsf{p}) = \ell_0(\mathsf{p}) \setminus \ell_j(\mathsf{p})$. Luego, por **M1**, sabemos que $s_{\hat{\mathsf{p}}} \in (\ell_0(\mathsf{p}) \setminus \ell_j(\mathsf{p}))$. Es decir, el matching $\hat{\mathsf{p}}$ corresponde a un evento que no esta en $\ell_j(\mathsf{p})$.

    Por lo tanto, por la afirmación anterior, necesariamente $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$. Esto es absurdo porque, como $\mathcal{N}_{\mathcal{S}_j} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, y por la *Propiedad A.7.1*, tenemos que $\mathcal{S}_j <:: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0$, entonces podemos aplicar el *Lema A.4.1* por el cual tenemos que $\hat{\mathsf{p}} = \hat{\hat{\mathsf{p}}}$.

Finalmente, vimos que para cada una de las reglas se llega a un absurdo, y por lo tanto la suposición no es válida. Entonces podemos afirmar que no existe $\hat{\cdot}$ matching entre $\mathcal{S}_j$ y σ tal que $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

□

A.4 Extensión determinística para todo subescenario en común

Lema A.4.1 (Extensión determinística para todo subescenario en común)**.** Sean $\mathcal{S}_0$, $\mathcal{P}$, $\mathcal{S}_1$ y $\mathcal{N}_{\mathcal{S}_1}$ cuatro escenarios donde $\mathcal{N}_{\mathcal{S}_1} <: \mathcal{P}$, $\mathcal{S}_1 <: \mathcal{P}$ y $\mathcal{P} <:: \mathcal{S}_0$, sea una ejecución σ, dado un matching $\hat{\cdot}$ entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, y un matching $\hat{\hat{\cdot}}$ entre $\mathcal{S}_1$ y σ tal que $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\hat{\cdot}}|_{\mathcal{S}_0}$, entonces

$$\hat{\cdot}|_{\mathcal{P}} \text{ y } \hat{\hat{\cdot}}|_{\mathcal{P}} \text{ son matchings entre } \mathcal{P} \text{ y } \sigma, \text{ y } \hat{\cdot}|_{\mathcal{P}} = \hat{\hat{\cdot}}|_{\mathcal{P}}$$

Demostración.

Primero veamos que $\hat{\cdot}|_{\mathcal{P}}$ y $\hat{\hat{\cdot}}|_{\mathcal{P}}$ son matchings entre $\mathcal{P}$ y σ.

Dado que $\mathcal{N}_{\mathcal{S}_1} <: \mathcal{P}$ por la *Propiedad 3.1.2* podemos afirmar que $\hat{\cdot}|_{\mathcal{P}}$ es matching entre $\mathcal{P}$ y σ.
Análogamente, dado que $\mathcal{S}_1 <:: \mathcal{P}$ podemos afirmar que $\hat{\hat{\cdot}}|_{\mathcal{P}}$ también es matching entre $\mathcal{P}$ y σ.

Ahora veamos que necesariamente se cumple que $\hat{\cdot}|_{\mathcal{P}} = \hat{\hat{\cdot}}|_{\mathcal{P}}$.

Dado que $\mathcal{P} <:: \mathcal{S}_0$ por la *Definición 3.1.1* (**Sp1**) sabemos que $P_0 \subseteq P_{\mathcal{P}}$ entonces $(\hat{\cdot}|_{\mathcal{P}})|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$ y $(\hat{\hat{\cdot}}|_{\mathcal{P}})|_{\mathcal{S}_0} = \hat{\hat{\cdot}}|_{\mathcal{S}_0}$. Por lo tanto, dado que $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\hat{\cdot}}|_{\mathcal{S}_0}$ entonces $(\hat{\cdot}|_{\mathcal{P}})|_{\mathcal{S}_0} = (\hat{\hat{\cdot}}|_{\mathcal{P}})|_{\mathcal{S}_0}$.

Finalmente, como $\mathcal{P} <:: \mathcal{S}_0$, y $\hat{\cdot}|_{\mathcal{P}}$ y $\hat{\hat{\cdot}}|_{\mathcal{P}}$ son matchings entre $\mathcal{P}$ y σ tal que $(\hat{\cdot}|_{\mathcal{P}})|_{\mathcal{S}_0} = (\hat{\hat{\cdot}}|_{\mathcal{P}})|_{\mathcal{S}_0}$ podemos aplicar el *Lema A.5.1* concluyendo entonces que $\hat{\cdot}|_{\mathcal{P}} = \hat{\hat{\cdot}}|_{\mathcal{P}}$

□

A.5 Extensión determinística

Lema A.5.1 (Extensión determinística)**.** Sean $\mathcal{S}_0$ y $\mathcal{S}_1$ dos escenarios tal que $\mathcal{S}_1 <:: \mathcal{S}_0$ por un ranking $\lessdot$ sobre P_1, sea una ejecución $\sigma = \langle s, \tau \rangle$, sean $\hat{\cdot}$ y $\hat{\hat{\cdot}}$ dos matchings entre $\mathcal{S}_1$ y σ tal que $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\hat{\cdot}}|_{\mathcal{S}_0}$, entonces $\hat{\cdot} = \hat{\hat{\cdot}}$, es decir, dado un matching de $\mathcal{S}_0$ a lo sumo hay una única forma de extenderlo como matching de $\mathcal{S}_1$.

Demostración.

Supongamos que $\hat{\cdot} \neq \hat{\hat{\cdot}}$, entonces existe un punto $\mathsf{p} \in P_1 \setminus P_0$ tal que $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$. Elijamos p como el primero de los puntos respecto al ranking tal que difieren en el matching. Es decir, $\mathsf{p} = \text{mín}_{\lessdot}\{\mathsf{q} \in P_1 \setminus P_0 \mid \hat{\mathsf{q}} \neq \hat{\hat{\mathsf{q}}}\}$.

Por la *Definición 3.2.4* tenemos que $\lessdot\mathsf{p} \hookrightarrow \mathsf{p}$ en $\mathcal{S}_1$ (donde $\lessdot\mathsf{p} = \{\mathsf{q} \in P_1 \mid \mathsf{q} \lessdot \mathsf{p}\}$). Luego por la *Definición 3.2.2* se debe cumplir:

1 o bien $\mathsf{p} \in \lessdot\mathsf{p}$,

2 o existe un punto $\mathsf{p}' \in \lessdot\mathsf{p}$ tal que $\mathsf{p}' \hookrightarrow \mathsf{p}$ (cuando p es un punto concreto),

3 o $FirstOf(\mathsf{p}) \subseteq \lessdot\mathsf{p}$ (cuando $\mathsf{p} \in FirstRep$),

4 o $LastOf(\mathsf{p}) \subseteq \lessdot\mathsf{p}$ (cuando $\mathsf{p} \in LastRep$).

Vamos a probar que ninguno de estos casos se verifica. El caso (**1**) no se cumple porque $\lessdot$ es un orden total. En (**2**), como $\mathsf{p}' \hookrightarrow \mathsf{p}$, se debe verificar:

2.1 $\ell(\mathsf{p}) \subseteq \gamma(\mathsf{p}', \mathsf{p})$ cuando $\mathsf{p}' < \mathsf{p}$

2.2 $\ell(\mathsf{p}) \subseteq \gamma(\mathsf{p}, \mathsf{p}')$ cuando $\mathsf{p} < \mathsf{p}'$

Por un lado, por como fue elegido p y dado que $\mathsf{p}' \lessdot \mathsf{p}$ tenemos necesariamente que $\hat{\mathsf{p}}' = \hat{\hat{\mathsf{p}}}'$. Por el otro, como $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$ entonces o bien $\hat{\mathsf{p}} < \hat{\hat{\mathsf{p}}}$ (**A**) o bien $\hat{\mathsf{p}} > \hat{\hat{\mathsf{p}}}$ (**B**).

Si es (**A**), entonces $\hat{\mathsf{p}} < \hat{\hat{\mathsf{p}}}$.

- En (**2.1**), tenemos $\mathsf{p}' < \mathsf{p}$ y por **M3** sabemos que $\hat{\mathsf{p}}' < \hat{\mathsf{p}}$ entonces $\hat{\mathsf{p}}' < \hat{\mathsf{p}} < \hat{\hat{\mathsf{p}}}$.
  Luego, como $\hat{\mathsf{p}}' = \hat{\hat{\mathsf{p}}}'$ tenemos $\hat{\hat{\mathsf{p}}}' < \hat{\mathsf{p}} < \hat{\hat{\mathsf{p}}}$. Es decir, entre las posiciones de los matchings $\hat{\hat{\mathsf{p}}}'$ y $\hat{\mathsf{p}}$ esta la posición de $\hat{\mathsf{p}}$, que por **M1** es $s_{\hat{\mathsf{p}}} \in \ell(\mathsf{p})$.
  Sin embargo, por **M4** tenemos $s_{(\hat{\mathsf{p}}',\ \hat{\mathsf{p}})} \cap \gamma(\mathsf{p}', \mathsf{p}) = \emptyset$ y como por (**2.1**) ocurre $\ell(\mathsf{p}) \subseteq \gamma(\mathsf{p}', \mathsf{p})$ entonces $s_{(\hat{\mathsf{p}}',\ \hat{\mathsf{p}})} \cap \ell(\mathsf{p}) = \emptyset$. Es decir, entre las posiciones de los matchings $\hat{\mathsf{p}}'$ y $\hat{\mathsf{p}}$ no hay ningún evento de p lo cual es absurdo con la afirmación anterior.
- En (**2.2**), tenemos $\mathsf{p} < \mathsf{p}'$ y por **M3** sabemos que $\hat{\mathsf{p}} < \hat{\hat{\mathsf{p}}}'$ entonces $\hat{\mathsf{p}} < \hat{\hat{\mathsf{p}}} < \hat{\hat{\mathsf{p}}}'$.
  Luego, como $\hat{\mathsf{p}}' = \hat{\hat{\mathsf{p}}}'$ tenemos $\hat{\mathsf{p}} < \hat{\hat{\mathsf{p}}} < \hat{\mathsf{p}}'$. Es decir, entre las posiciones de los matchings $\hat{\mathsf{p}}$ y $\hat{\mathsf{p}}'$ esta la posición de $\hat{\mathsf{p}}$, que por **M1** es $s_{\hat{\mathsf{p}}} \in \ell(\mathsf{p})$.
  Sin embargo, por **M4** tenemos $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{p}}')} \cap \gamma(\mathsf{p}, \mathsf{p}') = \emptyset$ y como por (**2.2**) ocurre $\ell(\mathsf{p}) \subseteq \gamma(\mathsf{p}, \mathsf{p}')$ entonces $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{p}}')} \cap \ell(\mathsf{p}) = \emptyset$. Es decir, entre las posiciones de los matchings $\hat{\mathsf{p}}$ y $\hat{\mathsf{p}}'$ no hay ningún evento de p lo cual es absurdo con la afirmación anterior.

Si es (**B**), entonces $\hat{\mathsf{p}} > \hat{\hat{\mathsf{p}}}$. En este caso también se llega a un absurdo aplicando el mismo razonamiento que en (A) pero intercambiando $\hat{\cdot}$ y $\hat{\hat{\cdot}}$.

En (**3**) tenemos $FirstOf(\mathsf{p}) \subseteq \lessdot\mathsf{p}$ (cuando $\mathsf{p} \in FirstRep$)

Como $\mathsf{p} \in FirstRep$, por **M8**, tenemos que $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in FirstOf(\mathsf{p})\}$ y $\hat{\hat{\mathsf{p}}} = \min\{\hat{\hat{\mathsf{r}}}/\mathsf{r} \in FirstOf(\mathsf{p})\}$.

Ahora, por como fue elegido p, $\forall \mathsf{q} : \mathsf{q} \lessdot \mathsf{p}$ tenemos que $\hat{\mathsf{q}} = \hat{\hat{\mathsf{q}}}$. Luego, dado que $FirstOf(\mathsf{p}) \subseteq \lessdot\mathsf{p}$ ocurre que $\forall \mathsf{r} \in FirstOf(\mathsf{p})$ tenemos que $\hat{\mathsf{r}} = \hat{\hat{\mathsf{r}}}$, por lo tanto:

$$\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in FirstOf(\mathsf{p})\} = \min\{\hat{\hat{\mathsf{r}}}/\mathsf{r} \in FirstOf(\mathsf{p})\} = \hat{\hat{\mathsf{p}}}$$

Lo cual es absurdo con la suposición de que $\hat{\mathsf{p}} \neq \hat{\hat{\mathsf{p}}}$.

Por último, en (**4**) es fácil ver que también se llega a un absurdo siguiendo el mismo razonamiento de (**3**). Por lo tanto es incorrecta la suposición $\hat{\cdot} \neq \hat{\hat{\cdot}}$ y necesariamente $\hat{\cdot} = \hat{\hat{\cdot}}$.

□

A.6 Especialización de los antiescenarios

Propiedad A.6.1 (Especialización de los antiescenarios)**.** Sean $\mathcal{S}_0$ y $\mathcal{S}_1$ dos escenarios tal que $\mathcal{S}_1 <:: \mathcal{S}_0$ por un ranking $\lessdot$ sobre P_1, y $\mathcal{N}_{\mathcal{S}_1}$ un antiescenario generado por las reglas para $\mathcal{S}_{1\ j=1\ldots k}$ entonces:

$\mathcal{N}_{\mathcal{S}_1} <: \mathcal{S}_0$, es decir, todo antiescenario especializa al antecedente.

Demostración. De acuerdo a las reglas, todo $\mathcal{N}_{\mathcal{S}_1}$ se corresponde con alguno de los siguientes escenarios:

- $\mathcal{N}^{\mathcal{S}_0 \not\rightsquigarrow \mathsf{p}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}}$
- $\mathcal{N}^{\neg\gamma_1(\mathsf{p},\infty)} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\gamma_1(\mathsf{p},\infty)}$
- $\mathcal{N}^{\neg\gamma_1(\mathbf{0},\mathsf{p})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\gamma_1(\mathbf{0},\mathsf{p})}$
- $\mathcal{N}^{\neg\delta_1(\mathbf{0},\mathsf{p})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\delta_1(\mathbf{0},\mathsf{p})}$
- $\mathcal{N}^{\mathsf{p} \not<_1 \mathsf{q}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not<_1 \mathsf{q}}$
- $\mathcal{N}^{\mathsf{p} \not\asymp_1 \mathsf{q}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not\asymp_1 \mathsf{q}}$
- $\mathcal{N}^{\neg\gamma_1(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\gamma_1(\mathsf{p},\mathsf{q})}$
- $\mathcal{N}^{\neg\delta_1(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\delta_1(\mathsf{p},\mathsf{q})}$
- $\mathcal{N}^{\mathsf{p} \not\asymp_1 \mathsf{q}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not\asymp_1 \mathsf{q}}$
- $\mathcal{N}^{\neg\gamma_1(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\gamma_1(\mathsf{p},\mathsf{q})}$
- $\mathcal{N}^{\neg\delta_1(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\delta_1(\mathsf{p},\mathsf{q})}$
- $\mathcal{N}^{\mathsf{p} \leftrightarrow \mathsf{p}} = \bigoplus_{\mathsf{q} \in FirstOf_2(\mathsf{p})} (\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{n} <_F \mathsf{q}}) \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathsf{p} \not\equiv \mathsf{n}}$
- $\mathcal{N}^{\mathsf{p} \leftrightarrow \mathsf{p}} = \bigoplus_{\mathsf{q} \in LastOf_2(\mathsf{p})} (\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} <_L \mathsf{n}}) \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathsf{p} \not\equiv \mathsf{n}}$
- $\mathcal{N}^{\mathcal{S}_0 \not\Rightarrow \mathsf{p}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\not\Rightarrow \mathsf{p}}$

de esta forma se puede afirmar que la construcción de toda antiescenario $\mathcal{N}_{\mathcal{S}_1}$ incluye la fusión con un escenario $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'}$ donde $\mathsf{p}' \in P_1$. Por la *Propiedad 4.1.2* se puede afirmar que $\mathcal{N}_{\mathcal{S}_1} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'}$.

Luego, por la *Propiedad A.8.1*, tenemos que $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'} <: \mathcal{S}_0$.

Finalmente, como $\mathcal{N}_{\mathcal{S}_1} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'}$ y $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'} <: \mathcal{S}_0$, podemos afirmar que $\mathcal{N}_{\mathcal{S}_1} <: \mathcal{S}_0$.

□

A.7 Especialización de los consecuentes

Propiedad A.7.1 (Especialización de los consecuentes)**.** Sean $\mathcal{S}_0$ y $\mathcal{S}_1$ dos escenarios tal que $\mathcal{S}_1 <:: \mathcal{S}_0$ por un ranking $\lessdot$ sobre P_1 para todo punto $\mathsf{p} \in P_1$, entonces:

$$\mathcal{S}_1 <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \quad \text{y} \quad \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <:: \mathcal{S}_0.$$

Demostración. Es fácil ver que $\mathcal{S}_1 \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} = \mathcal{S}_1$, luego, por la *Propiedad 4.1.2*, tenemos que $\mathcal{S}_1 <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$.

Dado que $\mathcal{S}_1 <:: \mathcal{S}_0$ por un ranking $\lessdot$, por la *Propiedad A.8.1*, tenemos que $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <: \mathcal{S}_0$. Ahora veamos que existe un ranking $\lessdot'$ por el cual esta especialización es determinística. Es decir, que: $\forall \mathsf{p}_0 \in P_0, \mathsf{p}_\mathcal{P} \in P_\mathcal{P} \setminus P_0 \;.\; \mathsf{p}_0 \lessdot' \mathsf{p}_\mathcal{P}$ y $\lessdot' \mathsf{p}_\mathcal{P} \hookrightarrow \mathsf{p}_\mathcal{P}$ (donde $\lessdot' \mathsf{p}_\mathcal{P} = \{\mathsf{p} \in P_\mathcal{P} \mid \mathsf{p} \lessdot' \mathsf{p}_\mathcal{P}\}$).

Tomemos $\lessdot'$ como el ranking $\lessdot$ restringido a los puntos de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$.

Sabemos que el ranking $\lessdot$ verifica la siguiente propiedad: $\forall \mathsf{p}_0 \in P_0, \mathsf{p}_1 \in P_1 \setminus P_0 \;.\; \mathsf{p}_0 \lessdot \mathsf{p}_1$ y $\lessdot \mathsf{p}_1 \hookrightarrow \mathsf{p}_1$ (donde $\lessdot \mathsf{p}_1 = \{\mathsf{p} \in P_1 \mid \mathsf{p} \lessdot \mathsf{p}_1\}$). Entonces, dado que $\forall \mathsf{p}_0 \in P_0, \mathsf{p}_1 \in P_1 \setminus P_0 \;.\; \mathsf{p}_0 \lessdot \mathsf{p}_1$ y como $\forall \mathsf{p}_\mathcal{P} \in P_\mathcal{P} \setminus P_0$ tenemos que $\mathsf{p}_\mathcal{P} \in P_1 \setminus P_0$ podemos afirmar que $\forall \mathsf{p}_0 \in P_0, \mathsf{p}_\mathcal{P} \in P_\mathcal{P} \setminus P_0 \;.\; \mathsf{p}_0 \lessdot' \mathsf{p}_\mathcal{P}$.

Es fácil ver que $\forall \mathsf{p}_\mathcal{P} \in P_\mathcal{P} \setminus P_0$ $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}_\mathcal{P}}$ y este último escenario corresponde a:

- $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}_\mathcal{P}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \hookrightarrow \mathsf{p}_\mathcal{P}}$ cuando $\mathsf{p}_\mathcal{P}$ es un punto concreto, y donde $\mathsf{p}_\mathcal{P}\ \mathsf{q} \in ld_\lessdot(\mathsf{p}_\mathcal{P})$.

  En este caso, $\mathsf{q} \in P_\mathcal{P}$ y por la definición de $\mathcal{P}^{\mathsf{q} \hookrightarrow \mathsf{p}_\mathcal{P}}$ tenemos que $\mathsf{q} \hookrightarrow \mathsf{p}_\mathcal{P}$.

  Además sabemos que $\lessdot \mathsf{p}_\mathcal{P} \hookrightarrow_{\mathcal{S}_1} \mathsf{p}_\mathcal{P}$, entonces existe al menos un punto $\mathsf{q}' \lessdot \mathsf{p}_\mathcal{P}$ tal que $\mathsf{q}' \hookrightarrow_{\mathcal{S}_1} \mathsf{p}_\mathcal{P}$. Dado que $\mathsf{q} = ld_\lessdot(\mathsf{p}_\mathcal{P})$ entonces q corresponde al mínimo de estos puntos y podemos afirmar que $\mathsf{q} \lessdot \mathsf{p}_\mathcal{P}$. Luego, como $\mathsf{q} \in P_\mathcal{P}$ entonces $\mathsf{q} \in \lessdot'$, y vale que $\mathsf{q} \lessdot' \mathsf{p}_\mathcal{P}$. Por lo tanto, $\mathsf{q} \in \lessdot' \mathsf{p}_\mathcal{P}$ y, como vimos que $\mathsf{q} \hookrightarrow \mathsf{p}_\mathcal{P}$, podemos afirmar que $\lessdot' \mathsf{p}_\mathcal{P} \hookrightarrow \mathsf{p}_\mathcal{P}$.

- $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}_\mathcal{P}} = \bigoplus_{\mathsf{q} \in ld_\lessdot(\mathsf{p}_\mathcal{P})} (\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \hookrightarrow \mathsf{p}_\mathcal{P}})$ cuando $\mathsf{p}_\mathcal{P}$ es un punto representativo.

  Si $\mathsf{p}_\mathcal{P}$ es un representativo-primero entonces $ld_\lessdot(\mathsf{p}_\mathcal{P}) = FirstOf_1(\mathsf{p}_\mathcal{P})$.

  Por la definición de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}_\mathcal{P}}$, $\forall \mathsf{q} \in FirstOf_1(\mathsf{p}_\mathcal{P})$ entonces $\mathsf{q} \in P_\mathcal{P}$, y es fácil ver que $FirstOf_1(\mathsf{p}_\mathcal{P}) = FirstOf_\mathcal{P}(\mathsf{p}_\mathcal{P})$, luego tenemos que $FirstOf_\mathcal{P}(\mathsf{p}_\mathcal{P}) \hookrightarrow \mathsf{p}_\mathcal{P}$.

  Además, sabemos que $\lessdot \mathsf{p}_\mathcal{P} \hookrightarrow_{\mathcal{S}_1} \mathsf{p}_\mathcal{P}$, entonces $FirstOf_1(\mathsf{p}_\mathcal{P}) \subseteq \lessdot \mathsf{p}_\mathcal{P}$ y $\forall \mathsf{q} \in FirstOf_1(\mathsf{p}_\mathcal{P})\ \mathsf{q} \lessdot \mathsf{p}_\mathcal{P}$. Luego, como $\mathsf{q} \in P_\mathcal{P}$ entonces $\mathsf{q} \in \lessdot'$, y vale que $\mathsf{q} \lessdot' \mathsf{p}_\mathcal{P}$. Por lo tanto, dado que $FirstOf_1(\mathsf{p}_\mathcal{P}) = FirstOf_\mathcal{P}(\mathsf{p}_\mathcal{P})$, tenemos que $FirstOf_\mathcal{P}(\mathsf{p}_\mathcal{P}) \subseteq \lessdot' \mathsf{p}_\mathcal{P}$ y, como vimos que $FirstOf_\mathcal{P}(\mathsf{p}_\mathcal{P}) \hookrightarrow \mathsf{p}_\mathcal{P}$, podemos afirmar que $\lessdot' \mathsf{p}_\mathcal{P} \hookrightarrow \mathsf{p}_\mathcal{P}$.

  Por último, si $\mathsf{p}_\mathcal{P}$ corresponde a un representativo-último utilizando la argumentación anterior, pero con $LastOf$ (en lugar de $FirstOf$), también podemos afirmar que $\lessdot' \mathsf{p}_\mathcal{P} \hookrightarrow \mathsf{p}_\mathcal{P}$.

□

A.8 Especialización de los escenarios del camino

Propiedad A.8.1 (Especialización de los escenarios del camino)**.** Sean $\mathcal{S}_0$ y $\mathcal{S}_1$ dos escenarios tal que $\mathcal{S}_1 <:: \mathcal{S}_0$ por un ranking $\lessdot$ sobre P_1 para todo punto $\mathsf{p} \in P_1$, entonces:

$\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <: \mathcal{S}_0$, es decir, todo escenario del camino especializa al antecedente.

Demostración. Todo escenario $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ se construye por la siguiente definición:

- $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} = \mathcal{S}_0$ cuando $\mathsf{p} \in P_0$
- $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \hookrightarrow \mathsf{p}}$ cuando p es un punto concreto en $P_1 \setminus P_0$ y $\mathsf{q} \in ld_\lessdot(\mathsf{p})$
- $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} = \bigoplus_{\mathsf{q} \in ld_\lessdot(\mathsf{p})} (\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \hookrightarrow \mathsf{p}})$ cuando p es un punto representativo en $P_1 \setminus P_0$

por lo tanto, todo escenario $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ o bien es igual a $\mathcal{S}_0$, o esta compuesto por la fusión de $\mathcal{S}_0$. Entonces, por la *Propiedad 4.1.2* tenemos que $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} <: \mathcal{S}_0$.

□

A.9 Completitud de Reglas para ECD

Lema A.9.1 (Completitud de Reglas para ECD)**.** Sea $\mathcal{C} = \langle \mathcal{S}_0, \{\mathcal{S}_i\}_{i=1...k} \rangle$ un ECD, sea σ una ejecución, y sea $\hat{\cdot}$ un matching entre $\mathcal{S}_0$ y σ para el cual no existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{S}_i$ y σ, para ningún $i \in \{1..k\}$ tal que $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$, entonces:

existe $\mathcal{N}_{\mathcal{C}} = \langle \mathcal{N}_{\mathcal{S}_1} \oplus \mathcal{N}_{\mathcal{S}_2} \oplus ... \oplus \mathcal{N}_{\mathcal{S}_k} \rangle$ un antiescenario final, donde cada $\mathcal{N}_{\mathcal{S}_j\, j=1...k}$ es un antiescenario generado por las reglas de $\mathcal{S}_j$, que verifica $\sigma \vDash \mathcal{N}_{\mathcal{C}}$.

Demostración.

Para todo $i \in \{1..k\}$, por hipótesis tenemos que no existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{S}_i$ y σ, tal que $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$. Además, por definición de ECD, tenemos que $\mathcal{S}_i <:: \mathcal{S}_0$. Entonces, por *Lema A.10.1*, sabemos que las reglas generan un antiescenario $\mathcal{N}_{\mathcal{S}_i}$ con un matching $\hat{\hat{\cdot}}$ entre $\mathcal{N}_{\mathcal{S}_i}$ y σ, donde $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

Luego, por la *Propiedad A.6.1*, como $\mathcal{N}_{\mathcal{S}_i} <: \mathcal{S}_0$ tenemos $P_{\mathcal{S}_0} \subseteq P_{\mathcal{N}_{\mathcal{S}_i}}$. Además, como $\mathcal{C}$ es un escenario condicional, tenemos $P_{\mathcal{S}_i} \cap P_{\mathcal{S}_n} = P_{\mathcal{S}_0}$ para $i \neq n \in \{1...k\}$. Por lo tanto, $P_{\mathcal{N}_{\mathcal{S}_i}} \cap P_{\mathcal{N}_{\mathcal{S}_n}} = P_{\mathcal{S}_0}$ y entonces mediante la *Propiedad A.13.1* es fácil ver que $\mathcal{N}_{\mathcal{C}} = \langle \mathcal{N}_{\mathcal{S}_1} \oplus \mathcal{N}_{\mathcal{S}_2} \oplus ... \oplus \mathcal{N}_{\mathcal{S}_k} \rangle$ verifica $\sigma \vDash \mathcal{N}_{\mathcal{C}}$.

□

A.10 Completitud de Reglas para un consecuente

Lema A.10.1 (Completitud de Reglas para un consecuente)**.** Sean $\mathcal{S}_1$ y $\mathcal{S}_0$ dos escenarios donde $\mathcal{S}_1 <:: \mathcal{S}_0$, sea $\sigma = \langle s, \tau \rangle$ una ejecución, y sea $\hat{\cdot}$ un matching entre $\mathcal{S}_0$ y σ para el cual no existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{S}_1$ y σ con $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$, entonces:

existe $\mathcal{N}_{\mathcal{S}_1}$ un antiescenario generado por las reglas de $\mathcal{S}_1$, con un matching $\hat{\hat{\cdot}}$ entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

Demostración.

Tenemos que $\hat{\cdot}$ es un matching entre $\mathcal{S}_0$ y σ para el cual no existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{S}_1$ y σ con $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.
Entonces podemos consideremos las siguientes alternativas:

- **(A)** Para algún punto $\mathsf{p} \in P_1$ no existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y σ, con $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.
- **(B)** o bien, para todo punto $\mathsf{p} \in P_1$ existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y σ, con $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

Si es (**A**), como no existe matching $\hat{\hat{\cdot}}$ entre $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y σ, donde $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$, por el *Lema A.11.1*, podemos afirmar que existe un antiescenario generado por las reglas $\mathcal{N}_{\mathcal{S}_1}$ con un matching $\hat{\hat{\cdot}}$ entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

Si es (**B**), definimos $\mathcal{P}_t$ como el escenario del camino desde $\mathcal{S}_0$ a todos los puntos de $\mathcal{S}_1$, es decir, $\mathcal{P}_t = \bigoplus_{\mathsf{q} \in P_1} \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$. También definimos $\hat{\hat{\cdot}}$ como un matching entre $\mathcal{P}_t$ y σ, con $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$ que, por la *Propiedad A.12.1*, sabemos existe. Es importante observar que, por la *Propiedad 3.1.2*, $\forall \mathsf{p} \in P_1$ se tiene que $\hat{\hat{\cdot}}|_{\mathsf{p}}$ es un matching entre $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y σ.

Luego, por hipótesis, sabemos que $\hat{\hat{\cdot}}$ no es matching entre $\mathcal{S}_1$ y σ. Por lo tanto, $\hat{\hat{\cdot}}$ no verifica alguna de las siguientes condiciones de matching:

M1: $s_{\hat{\hat{\mathsf{p}}}} \in \ell_1(\mathsf{p})$; Como la condción no se verifica tenemos $s_{\hat{\hat{\mathsf{p}}}} \notin \ell_1(\mathsf{p})$.
Veamos los siguientes casos:

– $\mathsf{p} \in (P_1 \setminus P_0)$. Como $\mathsf{p} \in P_{\mathcal{P}_t}$ y $\hat{\cdot}$ es matching para $\mathcal{P}_t$, entonces la condición $\mathbf{M1}_{\mathcal{P}_t}$ se cumple para p. Es decir: $s_{\hat{\mathsf{p}}} \in \ell_{\mathcal{P}_t}(\mathsf{p})$

Luego, como $\ell_1(\mathsf{p}) = \ell_{\mathcal{P}_t}(\mathsf{p})$, podemos afirmar que la condición $\mathbf{M1}_{\mathcal{S}_1}$ se cumple para p.

– $\mathsf{p} \in P_0$. Dado que $\hat{\cdot}$ es matching para $\mathcal{S}_0$, entonces, la condición $\mathbf{M1}_{\mathcal{S}_0}$ se cumple para p. Es decir: $s_{\hat{\mathsf{p}}} \in \ell_0(\mathsf{p})$

Luego, si $\ell_1(\mathsf{p}) = \ell_0(\mathsf{p})$, como $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$, entonces el matching $\hat{\cdot}$ cumple la condición $\mathbf{M1}_{\mathcal{S}_1}$ para p. Por lo tanto, tenemos $\ell_1(\mathsf{p}) \neq \ell_0(\mathsf{p})$, y como $\ell_1(\mathsf{p}) \subseteq \ell_0(\mathsf{p})$, entonces $\ell_1(\mathsf{p}) \subsetneq \ell_0(\mathsf{p})$ y podemos elegir $\mathcal{N}_{\mathcal{S}_1}$ como $\mathcal{N}^{\mathcal{S}_0 \not\rightsquigarrow \mathsf{p}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\not\rightsquigarrow \mathsf{p}}$.

Veamos que hay un matching para este antiescenario. Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_{\mathsf{p}}$ es matching para $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$.

Ahora, comprobemos que $\hat{\cdot}|_{\{\mathsf{p}\}}$ es matching para $\mathcal{P}^{\not\rightsquigarrow \mathsf{p}}$:

* **M1**: $s_{\hat{\mathsf{p}}} \in \ell_{\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}}(\mathsf{p})$
  Dado que $\ell_0(\mathsf{p}) \setminus \ell_1(\mathsf{p}) \neq \emptyset$, tenemos $\ell_0(\mathsf{p}) \neq \emptyset$. Luego, $\mathbf{M1}_{\mathcal{S}_0}$ se cumple para $\hat{\cdot}$, y como $\ell_0(\mathsf{p}) \neq \emptyset$, sabemos que $s_{\hat{\mathsf{p}}} \in \ell_0(\mathsf{p})$. Por el otro lado, $\mathbf{M1}_{\mathcal{S}_1}$ no se cumple para $\hat{\cdot}$, entonces o bien $\ell_1(\mathsf{p}) = \emptyset$ o bien $s_{\hat{\mathsf{p}}} \notin \ell_1(\mathsf{p})$. Por lo tanto, podemos afirmar:
  $$s_{\hat{\mathsf{p}}} \in \ell_0(\mathsf{p}) \setminus \ell_1(\mathsf{p})$$
  y como $\ell_{\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}}(\mathsf{p}) = \ell_0(\mathsf{p}) \setminus \ell_1(\mathsf{p}) \neq \emptyset$ tenemos que $\mathbf{M1}_{\mathcal{P}^{\not\rightsquigarrow \mathsf{p}}}$ se verifica.
* **M2-8**: no aplican a los puntos del escenario.

Finalmente, por la *Propiedad A.14.1*, sabemos que $\hat{\cdot}_{\mathsf{p}}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}_{\mathsf{p}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

M2: si $\mathsf{p} \not\equiv_1 \mathsf{q}$ entonces $\hat{\mathsf{p}} \neq \hat{\mathsf{q}}$; Como la condición no se verifica, tenemos $\hat{\mathsf{p}} = \hat{\mathsf{q}}$.

Veamos que si $(\mathsf{p}, \mathsf{q}) \in (\neq_p \cup \neq_q)$ entonces, como **M2** no se cumple, $\hat{\cdot}$ no sería matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ o bien de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, lo que es absurdo con la definición de $\hat{\cdot}$ en (**B**).

Luego, como $(\mathsf{p}, \mathsf{q}) \notin (\neq_p \cup \neq_q)$, elegimos $\mathcal{N}_{\mathcal{S}_1}$ como $\mathcal{N}^{\mathsf{p} \not\equiv_1 \mathsf{q}} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{p} \not\equiv_1 \mathsf{q}}$.

Ahora veamos que hay un matching para este antiescenario. Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_{\mathsf{pq}}$ es matching para $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$.

Definamos una extensión de $\hat{\cdot}$ como el mapping $\hat{\cdot}_{\mathsf{u}}$:

– $\hat{\cdot}_{\mathsf{u}}(\mathsf{m}) = \hat{\mathsf{m}}$ si $\mathsf{m} \neq \mathsf{u}$
– $\hat{\cdot}_{\mathsf{u}}(\mathsf{m}) = \hat{\mathsf{p}}$ si $\mathsf{m} = \mathsf{u}$

De esta forma $\hat{\cdot}_{\mathsf{u}}|_{\mathsf{pq}}$ es matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$ donde $\hat{\cdot}_{\mathsf{u}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

Ahora, probemos que $\hat{\cdot}_{\mathsf{u}}|_{\{\mathsf{p},\mathsf{q}\}}$ es matching de $\mathcal{P}^{\mathsf{p} \not\equiv_1 \mathsf{q}}$:

– **M1**: aplica para u, p y q. Si esta condición no se cumple para p o q, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
  Ahora si u no verifica **M1** tenemos $s_{\hat{\cdot}_{\mathsf{u}}(\mathsf{u})} \notin \ell(\mathsf{u})$. En ese caso, como $s_{\hat{\cdot}_{\mathsf{u}}(\mathsf{u})} = s_{\hat{\cdot}_{\mathsf{u}}(\mathsf{p})} = s_{\hat{\cdot}_{\mathsf{u}}(\mathsf{q})}$ y $\ell(\mathsf{u}) = \ell(\mathsf{p}) \cup \ell(\mathsf{q})$, entonces $s_{\hat{\cdot}_{\mathsf{u}}(\mathsf{p})} \notin \ell(\mathsf{p})$ y $s_{\hat{\cdot}_{\mathsf{u}}(\mathsf{q})} \notin \ell(\mathsf{q})$. Ahora, como p y q cumplen **M1** llegamos a un absurdo, por lo tanto, **M1** también se verifica para u.
– **M2-7**: no aplican a los puntos del escenario.
– **M8**: aplica para u. Como tenemos que $\hat{\mathsf{p}} = \hat{\mathsf{q}}$ en ambos casos la condición se verifica:
  $\hat{\cdot}_{\mathsf{u}}(\mathsf{u}) = \min\{\hat{\cdot}_{\mathsf{u}}(\mathsf{r})/\mathsf{r} \in \mathit{FirstOf}(\mathsf{u})\} = \min\{\hat{\cdot}_{\mathsf{u}}(\mathsf{r})/\mathsf{r} \in \{\mathsf{p},\mathsf{q}\}\} = \min\{\hat{\cdot}_{\mathsf{u}}(\mathsf{p}), \hat{\cdot}_{\mathsf{u}}(\mathsf{q})\} = \min\{\hat{\mathsf{p}}, \hat{\mathsf{q}}\} = \hat{\mathsf{p}}$
  $\hat{\cdot}_{\mathsf{u}}(\mathsf{u}) = \max\{\hat{\cdot}_{\mathsf{u}}(\mathsf{r})/\mathsf{r} \in \mathit{LastOf}(\mathsf{u})\} = \max\{\hat{\cdot}_{\mathsf{u}}(\mathsf{r})/\mathsf{r} \in \{\mathsf{p},\mathsf{q}\}\} = \max\{\hat{\cdot}_{\mathsf{u}}(\mathsf{p}), \hat{\cdot}_{\mathsf{u}}(\mathsf{q})\} = \max\{\hat{\mathsf{p}}, \hat{\mathsf{q}}\} = \hat{\mathsf{p}}$

Entonces, por la *Propiedad A.14.1* sabemos que $\hat{\cdot}_{\mathsf{u}}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}_{\mathsf{u}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

M3: si $\mathsf{p} <_1 \mathsf{q}$ entonces $\hat{\mathsf{p}} < \hat{\mathsf{q}}$; Como la condición no se verifica, tenemos $\hat{\mathsf{p}} = \hat{\mathsf{q}}$ o bien $\hat{\mathsf{p}} > \hat{\mathsf{q}}$.

- Si $\hat{\mathsf{p}} = \hat{\mathsf{q}}$, podemos generar el antiescenario $\mathcal{N}_{\mathcal{S}_1}$ como $\mathcal{N}^{\mathsf{p}\divideontimes_1\mathsf{q}} = \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}} \oplus \mathcal{P}^{\mathsf{p}\divideontimes_1\mathsf{q}}$. Luego, por la misma argumentación del caso **M2**, sabemos que existe un matching $\hat{\cdot}_{\mathsf{u}}|_{\mathcal{N}_{\mathcal{S}_1}}$ entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}_{\mathsf{u}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

- Si $\hat{\mathsf{p}} > \hat{\mathsf{q}}$, veamos que si $(\mathsf{p},\mathsf{q}) \in (<_p \cup <_q)$ entonces, como **M3** no se cumple, $\hat{\cdot}$ no sería matching de $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}}$ o bien de $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$, lo que es absurdo con la definición de $\hat{\cdot}$ en (**B**).
  Luego, como $(\mathsf{p},\mathsf{q}) \notin (<_p \cup <_q)$, elegimos $\mathcal{N}_{\mathcal{S}_1}$ como $\mathcal{N}^{\mathsf{p}\not<_1\mathsf{q}} = \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}} \oplus \mathcal{P}^{\mathsf{p}\not<_1\mathsf{q}}$.

  Veamos que hay un matching para este antiescenario. Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}}$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_{\mathsf{pq}}$ es matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$.

  Ahora, comprobemos que $\hat{\cdot}|_{\{\mathsf{p},\mathsf{q}\}}$ es matching para $\mathcal{P}^{\mathsf{p}\not<_1\mathsf{q}}$:

  * **M1**: aplica para p o q. Pero sino se cumple, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
  * **M2**: no aplica a los puntos del escenario.
  * **M3**: aplica para (q,p). Como $\mathsf{q} < \mathsf{p}$ hay que comprobar que $\hat{\mathsf{q}} < \hat{\mathsf{p}}$, lo cual se verifica dado que estamos en este caso.
  * **M4-8**: no aplican a los puntos del escenario.

  Finalmente, por la *Propiedad A.14.1*, sabemos que $\hat{\cdot}_{\mathsf{p}}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}_{\mathsf{p}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

M4: $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{q}})} \cap \gamma_1(\mathsf{p},\mathsf{q}) = \emptyset$; Como la condición no se verifica, tenemos $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{q}})} \cap \gamma_1(\mathsf{p},\mathsf{q}) \neq \emptyset$.

Veamos que si $\gamma_p(\mathsf{p},\mathsf{q}) = \gamma_1(\mathsf{p},\mathsf{q})$ o $\gamma_q(\mathsf{p},\mathsf{q}) = \gamma_1(\mathsf{p},\mathsf{q})$ entonces, como **M4** no se cumple, $\hat{\cdot}$ no sería matching de $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}}$ o bien de $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$, lo que es absurdo con la definición de $\hat{\cdot}$ en (**B**).

Luego, como $\gamma_1(\mathsf{p},\mathsf{q}) \neq \gamma_p(\mathsf{p},\mathsf{q})$ y $\gamma_1(\mathsf{p},\mathsf{q}) \neq \gamma_p(\mathsf{p},\mathsf{q})$, tenemos $(\mathsf{p},\mathsf{q}) \notin (\gamma_p \cup \gamma_q)$ y elegimos $\mathcal{N}_{\mathcal{S}_1}$ como:

- $\mathcal{N}^{\neg\gamma_1(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}} \oplus \mathcal{P}^{\neg\gamma_1(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p},\mathsf{q}) \in <_1$.

  Veamos que hay un matching para este antiescenario. Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}}$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_{\mathsf{pq}}$ es matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$.
  Definamos una extensión de $\hat{\cdot}$ como el mapping $\hat{\cdot}_{\mathsf{n}}$:

  * $\hat{\cdot}_{\mathsf{n}}(\mathsf{m}) = \hat{\mathsf{m}}$ si $\mathsf{m} \neq \mathsf{n}$
  * $\hat{\cdot}_{\mathsf{n}}(\mathsf{m}) = \min\{i/\ \hat{\mathsf{p}} < i < \hat{\mathsf{q}}\ \text{ y } s_{\hat{i}} \in \gamma_1(\mathsf{p},\mathsf{q})\}$ si $\mathsf{m} = \mathsf{n}$

  De esta forma $\hat{\cdot}_{\mathsf{n}}|_{\mathsf{pq}}$ es matching de $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$ donde $\hat{\cdot}_{\mathsf{n}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

  Ahora, probemos que $\hat{\cdot}_{\mathsf{n}}|_{\{\mathsf{p},\mathsf{q},\mathsf{n}\}}$ es matching de $\mathcal{P}^{\neg\gamma_1(\mathsf{p},\mathsf{q})}$:

  * **M1**: aplica para u, p y q. Si esta condición no se cumple para p o q, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
    Ahora veamos para n. Como $\ell(\mathsf{n}) = \gamma_1(\mathsf{p},\mathsf{q}) \neq \emptyset$ tenemos que verificar que $s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{n})} \in \ell(\mathsf{n}) = \gamma_1(\mathsf{p},\mathsf{q})$, por lo tanto, se cumple.
  * **M2**: no aplica a los puntos del escenario.
  * **M3**: aplica para $(\mathsf{p},\ \mathsf{n})$ y $(\mathsf{n},\ \mathsf{q})$. Dado que $\hat{\cdot}_{\mathsf{n}}(\mathsf{p}) = \hat{\mathsf{p}} < \hat{\cdot}_{\mathsf{n}}(\mathsf{n}) < \hat{\mathsf{q}} = \hat{\cdot}_{\mathsf{n}}(\mathsf{q})$ en ambos casos se verifica.
  * **M4-8**: no aplican a los puntos del escenario.

  Finalmente, por la*Propiedad A.14.1*, sabemos que $\hat{\cdot}_{\mathsf{n}}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}_{\mathsf{n}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

– $\mathcal{N}^{\neg\gamma_1(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\gamma_1(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p},\mathsf{q}) \in \not\equiv_1$

Veamos que hay un matching para este antiescenario. Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_{\mathsf{pq}}$ es matching para $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$.

Definamos una extensión de $\hat{\cdot}$ como el mapping $\hat{\cdot}_\mathsf{i}$:

* $\hat{\cdot}_\mathsf{i}(\mathsf{m}) = \hat{\mathsf{m}}$ si $\mathsf{m} \notin \{\mathsf{f},\mathsf{l},\mathsf{i}\}$
* $\hat{\cdot}_\mathsf{i}(\mathsf{m}) = \min\{\hat{\mathsf{p}},\hat{\mathsf{q}}\}$ si $\mathsf{m} = \mathsf{f}$
* $\hat{\cdot}_\mathsf{i}(\mathsf{m}) = \max\{\hat{\mathsf{p}},\hat{\mathsf{q}}\}$ si $\mathsf{m} = \mathsf{l}$
* $\hat{\cdot}_\mathsf{i}(\mathsf{m}) = \min\{j/\ \min\{\hat{\mathsf{p}},\hat{\mathsf{q}}\} < j < \max\{\hat{\mathsf{p}},\hat{\mathsf{q}}\}$ y $s_{\hat{\mathsf{j}}} \in \gamma_1(\mathsf{p},\mathsf{q})\}$ si $\mathsf{m} = \mathsf{i}$

De esta forma $\hat{\cdot}_\mathsf{n}|_{\mathsf{pq}}$ es matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$ donde $\hat{\cdot}_\mathsf{n}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

Ahora, probemos que $\hat{\cdot}_\mathsf{n}|_{\{\mathsf{p},\mathsf{q},\mathsf{f},\mathsf{l},\mathsf{i}\}}$ es matching de $\mathcal{P}^{\neg\gamma_1(\mathsf{p},\mathsf{q})}$:

* **M1**: aplica para f, l, i, p y q. Si esta condición no se cumple para p o q, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
  Ahora si f no verifica **M1** tenemos $s_{\hat{\cdot}_\mathsf{i}(\mathsf{f})} \notin \ell(\mathsf{f})$. Luego, como $\ell(\mathsf{f}) = \ell(\mathsf{p}) \cup \ell(\mathsf{q})$, sabemos que $s_{\hat{\cdot}_\mathsf{i}(\mathsf{f})} \notin \ell(\mathsf{p}) \cup \ell(\mathsf{q})$. Ahora, como $s_{\hat{\cdot}_\mathsf{i}(\mathsf{f})} = s_{\hat{\cdot}_\mathsf{i}(\mathsf{p})}$ o bien $s_{\hat{\cdot}_\mathsf{i}(\mathsf{f})} = s_{\hat{\cdot}_\mathsf{i}(\mathsf{q})}$ entonces $s_{\hat{\cdot}_\mathsf{i}(\mathsf{p})} \notin \ell(\mathsf{p})$ o bien $s_{\hat{\cdot}_\mathsf{i}(\mathsf{q})} \notin \ell(\mathsf{q})$. Pero como p y q cumplen **M1** llegamos a un absurdo. Por lo tanto, **M1** también se verifica para f. Utlizando el mismo razonamiento se prueba que l tambien cumple con **M1**.

  Ahora veamos que se verifica para i. Como $\ell(\mathsf{i}) = \gamma_1(\mathsf{p},\mathsf{q}) \neq \emptyset$ tenemos que verificar que $s_{\hat{\cdot}_\mathsf{n}(\mathsf{i})} \in \ell(\mathsf{i}) = \gamma_1(\mathsf{p},\mathsf{q})$, por lo tanto se cumple.
* **M2**: no aplica a los puntos del escenario.
* **M3**: aplica para (f, i) y (i, l). Dado que $\hat{\cdot}_\mathsf{i}(\mathsf{f}) = \min\{\hat{\mathsf{p}},\hat{\mathsf{q}}\} < \hat{\cdot}_\mathsf{i}(\mathsf{i}) < \max\{\hat{\mathsf{p}},\hat{\mathsf{q}}\} = \hat{\cdot}_\mathsf{i}(\mathsf{l})$ en ambos casos se verifica.
* **M4-7**: no aplican a los puntos del escenario.
* **M8**: aplica para f y l. En ambos casos la condición se verifica:
  $\hat{\cdot}_\mathsf{i}(\mathsf{f}) = \min\{\hat{\cdot}_\mathsf{i}(\mathsf{r})/\mathsf{r} \in \mathit{FirstOf}(\mathsf{f})\} = \min\{\hat{\cdot}_\mathsf{i}(\mathsf{r})/\mathsf{r} \in \{\mathsf{p},\mathsf{q}\}\} = \min\{\hat{\cdot}_\mathsf{i}(\mathsf{p}),\hat{\cdot}_\mathsf{i}(\mathsf{q})\} = \min\{\hat{\mathsf{p}},\hat{\mathsf{q}}\}$
  $\hat{\cdot}_\mathsf{i}(\mathsf{l}) = \max\{\hat{\cdot}_\mathsf{i}(\mathsf{r})/\mathsf{r} \in \mathit{LastOf}(\mathsf{l})\} = \max\{\hat{\cdot}_\mathsf{i}(\mathsf{r})/\mathsf{r} \in \{\mathsf{p},\mathsf{q}\}\} = \max\{\hat{\cdot}_\mathsf{i}(\mathsf{p}),\hat{\cdot}_\mathsf{i}(\mathsf{q})\} = \max\{\hat{\mathsf{p}},\hat{\mathsf{q}}\}$

Finalmente, por la *Propiedad A.14.1*, sabemos que $\hat{\cdot}_\mathsf{i}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}_\mathsf{i}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

M5: $s_{\hat{\mathsf{p}})} \cap \gamma_1(\mathbf{0},\mathsf{p}) = s_{(\hat{\mathsf{p}}} \cap \gamma_1(\mathsf{p},\infty) = \emptyset$; Como la condición no se verifica, tenemos al menos alguno de los siguientes casos:

– $s_{\hat{\mathsf{p}})} \cap \gamma_1(\mathbf{0},\mathsf{p}) \neq \emptyset$

Veamos que si $\gamma_p(\mathbf{0},\mathsf{p}) = \gamma_1(\mathbf{0},\mathsf{p})$ entonces, como **M5** no se cumple, $\hat{\cdot}$ no sería matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, lo que es absurdo con la definición de $\hat{\cdot}$ en (**B**).
Luego, como $\gamma_p(\mathbf{0},\mathsf{p}) \neq \gamma_1(\mathbf{0},\mathsf{p})$, tenemos $(\mathbf{0},\mathsf{p}) \in \gamma_1 \smallsetminus \gamma_p$ y entonces elegimos $\mathcal{N}_{\mathcal{S}_1}$ como $\mathcal{N}^{\neg\gamma_1(\mathbf{0},\mathsf{p})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\gamma_1(\mathbf{0},\mathsf{p})}$.

Veamos que hay un matching para este antiescenario. Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_{\mathsf{p}}$ es matching para $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$.

Definamos una extensión de $\hat{\cdot}$ como el mapping $\hat{\cdot}_\mathsf{n}$:

* $\hat{\cdot}_\mathsf{n}(\mathsf{m}) = \hat{\mathsf{m}}$ si $\mathsf{m} \neq \mathsf{n}$
* $\hat{\cdot}_\mathsf{n}(\mathsf{m}) = \max\{i/i < \hat{\mathsf{p}}$ y $s_{\hat{\mathsf{i}}} \in \gamma_1(\mathbf{0},\mathsf{p})\}$ si $\mathsf{m} = \mathsf{n}$

De esta forma $\hat{\cdot}_\mathsf{n}|_\mathsf{p}$ es matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ donde $\hat{\cdot}_\mathsf{n}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

Ahora, probemos que $\hat{\cdot}_\mathsf{n}|_{\{\mathsf{p},\mathsf{n}\}}$ es matching de $\mathcal{P}^{\neg\gamma_1(\mathbf{0},\mathsf{p})}$:

* **M1**: aplica para n y p. Si esta condición no se cumple para p, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
* Ahora veamos que se verifica para n. Como $\ell(\mathsf{n}) = \gamma_1(\mathbf{0},\mathsf{p}) \smallsetminus \gamma_p(\mathbf{0},\mathsf{p}) \neq \emptyset$ tenemos que verificar que $s_{\hat{\cdot}_\mathsf{n}(\mathsf{n})} \in \ell(\mathsf{n}) = \gamma_1(\mathbf{0},\mathsf{p}) \smallsetminus \gamma_p(\mathbf{0},\mathsf{p})$, por lo tanto se cumple.

* **M2**: no aplica a los puntos del escenario.
* **M3**: aplica para (n, p). Dado que $\hat{:}_{\mathsf{n}}(\mathsf{n}) < \hat{\mathsf{p}} = \hat{:}_{\mathsf{n}}(\mathsf{p})$ la condición se verifica.
* **M4-8**: no aplican a los puntos del escenario.

Finalmente, por la *Propiedad A.14.1*, sabemos que $\hat{:}_{\mathsf{n}}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{:}_{\mathsf{n}}|_{\mathcal{S}_0} = \hat{:}|_{\mathcal{S}_0}$.

– $s_{(\hat{\mathsf{p}}} \cap \gamma_1(\mathsf{p}, \infty) \neq \emptyset$

Veamos que si $\gamma_p(\mathsf{p}, \infty) = \gamma_1(\mathsf{p}, \infty)$ entonces, como **M5** no se cumple, $\hat{:}$ no sería matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, lo que es absurdo con la definición de $\hat{:}$ en (**B**).

Luego, como $\gamma_p(\mathsf{p}, \infty) \neq \gamma_1(\mathsf{p}, \infty)$, tenemos $(\mathsf{p}, \infty) \in \gamma_1 \smallsetminus \gamma_p$ y entonces elegimos $\mathcal{N}_{\mathcal{S}_1}$ como $\mathcal{N}^{\neg\gamma_1(\mathsf{p},\infty)} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\gamma_1(\mathsf{p},\infty)}$.

Veamos que hay un matching para este antiescenario. Dado que $\hat{:}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{:}|_{\mathsf{p}}$ es matching para $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$.

Definamos una extensión de $\hat{:}$ como el mapping $\hat{:}_{\mathsf{n}}$:

* $\hat{:}_{\mathsf{n}}(\mathsf{m}) = \hat{\mathsf{m}}$ si $\mathsf{m} \neq \mathsf{n}$
* $\hat{:}_{\mathsf{n}}(\mathsf{m}) = \min\{i / \hat{\mathsf{p}} < i \text{ y } s_{\hat{i}} \in \gamma_1(\mathsf{p}, \infty)\}$ si $\mathsf{m} = \mathsf{n}$

De esta forma $\hat{:}_{\mathsf{n}}|_{\mathsf{p}}$ es matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ donde $\hat{:}_{\mathsf{n}}|_{\mathcal{S}_0} = \hat{:}|_{\mathcal{S}_0}$.

Ahora, probemos que $\hat{:}_{\mathsf{n}}|_{\{\mathsf{p},\mathsf{n}\}}$ es matching de $\mathcal{P}^{\neg\gamma_1(\mathsf{p},\infty)}$:

* **M1**: aplica para n y p. Si esta condición no se cumple para p, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.

  Ahora veamos que se verifica para n. Como $\ell(\mathsf{n}) = \gamma_1(\mathsf{p}, \infty) \smallsetminus \gamma_p(\mathsf{p}, \infty) \neq \emptyset$ tenemos que verificar que $s_{\hat{:}_{\mathsf{n}}(\mathsf{n})} \in \ell(\mathsf{n}) = \gamma_1(\mathsf{p}, \infty) \smallsetminus \gamma_p(\mathsf{p}, \infty)$, por lo tanto se cumple.
* **M2**: no aplica a los puntos del escenario.
* **M3**: aplica para (p, n). Dado que $\hat{\mathsf{p}} = \hat{:}_{\mathsf{n}}(\mathsf{p}) < \hat{:}_{\mathsf{n}}(\mathsf{n})$ la condición se verifica.
* **M4-8**: no aplican a los puntos del escenario.

Finalmente, por la *Propiedad A.14.1*, sabemos que $\hat{:}_{\mathsf{n}}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{:}_{\mathsf{n}}|_{\mathcal{S}_0} = \hat{:}|_{\mathcal{S}_0}$.

M6: $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \vDash \delta_1(\mathsf{p}, \mathsf{q})$; Como la condición no se verifica, tenemos $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \nvDash \delta_1(\mathsf{p}, \mathsf{q})$.
Elegimos $\mathcal{N}_{\mathcal{S}_1}$ como:

– $\mathcal{N}^{\neg\delta_1(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\delta_1(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p}, \mathsf{q}) \in <_1$.

Ahora veamos que hay un matching para este antiescenario. Dado que $\hat{:}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{:}|_{\mathsf{pq}}$ es matching para $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$.

Ahora, probemos que $\hat{:}|_{\{\mathsf{p},\mathsf{q}\}}$ es matching de $\mathcal{P}^{\neg\delta_1(\mathsf{p},\mathsf{q})}$:

* **M1**: aplica para p y q. Si esta condición no se cumple, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
* **M2**: no aplica a los puntos del escenario.
* **M3**: aplica para (p, q). Si esta condición no se cumple, ya vimos que por el caso **M3** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
* **M4-5**: no aplican a los puntos del escenario.
* **M6**: aplica para (p, q). Como sabemos que $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \nvDash \delta_1(\mathsf{p}, \mathsf{q})$ entonces $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \vDash \neg\delta_1(\mathsf{p}, \mathsf{q}) = \delta(\mathsf{p}, \mathsf{q})$ por lo tanto se cumple.
* **M7-8**: no aplican a los puntos del escenario.

Entonces, por la *Propiedad A.14.1* sabemos que $\hat{:}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{:}|_{\mathcal{S}_0} = \hat{:}|_{\mathcal{S}_0}$.

– $\mathcal{N}^{\neg\delta_1(\mathsf{p},\mathsf{q})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\neg\delta_1(\mathsf{p},\mathsf{q})}$ cuando $(\mathsf{p},\mathsf{q}) \in \not\equiv_1$

Ahora veamos que hay un matching para este antiescenario. Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_{\mathsf{pq}}$ es matching para $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$.

Ahora, probemos que $\hat{\cdot}|_{\{\mathsf{p},\mathsf{q}\}}$ es matching de $\mathcal{P}^{\neg\delta_1(\mathsf{p},\mathsf{q})}$:

* **M1**: aplica para p y q. Si esta condición no se cumple, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
* **M2**: aplica para (p, q). Si esta condición no se cumple, ya vimos que por el caso **M2** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
* **M3-5**: no aplican a los puntos del escenario.
* **M6**: aplica para (p, q). Como sabemos que $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \not\models \delta_1(\mathsf{p},\mathsf{q})$ entonces $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \models \neg\delta_1(\mathsf{p},\mathsf{q}){=}\delta(\mathsf{p},\mathsf{q})$ por lo tanto se cumple.
* **M7-8**: no aplican a los puntos del escenario.

Entonces, por la *Propiedad A.14.1* sabemos que $\hat{\cdot}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

M7: $\Delta(\tau_{\hat{\mathsf{p}}]}) \models \delta_1(\mathbf{0},\mathsf{p})$; Como la condición no se verifica, tenemos $\Delta(\tau_{\hat{\mathsf{p}}]}) \not\models \delta_1(\mathbf{0},\mathsf{p})$.

Veamos que si $\delta_1(\mathbf{0},\mathsf{p}) = \delta_p(\mathbf{0},\mathsf{p})$ entonces, como **M7** no se cumple, $\hat{\cdot}$ no sería matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, lo que es absurdo con la definición de $\hat{\cdot}$ en (**B**).

Luego, como $\delta_p(\mathbf{0},\mathsf{q}) \neq \delta_p(\mathbf{0},\mathsf{q})$, tenemos $(\mathbf{0},\mathsf{p}) \in \delta_1 \subsetneq \delta_p$ y entonces elegimos $\mathcal{N}_{\mathcal{S}_1}$ como $\mathcal{N}^{\neg\delta_1(\mathbf{0},\mathsf{p})} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\neg\delta_1(\mathbf{0},\mathsf{p})}$.

Ahora veamos que hay un matching para este antiescenario. Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_{\mathsf{pq}}$ es matching para $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$.

Entonces probemos que $\hat{\cdot}|_{\{\mathsf{p}\}}$ es matching de $\mathcal{P}^{\neg\delta_1(\mathbf{0},\mathsf{p})}$:

– **M1**: aplica para p. Si esta condición no se cumple, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
– **M2-6**: no aplican a los puntos del escenario.
– **M7**: aplica para p. Como sabemos que $\Delta(\tau_{\hat{\mathsf{p}}]}) \not\models \delta_1(\mathbf{0},\mathsf{p})$ entonces $\Delta(\tau_{\hat{\mathsf{p}}]}) \models \neg\delta_1(\mathbf{0},\mathsf{p}){=}\delta(\mathbf{0},\mathsf{p})$ por lo tanto se cumple.
– **M8**: no aplica a los puntos del escenario.

Entonces, por la *Propiedad A.14.1* sabemos que $\hat{\cdot}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

M8: si $\mathsf{p} \in \mathit{FirstRep}_1$ (resp. $\mathit{LastRep}_1$) entonces $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_1(\mathsf{p})\}$ (resp. max y LastOf_1).
Como la condición no se verifica, tenemos al menos alguno de los siguientes casos:

– $\mathsf{p} \in \mathit{FirstRep}_1$ y $\hat{\mathsf{p}} \neq \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_1(\mathsf{p})\}$

Veamos que si $\mathit{FirstOf}_1(\mathsf{p}) = \mathit{FirstOf}_p(\mathsf{p})$ entonces, como **M8** no se cumple, $\hat{\cdot}$ no sería matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, lo que es absurdo con la definición de $\hat{\cdot}$ en (**B**).

Luego, como $\exists \mathsf{q}(\mathsf{p},\mathsf{q}) \in <_{F1} \setminus <_{Fp}$, elegimos $\mathcal{N}_{\mathcal{S}_1}$ como $\mathcal{N}^{\mathsf{p} \mapsto \mathsf{p}} = \bigoplus_{\mathsf{q} \in \mathit{FirstOf}_1(\mathsf{p})} (\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{n} <_F \mathsf{q}}) \oplus \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus \mathcal{P}^{\mathsf{p} \not\equiv \mathsf{n}}$.

Veamos que hay un matching para este antiescenario. Sea $\mathcal{S}_f = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}} \oplus (\bigoplus_{\mathsf{q} \in \mathit{FirstOf}_1(\mathsf{p})} \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}})$. Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, y $\forall \mathsf{q} \in \mathit{FirstOf}_1(\mathsf{p})\ \mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_f$ es matching para $\mathcal{S}_f$.

Definamos una extensión de $\hat{\cdot}$ como el mapping $\hat{\cdot}_{\mathsf{n}}$:

* $\hat{\cdot}_{\mathsf{n}}(\mathsf{m}) = \hat{\mathsf{m}}$ si $\mathsf{m} \neq \mathsf{n}$
* $\hat{\cdot}_{\mathsf{n}}(\mathsf{m}) = \min\{\hat{\mathsf{r}}/\mathsf{r} \in FirstOf_1(\mathsf{p})\}$ si $\mathsf{m} = \mathsf{n}$

De esta forma $\hat{\cdot}_{\mathsf{n}}|_f$ es matching de $\mathcal{S}_f$.

Ahora, probemos que $\hat{\cdot}_{\mathsf{n}}$ es matching de $\mathcal{S}_n = \mathcal{P}^{\mathsf{p}\neq\mathsf{n}} \oplus (\bigoplus_{\mathsf{q}\in FirstOf_1(\mathsf{p})} \mathcal{P}^{\mathsf{n}<_F\mathsf{q}})$:

* **M1**: aplica para n y para $\mathsf{q} \in FirstOf_1(\mathsf{p})$. Si esta condición no se cumple para algún q, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
  Ahora si n no verifica **M1** tenemos $s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{n})} \notin \ell(\mathsf{n})$. Luego, como $\ell(\mathsf{n}) = \Sigma \cup \{\lambda\}$, sabemos que $s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{n})} \notin \Sigma \cup \{\lambda\}$. Ahora, como para algún $\mathsf{q} \in FirstOf_1(\mathsf{p})/s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{n})} = s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{q})}$ entonces $s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{q})} \notin \ell(\mathsf{q})$. Sin embargo vimos que todo q cumple **M1**, por lo tanto, llegamos a un absurdo y **M1** también se verifica para n.
* **M2**: aplica para (n, p). La condición se cumple:
$$\hat{\cdot}_{\mathsf{n}}(\mathsf{n}) = \min\{\hat{\mathsf{r}}/\mathsf{r} \in FirstOf_1(\mathsf{p})\} \neq \hat{\mathsf{p}} = \hat{\cdot}_{\mathsf{n}}(\mathsf{p})$$
* **M3**: no aplica a los puntos del escenario.
* **M4-7**: no aplican a los puntos del escenario.
* **M8**: aplica para n. La condición se verifica:
$\hat{\cdot}_{\mathsf{n}}(\mathsf{n}) = \min\{\hat{\cdot}_{\mathsf{n}}(\mathsf{r})/\mathsf{r} \in FirstOf(\mathsf{n})\} = \min\{\hat{\cdot}_{\mathsf{n}}(\mathsf{r})/\mathsf{r} \in FirstOf_1(\mathsf{p})\} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in FirstOf_1(\mathsf{p})\}$

Finalmente, como $\mathcal{N}_{\mathcal{S}_1} = \mathcal{S}_f \oplus \mathcal{S}_n$, por la *Propiedad A.14.1*, sabemos que $\hat{\cdot}_{\mathsf{n}}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}_{\mathsf{n}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

– $\mathsf{p} \in LastRep_1$ y $\hat{\mathsf{p}} \neq \max\{\hat{\mathsf{r}}/\mathsf{r} \in LastOf_1(\mathsf{p})\}$

Veamos que si $LastOf_1(\mathsf{p}) = LastOf_p(\mathsf{p})$ entonces, como **M8** no se cumple, $\hat{\cdot}$ no sería matching de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$, lo que es absurdo con la definición de $\hat{\cdot}$ en (**B**).

Luego, como $\exists \mathsf{q}(\mathsf{q}, \mathsf{p}) \in <_{L1} \setminus <_{Lp}$, elegimos $\mathcal{N}_{\mathcal{S}_1}$ como $\mathcal{N}^{\mathsf{p}\leftrightarrow\mathsf{p}} = \bigoplus_{\mathsf{q}\in LastOf_1(\mathsf{p})}(\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}} \oplus \mathcal{P}^{\mathsf{q}<_L\mathsf{n}}) \oplus \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathsf{p}\neq\mathsf{n}}$.

Veamos que hay un matching para este antiescenario. Sea $\mathcal{S}_l = \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus (\bigoplus_{\mathsf{q}\in LastOf_1(\mathsf{p})} \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}})$.
Dado que $\hat{\cdot}$ es un matching para $\mathcal{P}_t$, y $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}}$, y $\forall \mathsf{q} \in LastOf_1(\mathsf{p})$ $\mathcal{P}_t <: \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$, por la *Propiedad 3.1.2*, tenemos que $\hat{\cdot}|_l$ es matching para $\mathcal{S}_l$.
Definamos una extensión de $\hat{\cdot}$ como el mapping $\hat{\cdot}_{\mathsf{n}}$:

* $\hat{\cdot}_{\mathsf{n}}(\mathsf{m}) = \hat{\hat{\mathsf{m}}}$ si $\mathsf{m} \neq \mathsf{n}$
* $\hat{\cdot}_{\mathsf{n}}(\mathsf{m}) = \max\{\hat{\mathsf{r}}/\mathsf{r} \in LastOf_1(\mathsf{p})\}$ si $\mathsf{m} = \mathsf{n}$

De esta forma $\hat{\cdot}_{\mathsf{n}}|_l$ es matching de $\mathcal{S}_l$.

Ahora, probemos que $\hat{\cdot}_{\mathsf{n}}$ es matching de $\mathcal{S}_n = \mathcal{P}^{\mathsf{p}\neq\mathsf{n}} \oplus (\bigoplus_{\mathsf{q}\in LastOf_1(\mathsf{p})} \mathcal{P}^{\mathsf{n}<_L\mathsf{q}})$:

* **M1**: aplica para n y para $\mathsf{q} \in LastOf_1(\mathsf{p})$. Si esta condición no se cumple para algún q, ya vimos que por el caso **M1** se generaba un antiescenario $\mathcal{N}_{\mathcal{S}_1}$.
  Ahora si n no verifica **M1** tenemos $s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{n})} \notin \ell(\mathsf{n})$. Luego, como $\ell(\mathsf{n}) = \Sigma \cup \{\lambda\}$, sabemos que $s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{n})} \notin \Sigma \cup \{\lambda\}$. Ahora, como para algún $\mathsf{q} \in LastOf_1(\mathsf{p})/s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{n})} = s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{q})}$ entonces $s_{\hat{\cdot}_{\mathsf{n}}(\mathsf{q})} \notin \ell(\mathsf{q})$. Sin embargo vimos que todo q cumple **M1**, por lo tanto, llegamos a un absurdo y **M1** también se verifica para n.
* **M2**: aplica para (n, p). La condición se cumple:
$$\hat{\cdot}_{\mathsf{n}}(\mathsf{n}) = \max\{\hat{\mathsf{r}}/\mathsf{r} \in LastOf_1(\mathsf{p})\} \neq \hat{\mathsf{p}} = \hat{\cdot}_{\mathsf{n}}(\mathsf{p})$$
* **M3**: no aplica a los puntos del escenario.
* **M4-7**: no aplican a los puntos del escenario.
* **M8**: aplica para n. La condición se verifica:
$\hat{\cdot}_{\mathsf{n}}(\mathsf{n}) = \max\{\hat{\cdot}_{\mathsf{n}}(\mathsf{r})/\mathsf{r} \in LastOf(\mathsf{p})\} = \max\{\hat{\cdot}_{\mathsf{n}}(\mathsf{r})/\mathsf{r} \in LastOf_1(\mathsf{p})\} = \max\{\hat{\mathsf{r}}/\mathsf{r} \in LastOf_1(\mathsf{p})\}$

Finalmente, como $\mathcal{N}_{\mathcal{S}_1} = \mathcal{S}_l \oplus \mathcal{S}_n$, por la *Propiedad A.14.1*, sabemos que $\hat{\cdot}_{\mathsf{n}}|_{\mathcal{N}_{\mathcal{S}_1}}$ es un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\cdot}_{\mathsf{n}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

□

A.11 Antiescenario para Escenarios del Camino

Lema A.11.1 (Antiescenario para Escenarios del Camino)**.** Dados $\mathcal{S}_0$ y $\mathcal{S}_1$ dos escenarios tal que $\mathcal{S}_1 <:: \mathcal{S}_0$ por un ranking $\lessdot$ sobre P_1, y un punto $\mathsf{p} \in P_1$, $\sigma = \langle s, \tau \rangle$ una ejecución, y $\hat{\cdot}$ un matching entre $\mathcal{S}_0$ y σ para el cual no existe matching $\hat{\hat{\cdot}}$ entre el escenario del camino $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ y σ, donde $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$, entonces:

existe $\mathcal{N}_{\mathcal{S}_1}$ un antiescenario de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ generado por las reglas, con un matching $\hat{\hat{\cdot}}$ entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, donde $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

Demostración.

Sea $\mathsf{p}' = \min_{\lessdot}\{\mathsf{p}'/\mathsf{p}' \leqslant \mathsf{p}$ tal que no existe matching $\hat{\hat{\cdot}}$ entre $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'}$ y σ con $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}\}$. Es decir, el mínimo punto del camino a p que no tiene un matching que extiende a $\hat{\cdot}$.

Por hipótesis sabemos que p' existe. Además $\mathsf{p}' \in (P_{\mathcal{P}} \setminus P_0)$, porque si $\mathsf{p}' \in P_0$, entonces $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'} = \mathcal{S}_0$, y dado que $\hat{\cdot}|_{\mathcal{S}_0}$ es matching entre $\mathcal{S}_0$ y σ, se llega a un absurdo con la definición de p'.

Ahora bien, p' puede ser un punto instante, representativo o concreto:

- Si es instante, como $\mathcal{S}_1 <:: \mathcal{S}_0$, por la *Definición 3.2.4*, tenemos $\lessdot\mathsf{p}' \hookrightarrow \mathsf{p}'$ que no puede ser cierta en el caso que p' sea instante. Por lo tanto, p' no puede ser instante.
- Si es representativo, tenemos $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'} = \bigoplus_{\mathsf{q} \in ld_{\lessdot}(\mathsf{p}')}(\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \hookrightarrow \mathsf{p}'})$ y es facil ver que $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'} = (\bigoplus_{\mathsf{q} \in ld_{\lessdot}(\mathsf{p}')} \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}) \oplus (\bigoplus_{\mathsf{q} \in ld_{\lessdot}(\mathsf{p}')} \mathcal{P}^{\mathsf{q} \hookrightarrow \mathsf{p}'})$. Dado que para todo $\mathsf{q} \in ld_{\lessdot}(\mathsf{p}')$ tenemos que $\mathsf{q} \lessdot \mathsf{p}'$, entonces por la definición de p', sabemos que hay un matching $\hat{\cdot}_{\mathsf{q}}$ con $\hat{\cdot}_{\mathsf{q}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$. Luego, por la *Propiedad A.12.1*, sabemos existe un matching $\hat{\hat{\cdot}}$ para el subescenario $\bigoplus_{\mathsf{q} \in ld_{\lessdot}(\mathsf{p}')} \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$ con $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

  Para el subescenario $\bigoplus_{\mathsf{q} \in ld_{\lessdot}(\mathsf{p}')} \mathcal{P}^{\mathsf{q} \hookrightarrow \mathsf{p}'}$ es fácil ver que el mapping $\hat{\cdot}_{\mathsf{r}}$ definido como:

  - $\hat{\cdot}_{\mathsf{r}}(\mathsf{m}) = \hat{\hat{\mathsf{m}}}$ si $\mathsf{m} \neq \mathsf{p}'$
  - $\hat{\cdot}_{\mathsf{r}}(\mathsf{m}) = \min\{\hat{\hat{\mathsf{r}}}/\mathsf{r} \in FirstOf(\mathsf{p}')\}$ si $\mathsf{m} = \mathsf{p}'$ y $\mathsf{p}' \in FirstRep$,
  - $\hat{\cdot}_{\mathsf{r}}(\mathsf{m}) = \max\{\hat{\hat{\mathsf{r}}}/\mathsf{r} \in LastOf(\mathsf{p}')\}$ si $\mathsf{m} = \mathsf{p}'$ y $\mathsf{p}' \in LastRep$.

  es un matching donde $\hat{\cdot}_{\mathsf{r}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

  Por la *Propiedad A.14.1* sabemos que $\hat{\cdot}_{\mathsf{r}}$ es un matching para $(\bigoplus_{\mathsf{q} \in ld_{\lessdot}(\mathsf{p}')} \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}) \oplus (\bigoplus_{\mathsf{q} \in ld_{\lessdot}(\mathsf{p}')} \mathcal{P}^{\mathsf{q} \hookrightarrow \mathsf{p}'})$. Luego, tenemos que $\hat{\cdot}_{\mathsf{r}}$ es matching válido para $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}'}$ con $\hat{\cdot}_{\mathsf{r}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$, que es absurdo con la definición de p'. Por lo tanto, p' no puede ser representativo.

- Si es concreto, definamos $\mathcal{N}_{\mathcal{S}_1}$ como el antiescenario de $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{p}}$ generado por la regla **Puntos sin matching 4.2.3** para el punto p'. Es decir:

  $$\mathcal{N}^{\mathcal{S}_0 \not\rightsquigarrow \mathsf{p}'} = \mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}} \oplus \mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}'}, \text{ donde } \{\mathsf{q}\} = ld_{\lessdot}(\mathsf{p}')$$

  Como $\mathsf{q} \lessdot \mathsf{p}'$, según la definición de p', existe un matching $\hat{\hat{\cdot}}$ entre $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$ y σ, donde $\hat{\hat{\cdot}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$. Ahora veamos que $\hat{\hat{\cdot}}|_{\{\mathsf{q}\}}$ también es un matching para $\mathcal{P}^{\mathsf{q} \not\rightsquigarrow \mathsf{p}'}$. Vamos a suponer que $\hat{\hat{\cdot}}|_{\{\mathsf{q}\}}$ no es un matching para este escenario, entonces no se cumple al menos una condición de matching, que para este escenario pueden ser:

  M1: $s_{\hat{\hat{\mathsf{q}}}} \in \ell(\mathsf{q})$

  Como el escenario $\mathcal{P}^{\mathcal{S}_0 \rightsquigarrow \mathsf{q}}$ define al punto q, y dado que $\hat{\hat{\cdot}}$ es un matching para este escenario entonces por la definición de matching sabemos que q verifica **M1**.

  M5: $s_{\hat{\hat{\mathsf{q}}})} \cap \gamma(\mathbf{0}, \mathsf{q}) = s_{(\hat{\hat{\mathsf{q}}} \cap \gamma(\mathsf{q}, \infty)} = \emptyset$

  1 Si $\mathsf{q} < \mathsf{p}'$, el escenario define $\gamma(\mathbf{0}, \mathsf{q}) = \emptyset$ y $\gamma(\mathsf{q}, \infty) = \ell(\mathsf{p}')$. Entonces ocurre $s_{\hat{\hat{\mathsf{q}}})} \cap \ell(\mathsf{p}') \neq \emptyset$. Es decir, que en σ aparece un evento de $\ell(\mathsf{p}')$ anterior a $\hat{\hat{\mathsf{q}}}$.

  2 Si $\mathsf{p}' < \mathsf{q}$, el escenario define $\gamma(\mathsf{q}, \infty) = \emptyset$. y $\gamma(\mathbf{0}, \mathsf{q}) = \ell(\mathsf{p}')$. Entonces ocurre $s_{(\hat{\hat{\mathsf{q}}} \cap \ell(\mathsf{p}')} \neq \emptyset$. Es decir, que en σ aparece un evento de $\ell(\mathsf{p}')$ luego de $\hat{\hat{\mathsf{q}}}$.

Definamos una extensión de $\hat{\cdot}$ como el mapping $\hat{\cdot}_{\mathsf{n}}$:

- $\hat{\cdot}_{\mathsf{n}}(\mathsf{m}) = \hat{\mathsf{m}}$ si $\mathsf{m} \neq \mathsf{p}'$
- $\hat{\cdot}_{\mathsf{n}}(\mathsf{m}) = \max\{i/\ i < \hat{\mathsf{q}}\ \text{ y } s_i \in \ell(\mathsf{p}')\}$ si $\mathsf{m} = \mathsf{p}'$ y $\mathsf{p}' < \mathsf{q}$
- $\hat{\cdot}_{\mathsf{n}}(\mathsf{m}) = \min\{i/\ i > \hat{\mathsf{q}}\ \text{ y } s_i \in \ell(\mathsf{p}')\}$ si $\mathsf{m} = \mathsf{p}'$ y $\mathsf{q} < \mathsf{p}'$

Es decir, si $\mathsf{p}' < \mathsf{q}$, se define el mapping de p' al último evento de $\ell(\mathsf{p}')$ anterior a $\hat{\mathsf{q}}$, y si $\mathsf{q} < \mathsf{p}'$, se define al primer evento de $\ell(\mathsf{p}')$ posterior a $\hat{\mathsf{q}}$. Luego, por (**1**) y (**2**) podemos observar que $\hat{\cdot}_{\mathsf{n}}|_{\{\mathsf{q},\mathsf{p}'\}}$ es un matching para $\mathcal{P}^{\mathsf{q}\hookrightarrow\mathsf{p}'}$.

Como $\hat{\cdot}_{\mathsf{n}}$ es un matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$ y para $\mathcal{P}^{\mathsf{q}\hookrightarrow\mathsf{p}'}$ donde $\hat{\cdot}_{\mathsf{n}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$, aplicando la *Propiedad A.14.1*, sabemos que $\hat{\cdot}_{\mathsf{n}}$ es un matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}} \oplus \mathcal{P}^{\mathsf{q}\hookrightarrow\mathsf{p}'}$. Por lo tanto, $\hat{\cdot}_{\mathsf{n}}$ es matching válido para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}'}$ con $\hat{\cdot}_{\mathsf{n}}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$, que es absurdo con la definición de p'. En consecuencia, $\hat{\cdot}|_{\{\mathsf{q}\}}$ necesariamente es matching de $\mathcal{P}^{\mathsf{q}\not\hookrightarrow\mathsf{p}'}$.

Finalmente, por la *Propiedad A.14.1* sabemos que $\hat{\cdot}$ es un matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}} \oplus \mathcal{P}^{\mathsf{q}\not\hookrightarrow\mathsf{p}'}$. Por lo tanto tenemos que $\hat{\cdot}$ es matching válido para el antiescenario $\mathcal{N}^{\mathcal{S}_0\not\rightsquigarrow\mathsf{p}'}$ donde $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}|_{\mathcal{S}_0}$.

□

A.12 Existencia de matching para fusión de escenarios del camino

Propiedad A.12.1 (Existencia de matching para fusión de escenarios del camino)**.** Sean $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}}$ y $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$ dos escenarios del camino, σ una ejecución, $\hat{\cdot}_{\mathsf{p}}$ un matching entre $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}}$ y σ, y $\hat{\cdot}_{\mathsf{q}}$ un matching entre $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$ y σ, donde $\hat{\cdot}_{\mathsf{p}}|_{\mathcal{S}_0} = \hat{\cdot}_{\mathsf{q}}|_{\mathcal{S}_0}$, entonces:

$$\text{existe un matching } \hat{\cdot} \text{ entre } \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}} \text{ y } \sigma, \text{ donde } \hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}_{\mathsf{p}}|_{\mathcal{S}_0}.$$

Demostración.

Definamos el mapping $\hat{\cdot}$:

- $\hat{\mathsf{m}} = \hat{\cdot}_{\mathsf{p}}(\mathsf{m})$ si $\mathsf{m} \in P_{\mathsf{p}}$
- $\hat{\mathsf{m}} = \hat{\cdot}_{\mathsf{q}}(\mathsf{m})$ si $\mathsf{m} \in P_{\mathsf{q}} \setminus P_{\mathsf{p}}$

De esta forma $\hat{\cdot}|_{\mathsf{p}}$ es un matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}}$ donde $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}_{\mathsf{p}}|_{\mathcal{S}_0}$.

Ahora probemos que $\hat{\cdot}|_{\mathsf{q}}$ es un matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$. Veamos que para todo $\mathsf{m} \in P_{\mathsf{q}}$:

- Si $\mathsf{m} \in P_{\mathsf{q}} \setminus P_{\mathsf{p}}$, por la definición del mapping $\hat{\cdot}$, tenemos $\hat{\mathsf{m}} = \hat{\cdot}_{\mathsf{q}}(\mathsf{m})$.
- Si $\mathsf{m} \in P_{\mathsf{q}} \cap P_{\mathsf{p}}$ es fácil ver que $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} <: \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{m}}$ y $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}} <: \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{m}}$, y como, por la *Propiedad A.7.1* sabemos que $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{m}} <:: \mathcal{S}_0$, podemos aplicar el *Lema A.4.1* según el cual $\hat{\cdot}_{\mathsf{p}}|_{\mathsf{m}} = \hat{\cdot}_{\mathsf{q}}|_{\mathsf{m}}$. Luego, como para este caso la definición del mapping $\hat{\cdot}$ corresponde a $\hat{\mathsf{m}} = \hat{\cdot}_{\mathsf{p}}(\mathsf{m})$, y vimos que $\hat{\cdot}_{\mathsf{p}}(\mathsf{m}) = \hat{\cdot}_{\mathsf{q}}(\mathsf{m})$, entonces $\hat{\mathsf{m}} = \hat{\cdot}_{\mathsf{q}}(\mathsf{m})$.

Por lo tanto, como $\forall \mathsf{m} \in P_{\mathsf{q}}$ tenemos que $\hat{\mathsf{m}} = \hat{\cdot}_{\mathsf{q}}(\mathsf{m})$, y dado que $\hat{\cdot}_{\mathsf{q}}(\mathsf{m})$ es un matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$, podemos afirmar que $\hat{\cdot}|_{\mathsf{q}}$ es un matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$.

Finalmente, por la *Propiedad A.14.1* sabemos que $\hat{\cdot}$ es un matching para $\mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{p}} \oplus \mathcal{P}^{\mathcal{S}_0\rightsquigarrow\mathsf{q}}$ donde $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}_{\mathsf{p}}|_{\mathcal{S}_0}$.

□

A.13 Existencia de matching para fusión de Antiescenarios

Propiedad A.13.1 (Existencia de matching para fusión de Antiescenarios)**.** Sea $\mathcal{S}_0$, $\mathcal{N}_{\mathcal{S}_1}$ y $\mathcal{N}_{\mathcal{S}_2}$ tres escenarios donde $P_1 \cap P_2 = P_0$, σ una ejecución, $\hat{\cdot}_0$ un matching entre $\mathcal{S}_0$ y σ, $\hat{\cdot}_1$ un matching entre $\mathcal{N}_{\mathcal{S}_1}$ y σ, y $\hat{\cdot}_2$ un matching entre $\mathcal{N}_{\mathcal{S}_2}$ y σ, donde $\hat{\cdot}_1|_{\mathcal{S}_0} = \hat{\cdot}_2|_{\mathcal{S}_0} = \hat{\cdot}_0$, entonces:

$$\text{existe un matching } \hat{\cdot} \text{ entre } \mathcal{N}_{\mathcal{S}_1} \oplus \mathcal{N}_{\mathcal{S}_2} \text{ y } \sigma, \text{ donde } \hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}_0.$$

Demostración.

Definamos el mapping $\hat{\cdot}$:

- $\hat{\mathsf{m}} = \hat{\cdot}_1(\mathsf{m})$ si $\mathsf{m} \in P_1$
- $\hat{\mathsf{m}} = \hat{\cdot}_2(\mathsf{m})$ si $\mathsf{m} \in P_2 \setminus P_1$

De esta forma $\hat{\cdot}|_1 = \hat{\cdot}_1$, luego $\hat{\cdot}|_1$ es un matching para $\mathcal{N}_{\mathcal{S}_1}$ donde $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}_0$.

Ahora probemos que $\hat{\cdot}|_2$ es un matching para $\mathcal{N}_{\mathcal{S}_2}$. Veamos que para todo $\mathsf{m} \in P_2$:

- Si $\mathsf{m} \in P_2 \setminus P_1$, por la definición del mapping $\hat{\cdot}$, tenemos $\hat{\mathsf{m}} = \hat{\cdot}_2(\mathsf{m})$.
- Si $\mathsf{m} \notin P_2 \setminus P_1$, entonces $\mathsf{m} \in P_1$. Luego, por la definición del mapping $\hat{\cdot}$ tenemos $\hat{\mathsf{m}} = \hat{\cdot}_1(\mathsf{m})$. Ahora, como tenemos que $P_1 \cap P_2 = P_0$, entonces $\mathsf{m} \in P_0$. Por lo tanto, $\hat{\mathsf{m}} = \hat{\cdot}_1(\mathsf{m}) = \hat{\cdot}_1(\mathsf{m})|_{\mathcal{S}_0}$, Pero por hipótesis $\hat{\cdot}_1(\mathsf{m})|_{\mathcal{S}_0} = \hat{\cdot}_2(\mathsf{m})|_{\mathcal{S}_0}$, entonces $\hat{\mathsf{m}} = \hat{\cdot}_2(\mathsf{m})|_{\mathcal{S}_0} = \hat{\cdot}_2(\mathsf{m})$.

Por lo tanto, como $\forall \mathsf{m} \in P_2$ tenemos que $\hat{\mathsf{m}} = \hat{\cdot}_2(\mathsf{m})$, y dado que $\hat{\cdot}_2(\mathsf{m})$ es un matching para $\mathcal{N}_{\mathcal{S}_2}$, podemos afirmar que $\hat{\cdot}|_2$ es un matching para $\mathcal{N}_{\mathcal{S}_2}$.

Finalmente, por la *Propiedad A.14.1* sabemos que $\hat{\cdot}$ es un matching para $\mathcal{N}_{\mathcal{S}_1} \oplus \mathcal{N}_{\mathcal{S}_2}$ donde $\hat{\cdot}|_{\mathcal{S}_0} = \hat{\cdot}_0$.

□

A.14 Fusión de escenarios con igual matching

Propiedad A.14.1 (Fusión de escenarios con igual matching)**.** Sean $\mathcal{S}_1$ y $\mathcal{S}_2$ dos escenario, $\sigma = \langle s, \tau \rangle$ una ejecución, $\hat{\cdot}$ un mapping donde $\hat{\cdot}|_{P_1}$ matching entre $\mathcal{S}_1$ y σ, y $\hat{\cdot}|_{P_2}$ es matching entre $\mathcal{S}_2$ y σ, entonces:

$$\hat{\cdot}|_{P_1 \cup P_2} \text{ es matching entre } \mathcal{S}_1 \oplus \mathcal{S}_2 \text{ y } \sigma.$$

Demostración.

Veamos que para el escenario $\mathcal{S}_1 \oplus \mathcal{S}_2 = \langle \Sigma_{12}, P_{12}, \ell_{12}, \not\equiv_{12}, <_{12}, <_{F12}, <_{L12}, \gamma_{12}, \delta_{12} \rangle$ las condiciones de matching para $\hat{\cdot}$ se satisfacen. Entonces para todos los puntos $\mathsf{p}, \mathsf{q} \in P_{12}$:

- **M1** $s_{\hat{\mathsf{p}}} \in \ell(\mathsf{p})$

  si $\mathsf{p} \in P_1 \setminus P_2$ entonces $\ell_{12}(\mathsf{p}) = \ell_1(\mathsf{p})$. Como $\hat{\cdot}$ es matching de $\mathcal{S}_1$, por **M1**, $s_{\hat{\mathsf{p}}} \in \ell_1(\mathsf{p}) = \ell_{12}(\mathsf{p})$.

  si $\mathsf{p} \in P_2 \setminus P_1$ entonces $\ell_{12}(\mathsf{p}) = \ell_2(\mathsf{p})$. Como $\hat{\cdot}$ es matching de $\mathcal{S}_2$, por **M1**, $s_{\hat{\mathsf{p}}} \in \ell_2(\mathsf{p}) = \ell_{12}(\mathsf{p})$.

  si $\mathsf{p} \in P_1 \cap P_2$ entonces $\ell_{12}(\mathsf{p}) = \ell_1(\mathsf{p}) \cap \ell_2(\mathsf{p})$. Como $\hat{\cdot}$ es matching de $\mathcal{S}_1$ y $\mathcal{S}_2$, por **M1**, $s_{\hat{\mathsf{p}}} \in \ell_1(\mathsf{p})$ y $s_{\hat{\mathsf{p}}} \in \ell_2(\mathsf{p})$. Luego $s_{\hat{\mathsf{p}}} \in \ell_1(\mathsf{p}) \cap \ell_2(\mathsf{p})$ y podemos afirmar $s_{\hat{\mathsf{p}}} \in \ell_{12}(\mathsf{p})$.

- **M2** si $\mathsf{p} \not\equiv \mathsf{q}$ entonces $\hat{\mathsf{p}} \neq \hat{\mathsf{q}}$

  Como $\not\equiv_{12} = (\not\equiv_1 \cup \not\equiv_2)$, y sabemos que si $(\mathsf{p}, \mathsf{q}) \in \not\equiv_1$ entonces $\hat{\mathsf{p}} \neq \hat{\mathsf{q}}$, y que si $(\mathsf{p}, \mathsf{q}) \in \not\equiv_2$ entonces $\hat{\mathsf{p}} \neq \hat{\mathsf{q}}$, podemos afirmar que si $(\mathsf{p}, \mathsf{q}) \in \not\equiv_{12}$ entonces $\hat{\mathsf{p}} \neq \hat{\mathsf{q}}$.

- **M3** si $\mathsf{p} < \mathsf{q}$ entonces $\hat{\mathsf{p}} < \hat{\mathsf{q}}$

  Como $<_{12} = (<_1 \cup <_2)$, y sabemos que si $(\mathsf{p}, \mathsf{q}) \in <_1$ entonces $\hat{\mathsf{p}} < \hat{\mathsf{q}}$, y que si $(\mathsf{p}, \mathsf{q}) \in <_2$ entonces $\hat{\mathsf{p}} < \hat{\mathsf{q}}$, podemos afirmar que si $(\mathsf{p}, \mathsf{q}) \in <_{12}$ entonces $\hat{\mathsf{p}} < \hat{\mathsf{q}}$.

- **M4** $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{q}})} \cap \gamma(\mathsf{p},\mathsf{q}) = \emptyset$

  Como $\gamma_{12} = (\gamma_1 \cup \gamma_2)$, y sabemos que $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{q}})} \cap \gamma_1(\mathsf{p},\mathsf{q}) = \emptyset$ y $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{q}})} \cap \gamma_2(\mathsf{p},\mathsf{q}) = \emptyset$, podemos afirmar que $s_{(\hat{\mathsf{p}},\ \hat{\mathsf{q}})} \cap \gamma_{12}(\mathsf{p},\mathsf{q}) = \emptyset$.

- **M5** $s_{\hat{\mathsf{p}})} \cap \gamma(\mathbf{0},\mathsf{p}) = s_{(\hat{\mathsf{p}}} \cap \gamma(\mathsf{p},\infty) = \emptyset$

  Como $\gamma_{12} = (\gamma_1 \cup \gamma_2)$, y sabemos que $s_{\hat{\mathsf{p}})} \cap \gamma_1(\mathbf{0},\mathsf{p}) = s_{(\hat{\mathsf{p}}} \cap \gamma_1(\mathsf{p},\infty) = \emptyset$ y $s_{\hat{\mathsf{p}})} \cap \gamma_2(\mathbf{0},\mathsf{p}) = s_{(\hat{\mathsf{p}}} \cap \gamma_2(\mathsf{p},\infty) = \emptyset$, podemos afirmar que $s_{\hat{\mathsf{p}})} \cap \gamma_{12}(\mathbf{0},\mathsf{p}) = s_{(\hat{\mathsf{p}}} \cap \gamma_{12}(\mathsf{p},\infty) = \emptyset$.

- **M6** $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \vDash \delta(\mathsf{p},\mathsf{q})$

  Como $\delta_{12}(\mathsf{p},\mathsf{q}) = \delta_1(\mathsf{p},\mathsf{q})$ si $(\mathsf{p},\mathsf{q}) \in \delta_1 \setminus \delta_2$, $\delta_{12}(\mathsf{p},\mathsf{q}) = \delta_2(\mathsf{p},\mathsf{q})$ si $(\mathsf{p},\mathsf{q}) \in \delta_2 \setminus \delta_1$, o $\delta_{12}(\mathsf{p},\mathsf{q}) = \delta_1(\mathsf{p},\mathsf{q}) \cap \delta_2(\mathsf{p},\mathsf{q})$ si $(\mathsf{p},\mathsf{q}) \in \delta_1 \cap \delta_2$, y sabemos que $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \vDash \delta_1(\mathsf{p},\mathsf{q})$ y $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \vDash \delta_2(\mathsf{p},\mathsf{q})$, podemos afirmar que $\Delta(\tau_{[\hat{\mathsf{p}},\ \hat{\mathsf{q}}]}) \vDash \delta_{12}(\mathsf{p},\mathsf{q})$.

- **M7** $\Delta(\tau_{\hat{\mathsf{p}}]}) \vDash \delta(\mathbf{0},\mathsf{p})$

  Como $\delta_{12}(\mathbf{0},\mathsf{p}) = \delta_1(\mathbf{0},\mathsf{p})$ si $(\mathbf{0},\mathsf{p}) \in \delta_1 \setminus \delta_2$, $\delta_{12}(\mathbf{0},\mathsf{p}) = \delta_2(\mathbf{0},\mathsf{p})$ si $(\mathbf{0},\mathsf{p}) \in \delta_2 \setminus \delta_1$, $\delta_{12}(\mathbf{0},\mathsf{p}) = \delta_1(\mathbf{0},\mathsf{p}) \cap \delta_2(\mathbf{0},\mathsf{p})$ si $(\mathbf{0},\mathsf{p}) \in \delta_1 \cap \delta_2$, y sabemos que $\Delta(\tau_{\hat{\mathsf{p}}]}) \vDash \delta_1(\mathbf{0},\mathsf{p})$ y $\Delta(\tau_{\hat{\mathsf{p}}]}) \vDash \delta_2(\mathbf{0},\mathsf{p})$, podemos afirmar que $\Delta(\tau_{\hat{\mathsf{p}}]}) \vDash \delta_{12}(\mathbf{0},\mathsf{p})$.

- **M8** $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}(\mathsf{p})\}$ si $\mathsf{p} \in \mathit{FirstRep}$, y $\hat{\mathsf{p}} = \max\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{LastOf}(\mathsf{p})\}$ si $\mathsf{p} \in \mathit{LastRep}$.

  En el caso que $\mathsf{p} \in \mathit{FirstRep}$, si $\mathsf{p} \in \mathit{FirstRep}_1 \setminus \mathit{FirstRep}_2$, como $<_{F12} = (<_{F1} \cup <_{F2})$, entonces $\mathit{FirstOf}_{12}(\mathsf{p}) = \mathit{FirstOf}_1(\mathsf{p})\}$. Luego, dado que $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_1(\mathsf{p})\}$, podemos afirmar $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_{12}(\mathsf{p})\}$.

  Si $\mathsf{p} \in \mathit{FirstRep}_2 \setminus \mathit{FirstRep}_1$, como $<_{F12} = (<_{F1} \cup <_{F2})$, entonces $\mathit{FirstOf}_{12}(\mathsf{p}) = \mathit{FirstOf}_2(\mathsf{p})\}$. Luego, dado que $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_2(\mathsf{p})\}$, podemos afirmar $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_{12}(\mathsf{p})\}$.

  Si $\mathsf{p} \in \mathit{FirstRep}_1 \cap \mathit{FirstRep}_2$, como $<_{F12} = (<_{F1} \cup <_{F2})$, entonces $\mathit{FirstOf}_{12}(\mathsf{p}) = \mathit{FirstOf}_1(\mathsf{p}) \cup \mathit{FirstOf}_2(\mathsf{p})\}$. Luego, dado que $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_1(\mathsf{p})\}$ y $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_2(\mathsf{p})\}$, podemos afirmar $\hat{\mathsf{p}} = \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_1(\mathsf{p}) \cup \mathit{FirstOf}_2(\mathsf{p})\}$, $= \min\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{FirstOf}_{12}(\mathsf{p})\}$.

  En el caso que $\mathsf{p} \in \mathit{LastRep}$, es fácil observar que verifcamos la condición $\hat{\mathsf{p}} = \max\{\hat{\mathsf{r}}/\mathsf{r} \in \mathit{LastOf}_{12}(\mathsf{p})\}$ usando la argumentación anterior con $\mathit{Last}, <_L$ y max respectivamente a $\mathit{First}, <_F$ y min.

□

Apéndice B

Construcción del Tableau

En esta sección se define la transformación desde un escenario VTS a un autómata temporizado. Dado un escenario, se describe el tableau que reconoce todas las ejecuciones con matching para el escenario.

B.1 Autómatas temporizados

Los autómatas temporizados son un formalismo ampliamente utilizando para analizar y representar sistemas temporizados. Estos son soportados por distintas herramientas (por ejemplo [BDM+98, BLL+95]). La semántica esta basada en sistema de transición de estados etiquetados para los que se hacen ejecuciones divergentes en el tiempo. Puede encontrarse una presentación formal completa en [AD94, BDM+98].

Definición B.1.1 (Autómata Temporizado)**.** Un *autómata temporizado* es una tupla $\mathcal{A} = \langle L, X, \Sigma, E, I, \mathsf{l}_0 \rangle$, donde

- L (denotado $locs(\mathcal{A})$) es un conjunto finito de locaciones,
- X es un conjunto de relojes (variables reales no-negativas),
- Σ (denotado $label(\mathcal{A})$) es un conjunto de eventos,
- E es un conjunto finito de aristas,
- $I : L \stackrel{tot}{\rightarrow} \Psi_X$ es una función total que asocia a cada locación una restricción de reloj (ver a continuación) llamada el invariante de la locación,
- y $\mathsf{l}_0 \in L$ es la locación inicial (denotada $init(\mathcal{A})$).

Cada arista en E es una tupla $\langle \mathsf{l}, a, \psi, \alpha, \mathsf{l}' \rangle$, donde:

- $\mathsf{l} \in L$ es la locación origen,
- $\mathsf{l}' \in L$ es la locación destino,
- $a \in \Sigma \cup \{\lambda\}$ es el evento,
- $\psi \in \Psi_X$ es la guarda,
- $\alpha \subseteq X$ es el conjunto de relojes a inicializar en la arista.

El conjunto de restricciones de relojes Ψ_X para un conjunto de relojes X es definido de acuerdo a la siguiente gramática: $\Psi_X \ni \psi ::= x \prec c | \psi \wedge \psi | \neg\psi$, donde $x \in X, \prec \in \{<, \leq\}$ y $c \in \mathbf{N}$.

Generalmente, un autómata temporizado $\mathcal{A}$ tiene un mapping asociado $Pr : locs(\mathcal{A}) \mapsto 2^{Props}$ el cual asigna a cada locación un subconjunto de variables proposicionales ($Props$). La composición paralela $\mathcal{A}_1 \parallel \mathcal{A}_2$ de autómatas temporizado $\mathcal{A}_1$ y $\mathcal{A}_2$ es definida usando el producto sincronizado de los autómatas [BDM+98].

B.2 Construcción del Tableau

A continuación, al referirse a un escenario $\mathcal{S}$, se estará haciendo referencia a $\langle \Sigma, P, \ell, \not\equiv, <, <_F, <_L, \gamma, \delta \rangle$.

Definición B.2.1 (Configuración)**.** Una *configuración* Θ de un escenario $\mathcal{S}$ es un subconjunto de P tal que:

- **C1** Θ es cerrado a izquierda sobre la relación $(< \cup <_F \cup <_L)$;
- **C2** si $\mathsf{p} \in FirstRep$ y $\mathsf{p} \in \Theta$ entonces $FirstOf(\mathsf{p}) \cap \Theta \neq \emptyset$;
- **C3** si $\mathsf{q} \in LastRep$ y $LastOf(\mathsf{q}) \subseteq \Theta$ entonces $\mathsf{q} \in \Theta$.

Llamaremos como $Conf \subseteq 2^P$ al conjunto de todas las configuraciones.

Definición B.2.2 (Extensión de Configuración)**.** El par $\langle a, F \rangle \in \Sigma \cup \{\lambda\} \times 2^P$ es una *extensión* de $\Theta \in Conf$ (denotado $\Theta \xrightarrow{a} \Theta \cup F$) sii:

- **E1** $\Theta \cup F \in Conf$;
- **E2** $(\nexists \mathsf{p}, \mathsf{q} \in F)\langle \mathsf{p}, \mathsf{q} \rangle \in (\not\equiv \cup <)$;
- **E3** $(\forall \mathsf{p} \in F)(a \in \ell(\mathsf{p}))$.

Definición B.2.3 (Restricciones Activas de Eventos)**.** Se define el conjunto de *restricciones activas de eventos* para una configuración Θ, $AR(\Theta) \subseteq (\not\equiv \cup <)$ de la siguiente manera:

- $\langle \mathsf{p}, \mathsf{q} \rangle \in AR(\Theta)$ sii $\mathsf{p} \in \Theta, \mathsf{q} \notin \Theta$;
- $\langle \mathbf{0}, \mathsf{p} \rangle \in AR(\Theta)$ sii $\mathsf{p} \notin \Theta$;
- $\langle \mathsf{p}, \infty \rangle \in AR(\Theta)$ sii $\mathsf{p} \in \Theta$).

Llamamos $\Gamma(\Theta) \stackrel{def}{=} \bigcup \gamma(\mathsf{p}, \mathsf{q})$ para todo $\langle \mathsf{p}, \mathsf{q} \rangle \in AR(\Theta)$. Adicionalmente, el conjunto *estricto de restricciones activas de eventos* para una configuración Θ respecto a un conjunto de puntos F es definido como $SAR(\Theta, F) \stackrel{def}{=} \{\langle \mathsf{p}, \mathsf{q} \rangle \in AR(\Theta) / \ q \notin F\}$. Llamamos $\Gamma_F(\Theta) \stackrel{def}{=} \bigcup \gamma(\mathsf{p}, \mathsf{q})$ para todo $\langle \mathsf{p}, \mathsf{q} \rangle \in SAR(\Theta, F)$.

Dado una restricción temporal φ y un reloj x, se define $\psi_x(\varphi)$ como la restricción de reloj sobre x en la cual para todo número real no-negativo t, $\psi_x(\varphi)[x \backslash t]$ es verdadero sii $t \vDash \varphi$. Por ejemplo, $\psi_x((a, b])$ es la restricción $a < x \wedge x \leq b$.

Dado un escenario $\mathcal{S}$ y $F \subseteq P$, se define el conjunto $R_F \stackrel{def}{=} \{x_\mathsf{p} / \mathsf{p} \in F \wedge \exists \mathsf{q}.\delta(\mathsf{p}, \mathsf{q}) \neq [\mathbf{0}, \infty]\}$. Dada una configuración Θ y una extensión F, se define $\psi_\Theta^F \stackrel{def}{=} \bigwedge_{\mathsf{p} \in \Theta \uplus \{\mathbf{0}\},\ \mathsf{q} \in F} \psi_{x_\mathsf{p}}(\delta(\mathsf{p}, \mathsf{q}))$.

Definición B.2.4 (Construcción del Tableau)**.** El *tableau* $\mathcal{T}_\mathcal{S}$ para $\mathcal{S}$ un escenario VTS corresponde al autómata temporizado $\langle L, X, \Sigma, E, I, \mathsf{l}_0 \rangle$ tal que:

- **T1** $L = Conf \uplus \{\mathsf{s}_{\mathsf{trap}}\}$, llamamos $\mathsf{s}_{\mathsf{accept}}$ a la configuración P (es decir la configuración con todos los puntos en el escenario);
- **T2** $X = \{x_\mathsf{p} / \mathsf{p} \in P \uplus \{\mathbf{0}\}\}$;
- **T3** $E = \{\langle \Theta, a, \psi_\Theta^F, R_F, \Theta' \rangle / \Theta \xrightarrow{a} \Theta' \wedge a \notin \Gamma_F(\Theta)\}$
  $\cup \{\langle \Theta, a, true, \emptyset, \Theta \rangle / a \in \Sigma \wedge a \notin \Gamma(\Theta)\}$
  $\cup \{\langle \Theta, a, true, \emptyset, \mathsf{s}_{\mathsf{trap}} \rangle / a \in \Gamma(\Theta)\}$
  $\cup \{\langle \mathsf{s}_{\mathsf{trap}}, a, true, \emptyset, \mathsf{s}_{\mathsf{trap}} \rangle / a \in \Sigma\}$ donde $\Theta' = \Theta \cup F$;
- **T4** $(\forall \mathsf{l} \in L)(I(\mathsf{l}) \equiv true)$;
- **T5** l_0 es la configuración vacía.

Bibliografía

[ABKO04] Alejandra Alfonso, Victor Braberman, Nicolas Kicillof, and Alfredo Olivero. Visual timed event scenarios. In *Proc. of the 26th ACM/IEEE ICSE '04*. ACM Press, 2004.

[ACD97] George S. Avrunin, James C. Corbett, and Laura K. Dillon. Analyzing partially-implemented real-time systems. In *Proc. of the 18th ACM/IEEE Conf. ICSE '97*, pages 228–238. IEEE, 1997.

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[AEN99] Nina Amla, E. Allen Emerson, and Kedar S. Namjoshi. Efficient decompositional model checking for regular timing diagrams. In *Porc. of Intl. Conf. on Correct Hardware Design and Verification Methods*, volume 1703 of *LNCS*, pages 67–81. Springer Verlag, 1999.

[Alf03] A. Alfonso. Un lenguaje visual para la especificación y verificación automática de requerimientos de tiempo real complejos. Master's thesis, FCEyN. Univ. de Buenos Aires, 2003.

[AM04] Rajeev Alur and P. Madhusudan. Decision problems for timed automata: A survey. In Marco Bernardo and Flavio Corradini, editors, *SFM*, volume 3185 of *Lecture Notes in Computer Science*, pages 1–24. Springer, 2004.

[BDM+98] M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine. Kronos: A model-checking tool for real-time systems. In *Proc. of the 10th Intl. Conf. CAV '98*, volume 1427 of *LNCS*, pages 546–550. Springer Verlag, 1998.

[BGO02] Victor Braberman, Diego Garbervetsky, and Alfredo Olivero. Improving the verification of timed systems using influence information. In *Proc. of the 8th Intl. Conf. TACAS '02*, volume 2280 of *LNCS*, pages 21–36. Springer Verlag, 2002.

[BGO04] Victor Braberman, Diego Garbervetsky, and Alfredo Olivero. Obsslice: A timed automata slicer based on observers. In *Proc. of the 16th Intl. Conf. CAV '04*, LNCS. Springer Verlag, 2004.

[BKO05a] Víctor Braberman, Nicolás Kicillof, and Alfredo Olivero. A scenario-matching approach to the description and model checking of real-time properties. *IEEE Transactions on software Engineering*, 31(12), 2005.

[BKO05b] Victor Braberman, Nicolas Kicillof, and Alfredo Olivero. Visual Timed Event Scenarios. Technical Report 05-011, Computer Science Department – School of Science - University of Buenos Aires, 2005.

[BLL+95] Johan Bengtsson, Kim Guldstrand Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi. UPPAAL - a tool suite for automatic verification of real-time systems. In *Proc. of the Intl. Conf. on Hybrid Systems*, pages 232–243. Springer Verlag, 1995.

[DAC99] Matthew B. Dwyer, George S. Avrunin, and James C. Corbett. Patterns in property specifications for finite-state verification. In *Proc. of the 21th ACM/IEEE ICSE '99*, pages 411–420. ACM Press, 1999.

[GM03] D. Giannakopoulou and J. Magee. Fluent model checking for event-based systems. In *Proc. of the ACM/SIGSOFT Intl. Conf. ESEC/FSE 2003*, pages 257–266. ACM, September 2003.

[HM02] D. Harel and R. Marelly. Playing with time: On the specification and execution of time-enriched lscs. In *Proc. of the 10th IEEE/ACM Intl. Symp. MASCOTS '02*, pages 193–202. IEEE Computer Society, 2002.

[HR02] K. Havelund and G. Rosu. Synthesizing monitors for safety properties. In *Proc. of the 8th Intl. Conf. TACAS '02*, volume 2280 of *LNCS*, pages 342–356. Springer Verlag, 2002.

[IT00] ITU-T. Recommendation Z.120. Message Sequence Charts. Technical Report Z-120, International Telecommunication Union – Standardization Sector, Genève, 2000.

[MRK+97] L. E. Moser, Y. S. Ramakrishna, G. Kutty, P. M. Melliar-Smith, and L. K. Dillon. A graphical environment for the design of concurrent real-time systems. *ACM TOSEM*, 6(1), 1997.

[RS97] Jean-François Raskin and Pierre-Yves Schobbens. State clock logic: A decidable real-time logic. In Oded Maler, editor, *HART*, volume 1201 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 1997.

[SC02] Bikram Sengupta and Rance Cleaveland. Triggered message sequence charts. In *SIGSOFT FSE*, pages 167–176, 2002.

[SHE01] Margaret H. Smith, Gerard J. Holzmann, and Kousha Etessami. Events and constraints: A graphical editor for capturing logic requirements of programs. In *Proc. of the 5th IEEE Intl. Symp. RE '01*, pages 14–22, 2001.

[UKM02] S. Uchitel, J. Kramer, and J. Magee. Negative scenarios for implied scenario elicitation. In *Proc. of the 10th ACM/SIGSOFT Intl. Conf. FSE '02*, pages 109–118. ACM Press, 2002.