Facultad de Matemática, Astronomía y Física

Trabajo especial de la Licenciatura en Ciencias de la Computación[1]

Semánticas de Procesos para Sistemas Interactivos

Autor:
Eric Destefanis

Directores:
Dr. Matias Lee,
Dr. Pedro D'Argenio

27 de Septiembre del 2013

[1]Con el apoyo de la Beca de Fin de Carrera otorgada por la Universidad Nacional de Córdoba, y financiada por la Agencia Nacional de Promoción Científica y Tecnológica, en el marco del programa para promover la innovación productiva a través del fortalecimiento y consolidación de capital humano aplicado a la industria de las tecnologías de la información y las comunicaciones.

Resumen

Un sistema interactivo es un sistema dónde coexisten dos tipos de eventos: los que ejecuta el entorno y producen alguna reacción en el sistema; los que ejecuta el sistema y producen reacción en el entorno. Equivalencia por trazas, simulación, bisimulación, son algunas semánticas de procesos ampliamente estudiadas en la literatura. Estas semánticas, no encajan en el contexto de sistemas interactivos.

En este trabajo estudiamos semánticas de procesos para sistemas interactivos, modelados con autómatas de interfaz. Las semánticas se formalizan utilizando el concepto de noción de observabilidad. Una vez formalizado esto, estudiamos parcialmente cual semántica diferencia más que otra. Por último, se presenta definiciones relacionales para las semánticas que permiten este tipo de caracterización.

Clasificación: F.3.2 Semantics of Programming Languages, F.4.3 Formal Languages.

Palabras claves: semántica de procesos, procesos interactivos, comportamientos observables, simulación, bisimulación, trazas.

Agradecimientos

Agradezco a mi familia, que me apoyo durante toda la carrera.
A Matias Lee y Pedro D'Argenio por su dirección en este trabajo y por ayudarme en todo lo posible.
A mi novia Romina y mi amigo Pablo, pilares fundamentales durante mis estudios.
A los conocidos y amigos que hicieron de esta una etapa muy bonita ya sea en la facu, el fútbol, las juntadas o escaladas.
A los profesores por su excelencia en la enseñanza y, en general, a toda la comunidad de FaMAF por la increible ayuda que le brindan a sus alumnos.

Eric

Índice general

Capítulo 1

Introducción

En este trabajo realizamos un estudio sobre *sistemas concurrentes*. En este tipo particular de sistemas los objetos centrales de estudio son los *procesos*. Los procesos suelen ser sistemas (informáticos o no) que interactúan con otros sistemas (informáticos o no) para llevar a cabo una tarea o función. Un proceso puede representar entidades muy variadas, por ejemplo, una máquina expendedora de bebida, una caja fuerte o un jugador de ajedrez. También existen muchos ejemplos más cercanos al mundo de la computación: los servicios webs, los sistemas multiprocesadores, los sistemas distribuidos, los sistemas operativos o los protocolos de comunicación

La *teoría de procesos* se centra en el estudio de los mismos. Dos de las principales actividades en la teoría de procesos son el *modelado* y la *verificación*. El modelado es la actividad de representar procesos, mayormente a través de estructuras matemáticas o lenguajes de descripción de sistemas. Por otro lado, la verificación es la actividad de probar propiedades sobre los mismos. De esta forma si el modelado se ajusta al sistema real se garantiza el correcto funcionamiento de este último.

Para ambas tareas es necesario establecer primero un criterio que defina cuándo dos procesos se comportan de igual manera, o en otras palabras, cuándo son *equivalentes*. Dicho criterio constituye lo que se denomina *semántica* en la teoría de procesos. Qué características del comportamiento de los sistemas deben tenerse en cuenta para determinar un criterio de equidad dependerá de los siguientes factores: primero el modelo que se utiliza para representar a los procesos, luego, de la suposiciones que se realicen sobre el modelo y el ambiente dónde el sistema se encuentre.

En este trabajo estudiaremos semánticas para sistemas dónde la interacción entre procesos se realiza a través de “mensajes”. Estos mensajes se modelarán utilizando *acciones* con el enfoque denominado *concurrencia uniforme* [1], i.e. no será de nuestro interés la estructura interna de la acción. Por lo tanto, estas acciones pueden representar distintos tipos de eventos: “se presionó el botón X”, “se envía un pedido a una base de datos”, “se produjo un timeout”, etc. Para especi-

ficar el orden en que las acciones se realizan utilizamos *sistemas de transiciones*: grafos dirigidos donde las transiciones están etiquetadas con acciones. Este marco se adapta bien para distintos contextos, por ejemplo para describir y verificar protocolos de comunicación [7] o algoritmos de exclusión mutua [12].

En particular, estudiaremos los sistemas denominados *interactivos* [13]. La particularidad de estos sistemas es que poseen dos tipos de acciones visibles: *acciones de entrada* y *acciones de salida*. Las acciones de entradas son las acciones que ejecuta el ambiente y producen un estímulo sobre el sistema. Las acciones de salida son las que ejecuta el sistema por cuenta propia. Estas acciones son de diferente naturaleza, por lo cual, no es extraño que cada tipo presente características distintas. Esta tesis aborda el problema de definir cuando dos sistemas interactivos pueden considerarse equivalentes.

Organización de la tesis

En el Capítulo 2 realizamos un breve repaso sobre algunas semánticas para sistemas de transiciones donde no existe la diferenciación entre acciones de entrada y salida.

El Capítulo 3 es el núcleo de la tesis. Aquí se introducen los sistemas interactivos y las suposiciones del modelo. Mediante algunos ejemplos se fundamenta por qué las semánticas del Capítulo 2 no se ajustan a este contexto. Para definir el conjunto de posibles semánticas se aplica un enfoque basado en *nociones de observabilidad* [6, 11]. Una noción de observabilidad representa una posible semántica. Además se estudia el poder de distinción de las nociones de observabilidad.

Este estudio permite definir un orden parcial sobre las nociones de observabilidad donde el orden está definido por el poder de distinción de cada noción. Por último, se presenta una caracterización relacional para algunas de las semánticas.

El Capítulo 4 concluye la tesis. Aquí se mencionan los trabajos relacionados y conclusiones del trabajo.

Capítulo 2

Semanticas de Procesos

2.1. Sistema de transiciones etiquetadas

El término *proceso* está asociado a una entidad que coopera con otras entidades para lograr un objetivo. Estas entidades pueden ser de diferentes tipos: el usuario de un sistema, una celda de memoria, una interfaz de algún tipo, etc. Para poder trabajar con procesos, primero es necesario definir una estructura matemática con la cual poder representarlos. Para dicha causa, a continuación se detallan los Sistemas de Transiciones Etiquetadas.

Un Sistema de Transiciones Etiquetadas es una estructura que contiene un conjunto de estados, un conjunto de etiquetas, y una relación de transición que relaciona estados de a pares mediante etiquetas. Cuando utilizamos esta estructura para representar procesos, los estados suelen representar los estados propios del proceso, las etiquetas las posibles acciones que el proceso ejecuta, y las transiciones los cambios de estado que surgen cuando ocurre una acción. Se diferencia un estado en particular al cual llamaremos *estado inicial*. El estado inicial es el estado desde el que comienza la ejecución del proceso. Las acciones internas se denotan con una etiqueta especial τ, la cual no es observable. Los estados del sistema también son, en principio, no observables desde el entorno del mismo.

Definición 1. *Un Sistema de Transiciones Etiquetadas (LTS, del inglés) es una 4-upla* $S = \langle Q,\ q_0,\ Act,\ \rightarrow \rangle$ *dónde:*

- Q *es un conjunto de estados contable y no vacio.*
- $Act = L \cup \{\tau\}$*, dónde* L *es un conjunto de etiquetas no vacio.*
- $\rightarrow \subseteq Q \times Act \times Q$*, es la relación de transición.*
- $q_0 \in Q$*, y es llamado el estado inicial.*

Definición 2. *Sea* A *un conjunto. Luego,* A^* *es el conjunto de secuencias finitas sobre* A*, con* ε *representando la secuencia vacia. Si* $\sigma_1, \sigma_2 \in A^*$ *son secuencias finitas, entonces* $\sigma_1\sigma_2$ *es la concatenación de* σ_1 *y* σ_2*.*

En la tabla 2.1 se presenta la notación usada para las transiciones en sistemas de transiciónes etiquetadas.

$$
\begin{array}{lll}
q \xrightarrow{\mu} q' & \iff_{def} & (q, \mu, q') \in \rightarrow \\
q \xrightarrow{\mu_1 \dots \mu_n} q' & \iff_{def} & \exists q_0, \dots, q_n : q = q_0 \xrightarrow{\mu_1} q_1 \xrightarrow{\mu_2} \dots \xrightarrow{\mu_n} q_n = q' \\
q \xrightarrow{\mu_1 \dots \mu_n} & \iff_{def} & \exists q' : q \xrightarrow{\mu_1 \dots \mu_n} q' \\
q \xrightarrow{\mu_1 \dots \mu_n} \not\rightarrow & \iff_{def} & \nexists q' : q \xrightarrow{\mu_1 \dots \mu_n} q' \\
q \Rightarrow q' & \iff_{def} & q = q' \text{ o } q \xrightarrow{\tau \dots \tau} q' \\
q \overset{a}{\Rightarrow} q' & \iff_{def} & \exists q_1, q_2 : q \Rightarrow q_1 \xrightarrow{a} q_2 \Rightarrow q' \\
q \overset{a_1 \dots a_2}{\Rightarrow} q' & \iff_{def} & \exists q_0 \dots q_n : q = q_0 \overset{a_1}{\Rightarrow} q_1 \overset{a_2}{\Rightarrow} \dots \overset{a_n}{\Rightarrow} q_n = q' \\
q \overset{\sigma}{\Rightarrow} & \iff_{def} & \exists q' : q \overset{\sigma}{\Rightarrow} q' \\
q \overset{\sigma}{\not\Rightarrow} & \iff_{def} & \nexists q' : q \overset{\sigma}{\Rightarrow} q'
\end{array}
$$

Cuadro 2.1: Sea $P = \langle Q,\ q_0,\ Act,\ \rightarrow \rangle$ con $q, q' \in Q, \mu, \mu_i, a, a_i \in Act$, y $\sigma \in Act^*$.

2.2. Semánticas básicas

A continuación presentamos tres semánticas distintas: trazas, ready trace, y por último, bisimulación débil. Cada una de ellas, establece un criterio distinto para diferenciar procesos.

2.2.1. Semánica de Trazas

Supongamos que podemos observar sólo las acciones que un sistema va ejecutando, y que cuando lo decidamos, podemos dejar de observar. Si observamos al sistema durante una ejecución anotando en una hoja todas las acciones que éste va realizando, y luego de algún tiempo dejamos de observar, habremos realizado la observación de una traza del sistema. La semántica de trazas de un sistema está definida como todas las posibles trazas que éste pueda realizar.

Definición 3. *Sea $S = \langle Q,\ q_0,\ Act,\ \rightarrow \rangle$ un sistema de transiciones etiquetadas. $\sigma \in Act^*$ es una traza de S si $q_0 \overset{\sigma}{\Rightarrow}$. Llamaremos $T(S)$ al conjunto de trazas de S. Dos procesos S y P son equivalentes por traza, si $T(S) = T(P)$, notación $S =_T P$.*

Ejemplo 2.2.1. *Los sistemas presentados en la figura 2.1 son equivalentes por trazas, es decir, $S =_T P$. Vale la pena notar que en el proceso P luego de ejecutar la acción a desde p_0, siempre se puede ejecutar la acción b en el siguiente estado, lo cual no sucede en el caso de S.*

En cambio, los de la figura 2.2 no son equivalentes por trazas, dado que $b \in T(S)$ pero $b \notin T(P)$, i.e. $S \neq_T P$.

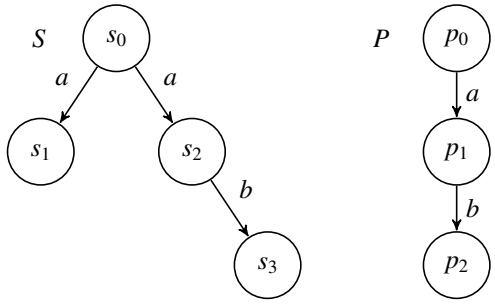


Figura 2.1: Ejemplo de procesos S y P que son equivalentes por semántica de trazas

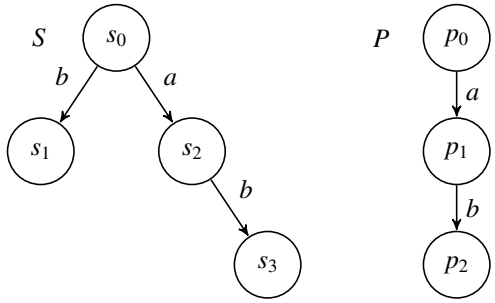


Figura 2.2: Ejemplo de procesos S y P que no son equivalentes por semántica de trazas

2.2.2. Ready Trace Semantics

A continuación se define otro tipo de semántica, cuya principal diferencia con la semántica de trazas, es que considera no solo el aspecto lineal de un sistema, sino que también trabaja con información sobre el *branching* de acciones que posee cada estado, en otra palabras, es posible distinguir que acciones puede ejecutar el sistema en cada estado.

Definición 4 (Ready Trace)**.** *Sea* $S = \langle Q, q_0, Act, \rightarrow \rangle$ *un LTS,* el conjunto de trazas ready *satisfactibles desde un estado* $q \in Q$*, notación* $rTrazas(q)$*, está definido como*

$$
\begin{aligned}
rTrazas(q) &= \{T\} \cup rTrazas_1(q) \cup rTrazas_2(q) \cup rTrazas_3(q) \\
rTrazas_1(q) &= \{a\phi \mid \exists q' : q \xrightarrow{a} q', a \in (Act - \{\tau\}), \phi \in rTrazas(q')\} \\
rTrazas_2(q) &= \{X\phi \mid q \xrightarrow{\tau}, \phi \in rTrazas(q), X = I(q)\} \\
rTrazas_3(q) &= \{\phi \mid \exists q' : q \Rightarrow q', \phi \in rTrazas(q')\}
\end{aligned}
$$

Dónde $I(q) = \{a \in Act \mid q \xrightarrow{a} \wedge\ a \neq \tau\}$. *El conjunto de trazas ready de* S, *notación rTrazas*(S), *es el conjunto de trazas de su estado inicial, i.e.* $rTrazas(S) = rTrazas(q_0)$. *Dos procesos* S *y* P *son* equivalentes por ready trace *si* $rTrazas(S) = rTrazas(P)$, *notación* $S =_{RT} P$.

Llamaremos estados estables a aquellos estados q que satisfacen $q \xrightarrow{\tau}\!\!\!\!\!/$.

El conjunto $rTrazas_1(q)$ es usado para modelar las acciones visibles que se ejecutan, el conjunto $rTrazas_2(q)$ para modelar las acciones que pueden ejecutarse en un estado estable q, y $rTrazas_3(q)$ para considerar la ejecucione de transiciones internas.

Es interesante notar que $S =_{RT} P$ implica que $S =_T P$, dado que $T(p) \subseteq rTrazas(p)$ para cualquier estado p, en particular, un estado inicial de un LTS. La contrarecíproca no es cierta, tal como muestra el ejemplo de la Figura 2.1.

Ejemplo 2.2.2. *Los dos procesos de la figura 2.3 por ejemplo, son equilvalentes por ready trace (notar que* s_1 *y* s_3 *no son estados estables), en cambio, los dos siguientes que podemos ver en la figura 2.4, no lo son, dado que la traza* $a\{b, c\}T$ *es una traza del proceso P pero no del proceso* S.

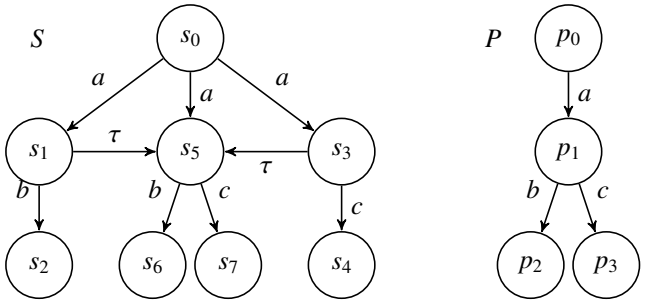


Figura 2.3: Ejemplo de procesos S y P que son equivalentes por semántica de Ready Trace

2.2.3. Bisimulación Débil (Weak Bisimulation)

Por último, presentamos la bisimulación débil. Ésta semántica se basa en la observación tanto de trazas y branching, como asi también de la estructura del sistema.

Definición 5. *(Bisimulación Débil) Sean* $S = \langle Q_S, s_0, Act_S, \rightarrow_S \rangle$ *y* $P = \langle Q_P, p_0, Act_P, \rightarrow_P \rangle$ *dos LTS. Una relación* $R \subseteq Q_S \times Q_P$ *es una* bisimulación débil (en inglés: weak bisimulation) *si para todo par de estados* $s \in Q_S, p \in Q_P$ *tal que* $s\ \mathsf{R}\ p$, *se satisfacen las siguientes propiedades de transferencia:*

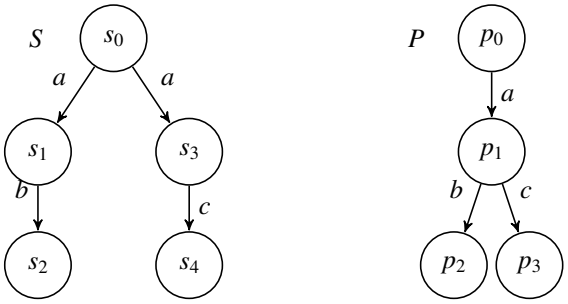


Figura 2.4: Ejemplo de procesos S y P que **no** son equivalentes por semántica de Ready Trace

1. *si* $s \xrightarrow{a} s'$ *entonces* $p \overset{a}{\Rightarrow} p'$ *y* $s' \; R \; p'$ *y*
2. *si* $p \xrightarrow{a} p'$ *entonces* $s \overset{a}{\Rightarrow} s'$ *y* $s' \; R \; p'$.

Los LTS S *y* P *son* débilmente bisimilares*, notación* $S =_{WB} P$*, si existe una bisimulación débil* R *tal que* $s_0 \; R \; p_0$.

Ejemplo 2.2.3. *Los procesos de la Figura 2.5, son débilmente bisimilares, mientras que los ejemplos mostrados en las Figuras 2.6 no lo son. Para que lo sean,* s_1 *y* s_2 *deberían estar relacionados con* p_1*. Los ejemplos de la Figura 2.4 tampoco lo son por razones similares.*

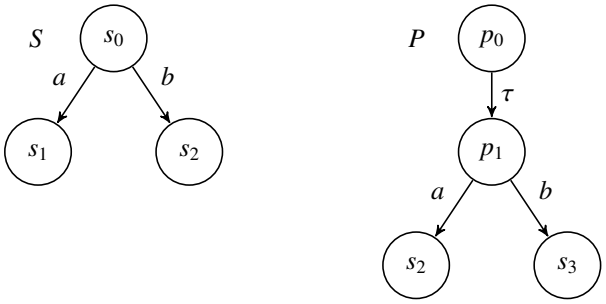


Figura 2.5: Ejemplo de procesos S y P que son equivalentes por semántica de bisimulación débil

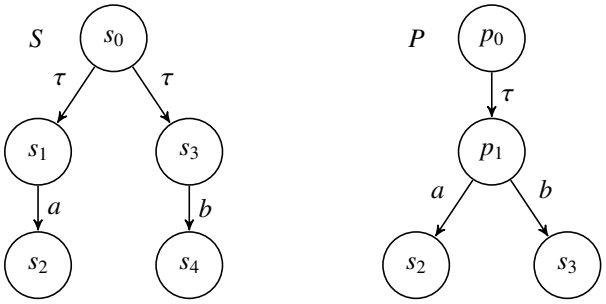


Figura 2.6: Ejemplo de procesos S y P que **no** son equivalentes por semántica de bisimulación débil

2.3. El espectro completo de semánticas para LTS

En este capítulo sólo hemos presentado 3 posibles semánticas para LTS, pero las posibles semánticas para este tipo de modelos son muchas más. En [10] se estudian semánticas para sistemas de transiciones concretos sin transiciones internas. El espectro que forman las semánticas estudiadas se presentan ordenados en la Figura 2.7. En este orden, la semantica que se encuentra arriba tiene mayor poder de distinción. Por ejemplo, si dos LTS están relacionados por una bisimulación (*bisimulation semántics*) entonces estarán relacionados por cualquier otra semántica que aparezca por debajo en este orden, exceptuando *tree semantic*. Cada semántica tiene sus propias caracteristicas, por lo cual son útiles en distintos contextos.

Por otro lado, en [4] el trabajo se centra en estudiar semánticas para procesos con acciones internas, divergencia y sub-especificados. En este caso, por estar los procesos sub-especificados, no se define el conjunto de observaciones que genera un proceso sino que cada proceso posee un conjunto de observaciones que se han realizado y un conjunto de observaciones que se podrían realizar.

El enfoque utilizado para generar los comportamientos de cada conjunto está basado en *tipos de observación* y *nociones de observabilidad*. Un tipo de observación establece una característica puntual del sistema que puede ser observada. Una noción de observabilidad es un conjunto de tipos de observaciones con cierta restricción. En base a la noción de observabilidad elegida se define el comportamiento observable de un LTS. Luego dos LTS son equivalentes, con respecto a una noción de observabilidad, si poseen el mismo comportamiento observable. Este marco de trabajo es mucho más general y por lo tanto más complejo. En la Fig. 2.8 se presenta el orden para el nuevo conjunto de semánticas. Utilizando un marco similar a este, en el próximo capítulo estudiaremos semánticas para sistemas interactivos.

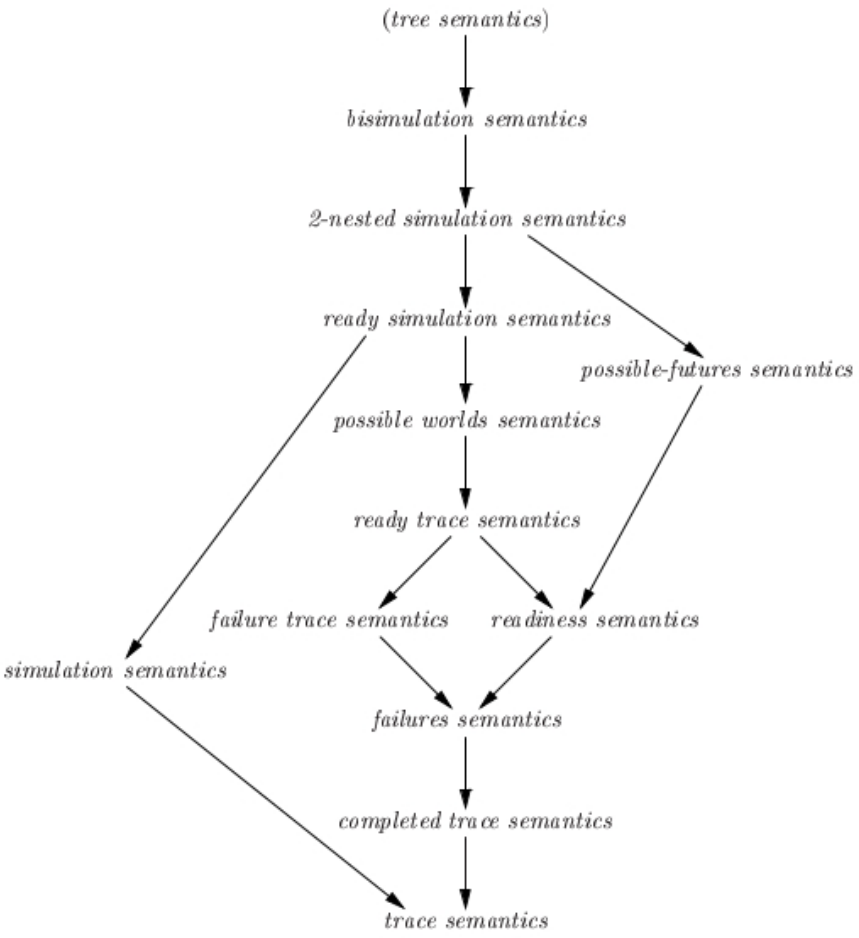


Figura 2.7: El espectro de semánticas para sistemas de transiciones sin acciones internas [10].

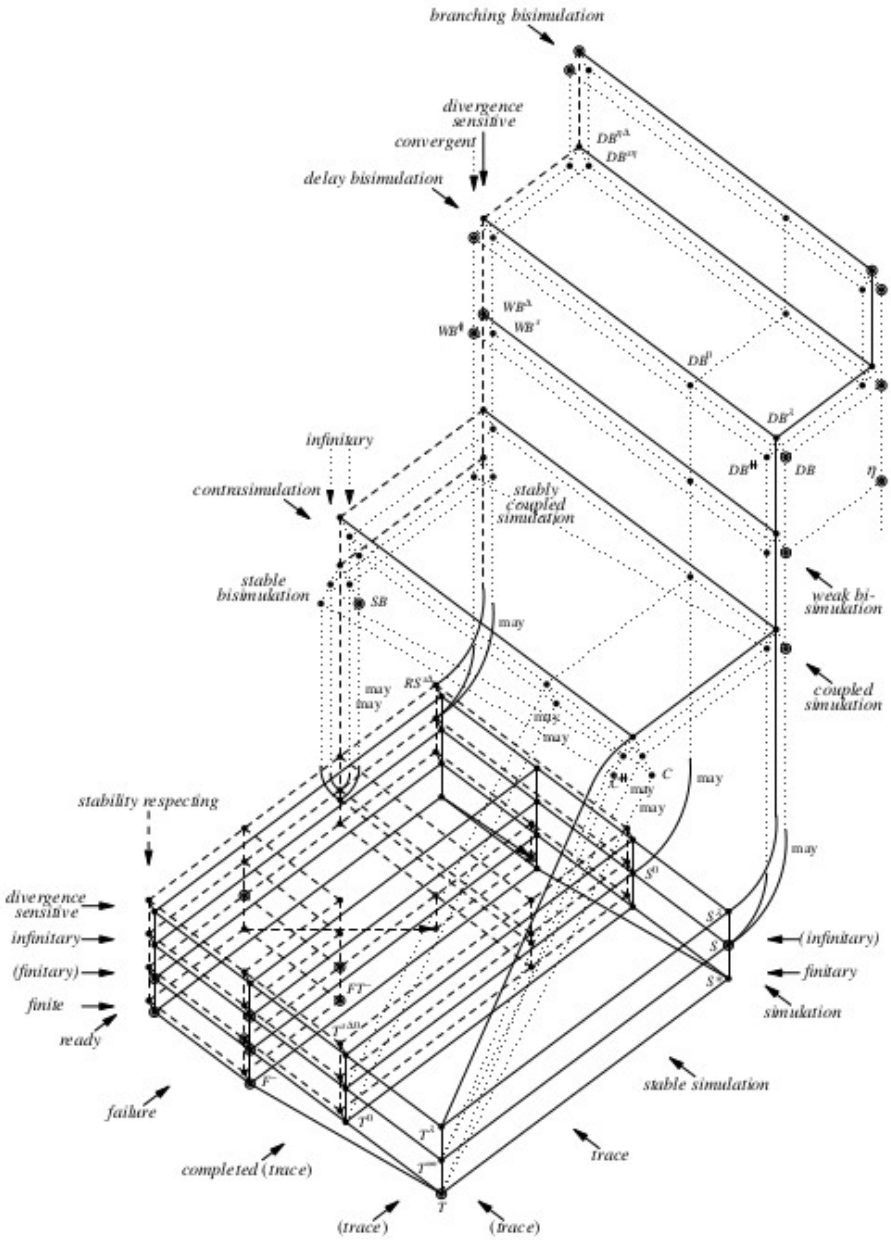


Figura 2.8: El espectro de semánticas para procesos sub-especificados con acciones internas [11].

Capítulo 3

Semánticas de Procesos Interactivos

A continuación, comenzaremos con el desarrollo de nuestro trabajo para extender el concepto de semánticas hacia sistemas interactivos. Comenzamos con algunos ejemplos que demuestran que las semánticas estudiadas en el capítulo anterior no son adecuadas para este tipo de sistemas.

3.1. Sistemas Interactivos

Un sistema interactivo es un sistema dónde coexisten dos tipos de eventos: los que se ejecutan por el entorno del sistema, para producir alguna reacción en el sistema; y los que ejecuta el sistema, los cuales pueden o no producir una reacción en el entorno. Este tipo de sistemas pueden modelarse mediante LTS dónde las acciones visibles se dividen en dos conjuntos disjuntos. El primer conjunto serán las acciones de entrada y se utilizarán para representar el primer tipo de eventos. El segundo conjunto serán las acciones de salida y se utilizarán para representar el segundo tipo de eventos. Un proceso suele ejecutarse en conjunto con otros con los que coopera.

En la Figura 3.1 vemos un ejemplo de sistema interactivo de una máquina expendedora de bebidas. En esta representación podemos ver que la máquina puede estar en distintos estados ($\{q_0, q_1, q_2, q_3\}$) y que en cada estado puede realizar distintas acciones ({*moneda?, café?, té?, servirCafé!,servirTé!*}). Notemos que el comportamiento que modela el gráfico es bastante intuitivo: si la máquina se encuentra en el estado q_0 espera que el usuario inserte una moneda, luego que elija entre té o café y en función de eso servir la bebida correspondiente.

Observemos que sólo se describen los aspectos que definen cómo es la interacción entre la máquina y su entorno, todo aspecto no relacionado a esto es dejado de lado. Por ejemplo, no se especifica características físicas de la máquina expendedora, tampoco la forma ni el valor de la moneda, etc. Entre los aspectos que hacen a la interacción podemos observar cierta notación para los distintos tipos de accio-

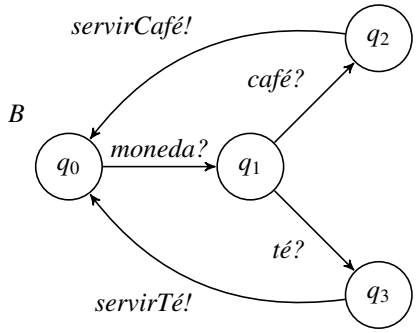


Figura 3.1: Una máquina expendedora de bebidas

nes: las acciones de entrada se denotan seguidas por un símbolo "?" y las de salida por un símbolo "!". Las denominadas *acciones de entrada* son las acciones que son ejecutadas por el *ambiente*, i.e. las entidades que interactúan con el sistema, y producen un estímulo sobre el mismo. Este estímulo está representado por el cambio de estado al realizar la transición. Las *acciones de salida* son las acciones que ejecuta el sistema y producen estímulos sobre el ambiente. En este ejemplo no se detalla, pero existe como hemos dicho un tercer tipo de acciones, *las ocultas o internas*, éstas se utilizan para representar acciones internas del sistema y se indican agregándoles el símbolo ";".

Los tres tipos de acciones a su vez se clasifican en *observables* y *no observables*. El primer grupo se compone por las acciones de entrada y de salida mientras que el segundo por las acciones ocultas. Esta división se debe a que existe el supuesto de que un observador del sistema puede observar cualquier acción de entrada o salida, y diferenciarla del resto, mientras que las acciones internas no pueden ser diferenciadas entre si, es decir, una acción de entrada (salida) *p?* puede diferenciarse de otra de entrada (salida) *q?*, mientras que una accion oculta *a;* es percibible de igual forma que una acción *b;* por parte de quién observa durante la ejecución del sistema, con lo cual no se diferencian.

Existe una última diferenciación sobre las acciones: *las acciones controlables* y las *no controlables*. Notemos que un proceso no tiene injerencia sobre las acciones de entrada. Es decir que si un proceso está en un estado en el cual sólo se transiciona mediante acciones de entrada, el sistema no progresará hasta que el ambiente realice un estímulo habilitado. Por ejemplo, la máquina expendedora nunca habilitará las opciones de seleccionar café o té si no se inserta primero una moneda. Por otro lado, el ambiente no tiene injerencia sobre las acciones de salida. En el ejemplo, luego de seleccionar café como bebida el usuario no podrá evitar que la máquina empiece a servirlo. Entonces, con respecto al usuario, las acciones *controlables* son las acciones de entrada, mientras que las *no controlables* son las acciones de salida. Las acciones internas, por ser internas al proceso serán *no controlables*.

A continuación definiremos la estructura que utilizaremos para representar sistemas interactivos, los autómatas de interfaz (vease [2] y [3]).

Definición 6. *Un autómata de interfaz (AI) es una tupla* $S = \langle Q, q_0, A^I, A^O, A^H, \rightarrow \rangle$ *dónde:*

(I) *Q es un conjunto finito de estados, tal que $q_0 \in Q$ es el estado inicial.*

(II) *A^I, A^O y A^H son conjuntos (disjuntos de a pares) finitos que representan acciones de entrada, salida y ocultas, respectivamente, con $A = A^I \cup A^O \cup A^H$.*

(III) *$\rightarrow \subseteq Q \times A \times Q$ es una relación finita llamada relación de transición, que es determinística con respecto a los inputs (i.e. $(q, a, q_1), (q, a, q_2) \in Q$, implica $q_1 = q_2$ para todo $a \in A^I$).*

En general llamaremos Q_S, A^I_S, $\rightarrow_S$, etc. a los conjuntos correspondientes para un cierto AI, S. En general, se utiliza en AI la misma notación que para sistemas de transiciones etiquetadas.

Realizamos las siguientes suposiciones con respecto a este tipo de sistemas. Éstas son sumamente importantes pues serán la base para definir las semánticas sobre el modelo.

1. Acciones generativas. El proceso bajo estudio es el que genera las acciones de salida y las internas. Es el único que las puede controlar y realizar de manera autónoma. El entorno sólo puede observarlas , pero nunca evitar su ejecución.

2. Entradas reactivas. El proceso bajo estudio no tiene control de las acciones de entrada y no puede ejecutarlas de manera atuónoma. Sólo podrá avanzar con una acción de entrada si dicha entrada es provista por el entorno. Es decir que el proceso sólo puede ejecutar una entrada como reacción al entorno que la provee.

3. Acciones instantáneas. Consideraremos que la ejecución de cada acción es instantánea.

4. Ausencia de (la paradoja de) Zenon. En un lapso acotado de tiempo sólo se puede ejecutar una cantidad finita de acciones. Esto nos permite hablar de divergencia y deadlock.

5. Reposo. Entre la ejecución de una acción y la siguiente puede transcurrir un cierto tiempo (no especificado por el modelo). Es decir, el proceso bajo estudio tiene la capacidad de reposar en cualquier estado por una cantidad de tiempo no específico.

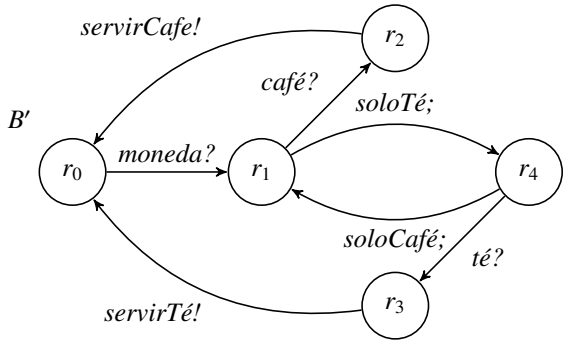


Figura 3.2: Una expendedora de bebida indecisa

6. Nos restringimos al tratamiento de concurrencia uniforme, lo cual quiere decir que la estructura interna de cada acción, no es investigada, sólo las abstraemos con etiquetas para diferenciarlas.

7. Weak fairness. Si un proceso puede ejecutar una acción de salida o interna en un estado, entonces no reposara indefinidamente y ejecutara una de ellas al cabo de un cierto tiempo.

En [4, 10] se presenta como ya hemos nombrado, un estudio extenso sobre diferentes semánticas para sistemas de transiciones etiquetadas. Sin embargo, estas semánticas no son acordes para sistemas interactivos, pues estos presentan características distintas.

En la Figura 3.2 se presenta un AI B', que representa una segunda máquina expendedora de bebidas. La misma tiene un comportamiento errático, pues luego de que un usuario inserte una moneda, por momentos ofrecerá sólo café y por momentos ofrecerá sólo té. Teniendo en cuenta las suposiciones recién realizadas se puede inferir que la interfaz descripta por B (Figura 3.1) es distinta a la interfaz descripta por B'. En B, un usuario luego de insertar una moneda podrá siempre elegir entre café o té; una vez realizada la elección, la máquina servirá la bebida elegida. Lo mismo no se puede asegurar en B': un usuario luego de insertar una moneda podrá elegir café o té, pues éstas son acciones que él controla, pero no necesariamente la interfaz responderá a ese estímulo. Esto dependerá de si el sistema se encuentra en el estado r_1 o r_4. En este caso, nos sería útil una semántica que diferencie estos dos sistemas, y que en general, diferencie dos sistemas que presentan diferencias en cuánto a las acciones de entrada que pueden ser ejecutadas desde el ambiente (muchas veces un usuario) sobre el sistema.

A continuación estudiamos como se comportan las semánticas presentadas en el capítulo anterior para estos ejemplos. Primero, notemos que es directo interpretar un sistema interactivo S como un LTS. Simplemente debemos tomar el conjunto de acciones visibles como la unión de las acciones de entrada y de salida, i.e. $A^V = A^I \cup A^O$, y luego obtener $Act = A^V \cup \{\tau\}$. Cada transición compuesta por una

acción oculta, ahora se compone con la etiqueta τ.

Las semánticas elegidas son interesantes para demostrar nuestra afirmación pues cubren distintos espectros de semánticas para LTS. La semántica de trazas es la más simple de las semánticas. Ready trace semantic podría considerarse intermedia, pues si bien refleja comportamientos lineales del sistema, tiene suficiente poder expresivo para expresar características particulares de cada estado estable, en este caso puntual, las acciones que el estado puede ejecutar. Observemos la dinámica de estas semánticas: dados dos LTS, primero se generan las trazas de cada sistema, luego éstas se comparan. Esta dinámica deja de lado la estructura de los sistemas y se centra en las trazas que cada uno puede generar. Esto no ocurre con la bisimulación débil, pues la propiedad de transferencia garantiza que los estados relacionados poseen una estructura similar, i.e. dos estados relacionados pueden imitar las transiciones que puede realizar el otro (quizás utilizando transiciones internas).

Si se compara B y B' utilizando semántica de trazas se obtiene que ambas interfaces son iguales pues vale $T(B) = T(B')$. Notemos que las acciones *soloTé;* y *soloCafé;* son acciones internas y por lo tanto no aparecen en las trazas de los procesos. Por otro lado si usamos ready trace semantic la diferencia se hace presente debido a que

$$\mathit{moneda?}\{\mathit{café?}, \mathit{té?}\}T \in \mathit{rTrazas}(q_0) \quad \text{y} \quad \mathit{moneda?}\{\mathit{café?}, \mathit{té?}\}T \notin \mathit{rTrazas}(r_0)$$

luego $\mathit{rTrazas}(q_0) \neq \mathit{rTrazas}(r_0)$. Notemos que si bien a primera vista pareciese que bisimulación débil tiene mayor poder de diferenciación, en realidad esto no es así. De hecho vale $B \approx B'$ mediante la relación

$$\{(q_0, r_0), (q_1, r_1), (q_1, r_4), (q_2, r_2), (q_3, r_3)\}$$

Esto se debe a que las acciones que se pueden realizar en el estado q_1 también pueden realizarse desde r_1 y r_4, quizás realizando previamente algunas transiciones internas.

De este ejemplo podemos concluir que las semánticas de bisimulación débil y trazas no son útiles para nuestro cometido pues igualan sistemas que como hemos dicho, nos interesa diferenciar dado que sus interfaces ante el usuario son claraente diferentes, en base a las suposiciones realizadas.

La equivalencia por ready trace logra diferenciar los modelos pues sus características permiten saber qué acciones puede ejecutar un estado estable, es decir, aquellos estados desde dónde no pueden ocurrir acciones internas. En [4, 10] se justifica esta característica dotando al usuario con un menú en el cual básicamente se pueden observar las acciones habilitadas en cualquier estado estable.

Como ya se ha mencionado, en [4, 10] no existe la diferencia entre acciones de entrada y acciones de salida, por lo tanto si extendieramos directamente la semántica ready trace en sistemas interactivos, el menú informaría que acciones de entrada y salida pueden suceder en un estado estable. Sin embargo, como veremos en el ejemplo a continuación, esta extension no es razonable.

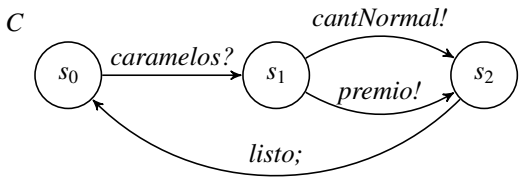


Figura 3.3: Una expendedora de caramelos generosa

Comparemos las interfaces de C y C' que se grafican en las Figuras 3.3 y 3.4, respectivamente. La primera es una máquina expendedora de caramelos generosa, pues no necesita monedas para funcionar. No sólo eso, luego del pedido de caramelos, a veces los entrega, y otras veces entrega un premio especial. Luego de ambos casos, retorna al estado inicial. El segundo caso es una máquina expendedora de caramelos generosa, pero con límites. Esto quiere decir que en algún momento, luego de cumplir con un pedido, puede no tener más premios que entregar, y a partir de ahi comportarse como una máquina que sólo devuelve la cantidad usual de caramelos.

Las máquinas han sido construidas de tal forma que no hay diferencias notorias durante su ejecución. Luego, un niño no debería poder diferenciar entre C o C' al utilizar una de estas máquinas. Una semántica que compare estos dos sistemas desde el punto de vista del usuario, debería identificarlos como equivalentes. Como ya hemos visto, la semántica de trazas y la de bisimulación débil no son útiles para comparar interfaces de usuario en este sentido.

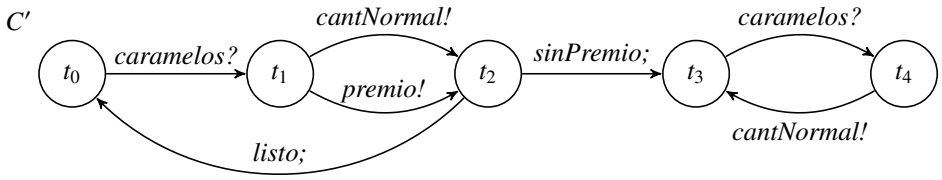


Figura 3.4: Una expendedora de caramelos generosa pero con límite

La semántica de ready trace nos permitió diferenciar las expendedoras de bebidas dado que no ofrecian las mismas acciones de entrada al usuario en momentos semejantes. Si realmente fuese una semántica útil para comparar la interacción que puede tener el usuario, debería identificar como equivalentes a C y C', es decir $rTrazas(C) = rTrazas(C')$. Esto no es así, dado que $caramelos?premio!caramelos?\{cantNormal!\}T \in rTrazas(C') - rTrazas(C)$.

Lo que buscamos mostrar con el analisis de estos sencillos ejemplos, es que cuándo trabajamos con sistemas interactivos nos interesa la diferencia entre acciones de entrada y de salida a nivel de semánticas, y esto no siempre es compatible

con extender sus definiciones de modo directo.

Las naturalezas de las acciones de entrada y salida son distintas. En el caso de la semántica de ready trace, no es común, de parte del usuario, poder ver las acciones de salida que estan habilitadas en el sistema, más aún, estas acciones son controladas por el sistema.

3.2. Nociones de Observabilidad

Nuestro estudio de semánticas esta basado en la definición de una noción de observabilidad. Una noción de observabilidad se define en base a las observaciones que podemos realizar de un sistema. Dichas observaciones se conocen como tipos o tests de observación.

3.2.1. Tipos de Observación

Los tipos de observación representan los eventos observables durante la ejecución de un proceso. Al trabajar con autómatas de interfaz las observaciones posibles para cada tipo de acción no necesariamente son las mismas.

A continuación, presentamos los tipos de observación que elegimos para nuestro modelo, algunos son extensiones directas o variaciones del trabajo realizado por van Glabbeek y otros surgen de elementos que hemos observado en trabajos que manipulan sistemas interactivos como por ejemplo [8].

T La observación del sistema es finalizada. Es posible terminarla en cualquier momento, luego de lo cual, no hay más observaciónes posibles.

a^O La ejecución de acciones de salida, del proceso son observables.

$\rightleftarrows$ El usuario interactua con la interfaz a través de inputs habilitados. Sin este test, no es posible observar acciones de entrada ejecutadas sobre el sistema.

$\rightleftarrows_{IE}$ El usuario puede interactuar con la interfaz a través de acciones de entrada ya sea que esten habilitados, o no. En ambos casos, el sistema no producirá un comportamiento visible.

F El usuario puede interactuar con la interfaz del sistema mediante acciones de entrada habilitadas o no. En caso de que una acción no esté habilitada, el sistema produce un mensaje de error y la ejecución del proceso se termina.

FT Similar al caso de F, pero ahora luego del mensaje de error, es posible continuar con la ejecución del sistema.

R Es posible observar el conjunto de inputs habilitados en el estado actual, luego de lo cual, finaliza la observación del sistema.

RT Similar al tipo de observación R, sólo que en este caso es posible continuar registrando observaciones del sistema luego de observar los inputs habilitados.

0 El observador puede detectar cuando el sistema alcanza un estado en el cual ya no puede ocurrir ningun tipo de acción (de entrada, salida o oculta), por ende, no son posibles mas observaciones (i.e. se alcanza un estado final).

δ Es posible observar inactividad por parte del sistema. Es decir, es posible observar que el sistema ya no tiene acciones de output o internas por ejecutar. Un sistema en un estado asi se denomina suspendido.

$\bigwedge$ El usuario tiene la posibilidad de hacer una cantidad arbitraria pero finita de copias del sistema.

η El usuario tiene los mecanismos para hacer una copia del sistema en el momento exacto en dónde una acción de entrada o una de salida esta por ser ejecutada sobre o por el sistema.

b El usuario tiene la maquinaria necesaria para hacer copias del sistema de forma continua durante un período de pura actividad interna. Si luego de hacerlo se observa ψ en el sistema, y en cada copia se observa ϕ, entonces decimos que se puede observar ϕ justo antes de ψ y lo representamos como $\phi\tau\psi$.

Δ El observador puede detectar que el sistema se encuentra realizando una cantidad infinita de actividad interna dónde no es posible interacción, i.e., el sistema no habilita acciones de entrada. En este caso, decimos que el sistema *diverge*.

λ El usuario puede detectar que el sistema deja de interactuar, ya sea porque diverge o porque ha llegado a un estado final. En este caso decimos que estamos en presencia de un *deadlock*.

$\neg, \neg_d$ Es posible testear el sistema bajo todas las posibles condiciones, mediante copias del mismo. Esto permite asegurar que una observación particular en un estado dado, no es posible. La diferencia entre $\neg$ y $\neg_d$ es que el primero asume que no ocurre actividad interna previo a cada copia, en cambio $\neg_d$ puede ser utilizado en cualquier momento.

ε La observación de $\varepsilon\phi$ indica que ϕ puede ser observado luego de alguna (o ninguna) actividad interna.

$\odot$ Cada ejecución de una acción interna es detectable y siempre produce la misma observación.

τ Esta cláusula dice que ningun tipo de actividad interna puede ocurrir antes de una observación que comience con $\neg, \bigvee, \bigwedge$. El conjunto de todas las observaciones que si pueden ser precedidas por actividad interna es denotado con $\mathbb{O}(\tau)$.

!! El usuario tiene la capacidad de crear copias del sistema exactamente luego de que se ejecute una acción visible. Este tipo de observación se conoce como el escenario de alerta.

$\bigvee$ Es posible observar $\bigvee_{i \in I} \phi_i$, para I conjunto de indices, si existe $i \in I$ tal que ϕ_i es una observación posible.

3.2.2. Noción de Observabilidad

A continuación, definimos el concepto en que enmarcamos las semánticas de sistemas interactivos: las nociones de observabilidad.

Definición 7. *Definimos como una noción de observabilidad, a un conjunto* $N \subseteq \{a^O, T, \rightleftarrows, \rightleftarrows_{IE}, F, R, FT, RT, 0, \delta, \bigwedge, \neg, \neg_d, \odot, \varepsilon, \tau, \bigvee, \eta, b, \lambda, \Delta, !!\}$ *que satisface:*

1. $\{a^O, T, \varepsilon, \tau, \bigvee\} \subseteq N$
2. $|\{\rightleftarrows_{IE}, F, FT, R, RT\} \cap N| \leq 1$ *y* $|\{\rightleftarrows_{IE}, F, R, FT, RT\} \cap N| = 1 \Rightarrow \rightleftarrows \in N$
3. $b \in N \Rightarrow \eta \in N \Rightarrow (\bigwedge \in N$ *y* $\odot \notin N)$
4. $\Delta \in N \Rightarrow 0 \in N$
5. $\{\odot\} \cap N \neq \emptyset \Rightarrow \{0, \delta\} \subseteq N$
6. $\{\neg, \neg_d\} \cap N \neq \emptyset \Rightarrow \rightleftarrows_{IE} \notin N$

La condición 1 establece cuales son los tipos básicos que constituyen una noción de observabilidad: las acciones de salida son siempre visibles (a^O); el usuario puede abandonar en cualquier momento la observación (T); el usuario siempre puede suponer que luego de la ejecución de una acción visible se produce actividad interna (ε); la condición global τ siempre es válida; finalmente, siempre es posible utilizar la disyunción ($\bigvee$). La segunda parte de la condición 2 ($|\{\rightleftarrows_{IE}, F, R, FT, RT\} \cap N| = 1 \Rightarrow \rightleftarrows \in N$) modela que si un usuario puede ejecutar una acción de entrada o observarlas de alguna forma, entonces esto significa que esta interactuando con la interfaz; por lo tanto la interfaz deberá responder a la ejecución de acciones de entradas habilitadas. La primera parte establece que el sistema podrá comportarse, a lo sumo, de una forma particular cuando se ejecuta una acción que no se encontraba habilitada. La condición 3 se da ya que b y η sólo tienen sentido cuando estamos en presencia de una noción que contiene el tipo de observación de copia, por definición.

Como puede verse en [11], *b* habilita a *eta*, puesto que básicamente es una generalización de la misma. La condición 4 se basa en que el concepto de divergencia va de la mano del concepto de terminación. No tiene sentido hablar de divergencia si no sabemos cuando un proceso ha terminado. Es decir, si tuvieramos el concepto de divergencia y la maquina no reaccionara bajo ningun tipo de estimulo en un tiempo suficiente, podriamos deducir que la maquina ha finalizado su trabajo, lo cual implica que tenemos una forma de detectar finalización. La condición 5 establece que si es posible ver la ocurrencia de acciones internas, entonces podemos detectar estados finales o suspendidos, dada nuestra suposición de weak fairness. Por último, la condición 6 dice que la posibilidad de testear un sistema en todas las condiciones posibles implica saber qué acciones de entrada no pueden ser ejecutadas desde un estado. Sin embargo, $\rightleftarrows_{IE}$ no permite diferenciar que acciones de entrada estan habilitadas y cuales no, con lo cual este tipo de observación pierde sentido en presencia de $\neg$ o $\neg_d$.

Para cada noción de observabilidad N, defiimos $\mathbb{O}_N$ como las *potenciales observaciones en N*, es decir las posibles observaciones que podrían realizarse en un sistema utilizando N. $\mathbb{O}_N$ esta definido inductivamente en función de los tipos en N.

Definición 8. *Sea* $S = \langle Q, q^0, A^I, A^O, A^H, \rightarrow \rangle$ *un AI y sea N una noción de observabilidad. Definimos el conjunto* $\mathbb{O}_N$*, para N, como el conjunto más chico que satisface las reglas de la Tabla 3.1 , reemplazando "*$a\varepsilon\phi$*" por "*$a\phi$*", "*$X\varepsilon\phi$ *por "*$X\phi$*", " *$\not{a}\varepsilon\phi$*" por " *$\not{a}\phi$*", y "*$\psi a\varepsilon\phi$*" por "*$\psi a\phi$*" en el caso en que* $!! \in N$.

$$\frac{}{T \in \mathbb{O}_N} \qquad \frac{}{0 \in \mathbb{O}_N} \qquad \frac{}{\Delta \in \mathbb{O}_N} \qquad \frac{}{\lambda \in \mathbb{O}_N}$$

$$\frac{\phi \in \mathbb{O}_N \quad a \in A^I \cup A^O \cup \{\odot\}}{a\varepsilon\phi \in \mathbb{O}_N} \qquad \frac{\phi \in \mathbb{O}_N \quad X \subseteq A^I \quad a \in A^I}{X\varepsilon\phi, X, \not{a}\varepsilon\phi, \not{a} \in \mathbb{O}_N} \qquad \frac{\phi \in \mathbb{O}_N}{\varepsilon\phi, \delta\phi}$$

$$\frac{\phi_i \in \mathbb{O}_N \quad i \in I}{\bigwedge_{i\in I}\phi_i, \bigvee_{i\in I}\phi_i \in \mathbb{O}_N} \qquad \frac{\phi \in \mathbb{O}_N}{\neg\phi, \neg_d\phi \in \mathbb{O}_N} \qquad \frac{\phi, \psi \in \mathbb{O}_N \quad a \in A^I \cup A^O}{\phi a\varepsilon\psi \in \mathbb{O}_N} \qquad \frac{\phi, \psi \in \mathbb{O}_N}{\phi\tau\psi \in \mathbb{O}_N}$$

Cuadro 3.1: Definición recursiva para fórmulas de ejecución

Dada una noción de observabilidad, el *comportamiento observable* de un proceso es el conjunto de elementos de $\mathbb{O}_N$ que pueden observarse en el mismo.

Definición 9. *Sea* $S = \langle Q, q_0, A^I, A^O, A^H \rangle$ *un AI, y sea N una noción de observabilidad. La función* $O_N : Q \rightarrow \mathcal{P}(\mathbb{O}_N)$ *está definida inductivamente en función de los tipos en N; para cada tipo de observación en N se deben satisfacer sus correspondientes cláusulas en la tabla 3.2. Además, la cláusula* $\eta_{\rightleftarrows}$ *se aplica cuándo* $\rightleftarrows \notin N$*,* $\eta_{\rightleftarrows_{IE}}$ *cuándo* $\rightleftarrows_{IE} \in N$ *y* $\eta_{\rightleftarrows}$ *en el resto de los casos.*

Definimos $O_N(S) = O_N(q_0)$.

Notar que $a\varepsilon\phi$ y $\phi a\varepsilon\psi$ (cuando !! $\notin N$) surgen como casos particulares de las cláusulas de la Tabla3.2. Por simplicidad, hemos decidido no definir cláusulas particulares para Δ, y λ en presencia de acciones de entrada.

Recordemos que en principio se asume que la actividad interna puede ocurrir en cualquier momento durante una observación. Para modelar esto se utiliza el tipo de observación ε y el mismo símbolo en las fórmulas de observación. Notemos que el tipo de observación !! permite al observador realizar copias justo luego de la observación de una acción, de un conjunto de acciones de entrada o una entrada rechazada. Por esta razón, en la Definición 8, si !! $\in N$, entonces el conjunto de observaciones permite formulas donde un símbolo a, X, $\not\mu$ no esté seguido por un símbolo ε.

El nuevo framework permite definir semánticas que nos permiten diferenciar de forma correcta los ejemplos estudiados previamente en este capítulo. Por ejemplo, la siguiente semántica

$$RTr = \{a^O, \bigwedge, \bigvee, \varepsilon, \tau, \rightleftarrows, RT\}\}$$

funciona correctamente con las máquinas expendedoras de bebidas (Fig. 3.1 y Fig. 3.2) y las máquinas expendedoras de caramelos (Fig. 3.3 y Fig. 3.4). En otras palabras

$$O_{RT}(B) \neq O_{RT}(B') \quad \text{y} \quad O_{RT}(C) = O_{RT}(C')$$

3.3. Preorden de Comportamientos Observables

Algo muy interesante de las semánticas tal como las hemos definido, es que mantienen entre ellas un orden particular en base a cuánto diferencian los distintos sistemas. Por ejemplo, una simple semántica de trazas divide al universo de sistemas en clases de equivalencia en dónde dos sistemas son equivalentes si sus comportamientos observables (para esta semántica) son iguales.

Esta claro que si agregamos el tipo de observación 0 a la semántica de trazas, es decir, si tenemos una semántica de trazas completas, entonces apareceran nuevas clases de equialencia, pero tendran la particularidad de que cada una de estas clases de equivalencia estara contenida en alguna de las que se obtienen con la semántica de trazas. En este sentido, podemos decir que la semántica de trazas diferencia menos (o su opuesto: identifica más) que la semántica de trazas completas.

A continuación, damos una formalización de esta idea, y brindamos algunas observaciones sobre que semánticas son mas fuertes, cuales son más débiles y cuales no son comparables.

Definición 10. *Sean S y P dos AI y N una noción de obserabilidad, S y P son $N-$ equivalentes, denotado con $S \equiv_N P$, si $O_N(S) = O_N(P)$.*

(T)	$T \in O_N(q)$	$\forall q \in Q$
(a^O)	$a\phi \in O_N(q)$	si $a \in A^O$ y $\exists q' \in Q$: $q \xrightarrow{a} q'$ y $\phi \in O_N(q')$
$(\rightleftarrows)$	$a\phi \in O_N(q)$	si $a \in A^I$ y $\exists q' \in Q$: $q \xrightarrow{a} q'$ y $\phi \in O_N(q')$
$(\rightleftarrows_{IE})$	$a\phi \in O_N(q)$	si $a \in A^I \setminus I(q)$ y $\phi \in O_N(q)$
(F)	$\not{a} \in O_N(q)$	si $a \in A^I \setminus I(q)$
(FT)	$\not{a}\phi \in O_N(q)$	si $a \in A^I \setminus I(q)$ y $\phi \in O_N(q)$
(R)	$X \in O_N(q)$	si $X = I(q)$
(RT)	$X\phi \in O_N(q)$	si $X = I(q)$ y $\phi \in O_N(q)$
(0)	$0 \in O_N(q)$	si $q \overset{a}{\not\to}$ para todo $a \in A$
$(\delta\phi)$	$\delta \in O_N(q)$	si $q \overset{a}{\not\to}$ para todo $a \in A^{OH}$
$(\bigwedge)$	$\bigwedge_{i\in I} \phi_i \in O_N(q)$	si $\phi_i \in O_N(q)$ para todo $i \in I$
$(\eta_{\not\rightleftarrows})$	$\phi a \psi \in O_N(q)$	si $a \in A^O$ y $\exists q' \in Q : q \xrightarrow{a} q'$ y se da que $\phi \in O_N(q)$ y $\psi \in O_N(q')$
$(\eta_{\rightleftarrows})$	$\phi a \psi \in O_N(q)$	si $a \in A^O \cup A^I$ y $\exists q' \in Q : q \xrightarrow{a} q'$ y se da que $\phi \in O_N(q)$ y $\psi \in O_N(q')$
$(\eta_{\rightleftarrows_{IE}})$	$\phi a \psi \in O_N(q)$	si $a \in A^I \cup A^O$ y $\exists q' \in Q : q \xrightarrow{a} q'$ y $\phi \in O_N(q)$ y $\psi \in O_N(q')$
		o se da que $a \in A^I$ y $q \overset{a}{\not\to}$ y $\phi, \psi \in O_N(q)$
(b)	$\phi \tau \psi \in O_N(q)$	si $\exists q' \in Q, a \in A^H : (q \xrightarrow{a} q' \bigvee q = q')$ y se da que $\phi \in O_N(q)$ y $\psi \in O_N(q')$
(Δ)	$\Delta \in O_N(q)$	si $\exists q_0, q_1, \ ... \in Q$ tal que $q = q_0 \xrightarrow{\tau} q_1 \xrightarrow{\tau}$ ... y $q_i \overset{a}{\not\to}$ para $a \in A^I$
(λ)	$\lambda \in O_N(q)$	si $\exists q_0, q_1, \ ... \in Q$ tal que $q = q_0 \xrightarrow{\tau} q_1 \xrightarrow{\tau} q_2$ ... con $q_i \overset{a}{\not\to}$
		para $a \in A^I$, o $q \overset{a}{\not\to}$ para todo $a \in A$
$(\neg)$	$\neg\phi \in O_N(q)$	si $\phi \notin O_N(q)$
$(\neg_d)$	$\neg_d\phi \in O_N(q)$	si $\phi \notin O_N(q)$
$(\odot)$	$\odot\phi \in O_N(q)$	si $\exists q' \in Q, a \in A^H : q \xrightarrow{a} q'$ y $\phi \in O_N(q')$
(ε)	$\varepsilon\phi \in O_N(q)$	si $\phi \in O_N(q)$
(τ)	$\phi \in O_N(q)$	si $\exists q' \in Q, a \in A^H : q \xrightarrow{a} q'$ y $\phi \in O_N(q)$ y $\phi \in \mathfrak{O}(\tau)$

Cuadro 3.2: Semántias de observaciones

Definición 11. *Sean N y M dos nociones de observabilidad. Diremos que N es menor o igual a M, denotado con $N \preceq M$, si para todo par de AI S y P, se cumple que $S \equiv_M P \Rightarrow S \equiv_N P$. Para dados V y W subconjuntos de nociónes de observabilidad, diremos que $V \preceq^c W$ si $N \preceq M$ para todo par de nociones de observabilidad N y M, tal que $V \subseteq N$ y M es el resultado de reemplazar V por W en N.*

En general, cuando tenemos nociones de observabilidad N y M tal que $N \subseteq M$, es común que se de también que $N \preceq M$. El único caso que va en contra de dicha afirmación es en el que $\rightleftarrows_{IE} \in M$ pero $\rightleftarrows_{IE} \notin N$.

Lema 1. *Sean N y M dos nociones de observabilidad tales que $\rightleftarrows_{IE}$ esta incluido en ambas o en ninguna de las dos, entonces la siguiente afirmación es correcta: si $N \subset M$ entonces $N \preceq M$.*

Demostración. Por simplicidad, asumimos $M = N \cup \{X\}$, dónde X es un tipo de observación distinto de $\rightleftarrows_{IE}$, lo cual implica el lema por transitividad. Cuándo hablamos de cláusulas, nos referimos a las presentadas en la Definición 9. Sean S y P dos autómatas de interfaz tales que $O_M(S) = O_M(P)$. Luego, basta con demostrar que también se da $O_N(S) = O_N(P)$. Asumimos por el absurdo que $O_N(S) \neq O_N(P)$, y sin perdida de generalidad, existe $\phi \in O_N(S) - O_N(P)$. Pero por definición $\phi \in O_N(S)$ se da mediante las cláusulas de N, y como $N \subseteq M$ tenemos que $\phi \in O_M(S)$. Por consiguiente $\phi \in O_M(P)$. Ahora, basta con demostrar que para X distinto de $\rightleftarrows_{IE}$, $\phi \in O_M(P) \implies \phi \in O_N(P)$. Podemos ver algunos casos:

Si $X = \bigwedge$, entonces ϕ no posee ningun $\bigwedge$, y por ende sólo las cláusulas de N son utilizadas para probar que $\phi \in O_M(P)$. Luego, utilizando las mismas cláusulas puede verse que $\phi \in O_N(P)$.

Si $X = F$, entonces ϕ no posee ninguna observación de la forma $\not{a}$? (pues $F \notin N$), y luego sólo cláusulas de N son utilizadas para proba que $\phi \in O_M(P)$. Luego, con esas cláusulas se ve que $\phi \in O_N(P)$.

De igual forma es posible demostrar la afirmación con todos los tipos de observación, a excepción de $\rightleftarrows_{IE}$, dado que si $\rightleftarrows_{IE} \in M$, entonces su cláusula puede ser necesaria para probar $\phi \in O_M(P)$, por ende, puede no existir conjunto de cláusulas que lo demuestren sin utilizar $\rightleftarrows_{IE}$, y luego la afirmación no es comprobable siguiendo la estructura de esta demostración (más aún, la afirmación es falsa para $X = \rightleftarrows_{IE}$).

□

A continuación, describimos algunos pares de la relación $\preceq^c$.

Lema 2. *Las siguientes relaciones se satisfacen:*

$$\{F\} \preceq^c \{R\} \qquad \{F\} \preceq^c \{FT\} \qquad \{FT\} \preceq^c \{RT\}$$

$$\{R\} \preceq^c \{RT\} \qquad \{\rightleftarrows_{IE}\} \preceq^c \{FT\} \qquad \{\lambda\} \preceq^c \{\Delta, 0\} \qquad \{\bigwedge\} \preceq^c \{\neg, \bigvee\}$$

$$\{\neg_d\} \preceq^c \{\varepsilon, \neg\}$$

Demostración. Para todos los casos $V \preceq^c W$ asumimos que M es la noción de observabilidad que resulta de reemplazar V por W en una dada noción N. Sea S un AI y q un estado de S.

- $\{F\} \preceq^c \{R\}$:
  $\not{a} \in O_N(q) \iff (X \in O_M(q) \text{ con } a \notin X)$
- $\{F\} \preceq^c \{FT\}$:
  $\not{a} \in O_N(q) \iff \not{a}T \in O_M(q)$
- $\{FT\} \preceq^c \{RT\}$:
  $\not{a}\phi \in O_N(q) \iff (X\phi \in O_M(q) \text{ con } a \notin X)$
- $\{R\} \preceq^c \{RT\}$:
  $X \in O_N(q) \iff XT \in O_M(q)$
- $\{\rightleftarrows_{IE}\} \preceq^c \{FT\}$:
  $a\phi \in O_N(q) \iff a\phi \bigvee \not{a}\phi \in O_M(q)$
- $\{\lambda\} \preceq^c \{\Delta, 0\}$:
  $\lambda \in O_N(q) \iff \Delta \bigvee 0 \in O_M(q)$.
- $\{\bigwedge\} \preceq^c \{\neg, \bigvee\}$:
  $\phi \bigwedge \psi \in O_N(q) \iff \neg(\neg\phi \bigvee \neg\psi) \in O_M(q)$.
- $\{\neg_d\} \preceq^c \{\varepsilon, \neg\}$:
  $\neg_d\psi \in O_N(q) \iff \varepsilon\neg\psi \in O_M(q)$.

□

Dado M un conjunto de tipos de observación, llamaremos M^{bas} a la noción de observabilidad con menor cantidad de elementos que continene a M, por ejemplo, $F^{\text{bas}} = \{a^O, \varepsilon, \tau, \bigvee, T, F, \rightleftarrows\}$. Obviamos las llaves alrededor de F al tratarse de un conjunto de cardinalidad uno. Notemos que el tipo $\rightleftarrows$ se incluye por la definición de nociones de observabilidad.

Lema 3. *Dados dos AI S y T, si $S \equiv_{\{\rightleftarrows_{IE}\}^{bas}} T$ entonces $S \equiv_{\{a^O\}^{bas}} T$.*

Demostración. Simplemente notar que $O_{\{a^O\}^{\text{bas}}}(S) = \{\psi \in O_{\{\rightleftarrows_{IE}\}^{\text{bas}}}(S) \mid \psi \in (A^O)^*\}$. □

En la Figura 3.6 presentamos parte del poset de semánticas generado por la relación $\preceq$ mediante un diagrama de Hasse. Este poset se construye en base a los Lemas 1, 2 y 3, y contraejemplos. Los contraejemplos sirven para demostrar que una noción de observabilidad no es menor o igual que otra.

Para simplificar el gráfico, utilizamos *metanodos*. Un metanodo es un conjunto de semánticas relacionadas de una forma particular. En la Figura 3.5, se grafican dos metanodos y cómo se interpretan las relaciones entre ellos.

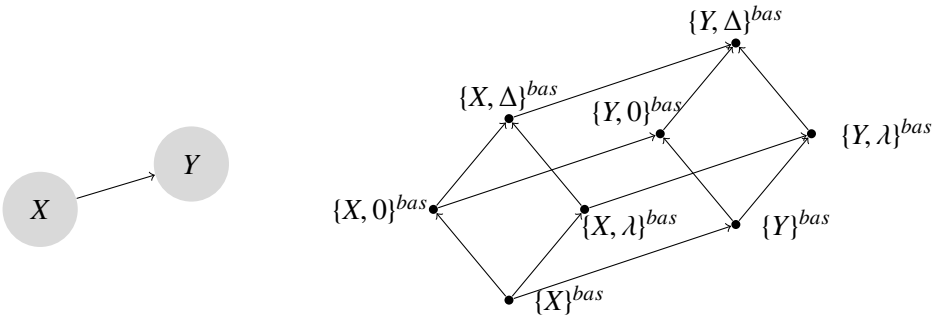


Figura 3.5: Dos metanodos relacionados y cómo se interpretan.

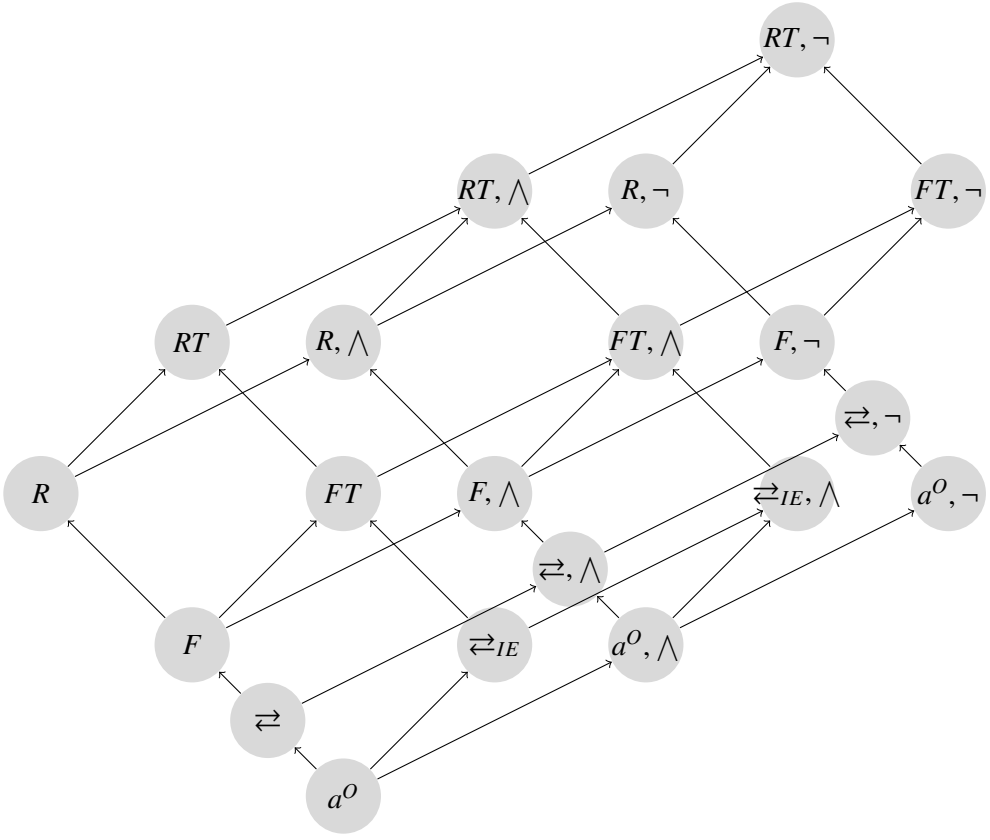


Figura 3.6: Parte del diagrama del poset de semánticas.

Equivalente por: $\{a^O\}^{\text{bas}}, \{a^O, \bigwedge\}^{\text{bas}}, \{a^O, \neg\}^{\text{bas}}$	
	
Diferenciable por	Contrajemplo
$\rightleftarrows^{\text{bas}}, \{\rightleftarrows, \bigwedge\}^{\text{bas}}, \{\rightleftarrows, \neg\}^{\text{bas}}$ $\rightleftarrows^{\text{bas}}_{IE}, \{\rightleftarrows_{IE}, \bigwedge\}^{\text{bas}}$	$x?\varepsilon T$ $x?\varepsilon a!\varepsilon T$

Cuadro 3.3

Para cada metanodo X, sólo nos enfocaremos en X^{bas}, pues extender los contraejemplos a las otras semánticas dentro del metanodo, es directo.

En la Tabla 3.3 se demuestra que la semántica $\{a^O\}^{\text{bas}}$ no diferencia más que las semánticas $\rightleftarrows^{\text{bas}}$ y $\rightleftarrows^{\text{bas}}_{IE}$, i.e.

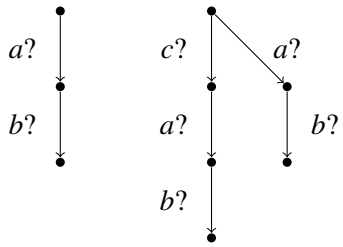
$$\rightleftarrows^{\text{bas}} \not\leq \{a^O\}^{\text{bas}} \quad \text{y} \quad \rightleftarrows^{\text{bas}}_{IE} \not\leq \{a^O\}^{\text{bas}}$$

Luego, por el Lema 2 para el caso de $\rightleftarrows^{\text{bas}}$ y el Lema 3 para el caso de $\rightleftarrows^{\text{bas}}_{IE}$, queda claro que estas dos últimas semánticas diferencian mas que $(a^O)^{\text{bas}}$. Por esta razón se encuentran arriba de ella en el diagrama. Notar que los mismos contraejemplos valen para las otras semánticas incluidas en los metanodos.

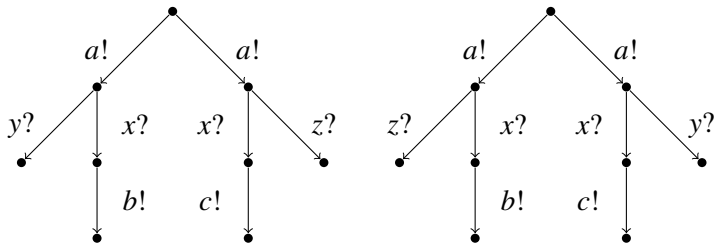
De los Lemas 1 y 2 se desprende que $\rightleftarrows^{\text{bas}} \leq F^{\text{bas}} \leq R^{\text{bas}}$. Luego para demostrar que $\rightleftarrows^{\text{bas}}_{IE}$ no es comparable con ninguna de estas semánticas basta con demostrar que $\rightleftarrows^{\text{bas}} \not\leq \rightleftarrows^{\text{bas}}_{IE}$ y $\rightleftarrows^{\text{bas}}_{IE} \not\leq R^{\text{bas}}$. Esto se demuestra en las Tablas 3.4 y 3.5 con el primer contraejemplo. Esta idea se puede generalizar de la siguiente manera: sean $N_0 \leq N_1 \leq \ ... \ N_p$ y $M_0 \leq M_1 \leq \ ... \ M_q$ dos cadenas en un poset. Para demostrar que N_i es incomparable con M_j para $0 \leq i \leq p, 0 \leq j \leq q$, basta con demostrar que $N_0 \not\leq M_q$ y que $M_0 \not\leq N_p$. Además, con el segundo contraejemplo demostramos que $FT^{\text{bas}} \not\leq R^{\text{bas}}$.

Por el Lema 2 tenemos que $R^{\text{bas}} \leq RT^{\text{bas}}$, $F^{\text{bas}} \leq FT^{\text{bas}}$, $\rightleftarrows^{\text{bas}}_{IE} \leq FT^{\text{bas}}$ y $FT^{\text{bas}} \leq RT^{\text{bas}}$.

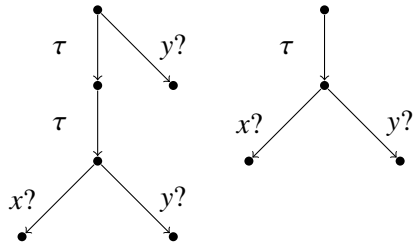
Sean N y M dos nociones de observabilidad tales que $N \cap \{F, FT\} \neq \emptyset$ y $M \cap \{R, RT\} \neq \emptyset$ entonces siempre vale que $M \not\leq N$. Esto se demuestra con el ejemplo de la Tabla 3.6 teniendo en cuenta que $\{FT, \neg\}^{\text{bas}}$ es el máximo elemento de $\{N' \mid N' \cap \{F, FT\} \neq \emptyset\}$ y R^{bas} es el menor elemento de $\{M' \mid M' \cap \{R, RT\} \neq \emptyset\}$. Esto tiene como consecuencia, teniendo en cuenta el segundo contraejemplo de la Tabla 3.5 que FT^{bas} es incomparable con R^{bas}.

Equivalente por: $\rightleftarrows_{IE}^{\text{bas}}, \{\rightleftarrows_{IE}, \bigwedge\}^{\text{bas}}$	
	
Diferenciable por	Contrajemplo
$\rightleftarrows^{\text{bas}}, \{\rightleftarrows, \bigwedge\}^{\text{bas}}$	$c?\varepsilon a?\varepsilon b?\varepsilon T$

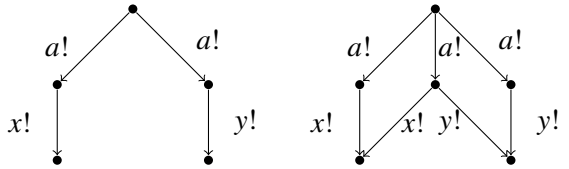
Cuadro 3.4

Equivalente por: R^{bas}	
	
Diferenciable por	Contrajemplo
$\rightleftarrows_{IE}^{\text{bas}}$	$a!\varepsilon y?\varepsilon x?\varepsilon c!\varepsilon T$
FT^{bas}	$a!\varepsilon \not\rightleftarrows \varepsilon x?\varepsilon c!\varepsilon T$
RT^{bas}	$a!\varepsilon\{x?, y?\}\varepsilon x?\varepsilon c!\varepsilon T$

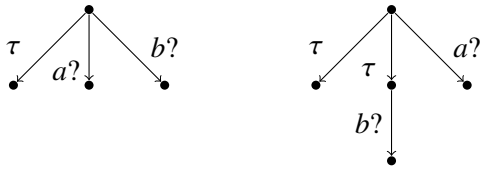
Cuadro 3.5

Equivalente por: $\{FT, \neg\}^{\text{bas}}$	
	
Diferenciable por	Contrajemplo
R^{bas}	$\{y?\}$

Cuadro 3.6

Equivalente por: X^{bas}, para $X \in \{a^O, \rightleftarrows, \rightleftarrows_{IE}, F, R, FT, RT\}$	
	
Diferenciable por	Contrajemplo
$\{X, \bigwedge\}^{\text{bas}}$, para $X \in \{a^O, \rightleftarrows, \rightleftarrows_{IE}, F, R, FT, RT\}$	$a!\varepsilon(b!\varepsilon T \bigwedge c!\varepsilon T)$

Cuadro 3.7

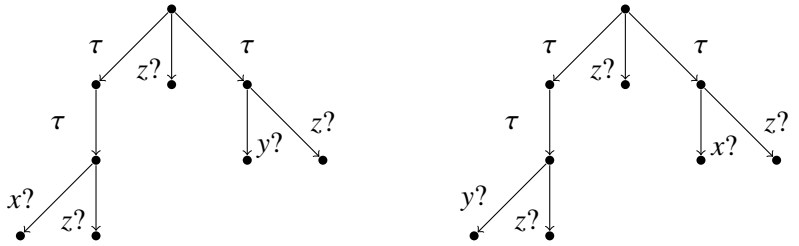
Equivalente por: $\{F, \bigwedge\}^{\text{bas}}$	
	
Diferenciable por	Contrajemplo
FT^{bas}	$\not\emptyset?\varepsilon a?\varepsilon T$

Cuadro 3.8

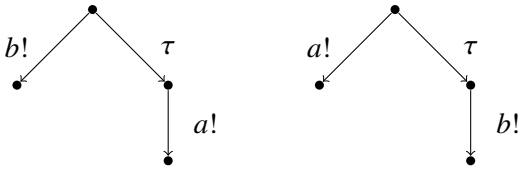
Hasta ahora hemos demostrado la construcción para los metanodos asociados a los tipos de observación $a^O, \rightleftarrows, \rightleftarrows_{IE}, F, FT, R$ y RT. Procedamos ahora con los metanodos de la forma $(X, \bigwedge)$, tal que $X \in \{F, FT, R, RT\}$. Para comenzar, demostremos que $\{R, \bigwedge\}^{\text{bas}}$ es incomparable con $\{FT, \bigwedge\}^{bas}$, lo cual nos garantiza el rombo que conforman estos metanodos: por la Tabla 3.6, $\{R, \bigwedge\}^{bas} \not\leq \{FT, \bigwedge\}^{bas}$; por otro lado, el contraejemplo de la Tabla 3.9 demuestra $\{FT, \bigwedge\}^{bas} \not\leq \{R, \bigwedge\}^{bas}$.

Se puede observar que $\{F, \bigwedge\}^{\text{bas}}$ es incomparable con R^{bas} y FT^{bas} por lo siguiente: $R^{\text{bas}} \not\leq \{F, \bigwedge\}^{bas}$ por lo expresado en referencia al contraejemplo de la Tabla 3.6; $\{F, \bigwedge\}^{\text{bas}} \not\leq FT^{\text{bas}}$ y $\{F, \bigwedge\}^{\text{bas}} \not\leq R^{\text{bas}}$ por el contraejemplo mostrado en la Tabla 3.7 (este ejemplo es posible por la diferenciación entre entradas y salidas); por último, el ejemplo de la Tabla 3.8 nos muestra que $FT^{\text{bas}} \not\leq \{F, \bigwedge\}^{\text{bas}}$. Finalmente observamos que RT^{bas} es incomparable con $\{R, \bigwedge\}^{\text{bas}}$ y $\{FT, \bigwedge\}^{\text{bas}}$ dado que: $\{R, \bigwedge\}^{\text{bas}} \not\leq RT^{\text{bas}}$ y $\{FT, \bigwedge\}^{bas} \not\leq RT^{\text{bas}}$ por el contraejemplo de la Tabla 3.7; $RT^{\text{bas}} \not\leq \{R, \bigwedge\}^{\text{bas}}$ por el tercer contraejemplo de la Tabla 3.5; $RT^{\text{bas}} \not\leq \{FT, \bigwedge\}^{\text{bas}}$ por lo ya comentado sobre el ejemplo de la Tabla 3.6.

Es fácil ver que los nodos $(a^O, \bigwedge)$, $(\rightleftarrows, \bigwedge)$ y $(\rightleftarrows_{IE}, \bigwedge)$ se encuentran bien posicionados. Esto se demuestra de la misma forma que en el caso de los nodos dónde el tipo de observación $\bigwedge$ no aparece, utilizando los Lemas 1, 3 y 2, junto con los contraejemplos de las Tablas 3.7, 3.3, 3.4 y 3.9.

Equivalente por: $\{R, \bigwedge\}^{\text{bas}}$, $\{F, \neg\}^{\text{bas}}$, $\{R, \neg\}^{\text{bas}}$	
	
Diferenciable por	Contrajemplo
$\{RT, \bigwedge\}^{\text{bas}}$ $\{FT, \bigwedge\}^{\text{bas}}, \{FT, \neg\}^{\text{bas}}$	$\{\}\varepsilon x?\varepsilon T$ $\not\subset?\varepsilon x?\varepsilon T$

Cuadro 3.9

Equivalente por: $\{R, \bigwedge\}^{\text{bas}}$, $\{FT, \bigwedge\}^{\text{bas}}$, $\{RT, \bigwedge\}^{\text{bas}}$	
	
Diferenciable por	Contrajemplo
$\{F, \neg\}^{\text{bas}}, \{R, \neg\}^{\text{bas}}, \{FT, \neg\}^{\text{bas}}$	$\varepsilon\neg(a!\varepsilon T)$

Cuadro 3.10

El rombo formado por los metanodos de la forma $(X, \neg)$ con $X \in \{F, FT, R, RT\}$ y su posición dentro del poset se demuestran de forma similar al caso anterior utilizando los contraejemplos de las Tablas 3.6, 3.9, 3.10.

Por último, las posiciones de $(a^O, \neg)$ y $(\rightleftarrows, \neg)$ se desprenden de los Lemas 1 y 2, y del contraejemplo de las Tabla 3.3.

En la Figura 3.7 observamos otra parte del poset de semánticas, en el cual empleamos nuevamente los metanodos ya descriptos. Para simplificar, utilizamos la siguiente notación: $C = \{\neg_d, \rightleftarrows\}^{bas}$ (en inglés llamada: *contrasimulation*), $CS = \{\neg_d, \rightleftarrows, \bigwedge\}^{bas}$ (*coupled simulation*), $WB = \{\neg, \rightleftarrows\}^{bas}$ (*weak bisimulation*), $DB = \{\neg, \rightleftarrows, !!\}^{bas}$ (*delay bisimulation*), $DB^{\eta} = \{\neg, \rightleftarrows, !!, \eta\}^{bas}$ (*delay bisimulation with* η) y $BB = \{\neg, \rightleftarrows, !!, b, \eta\}^{bas}$ (*branching bisimulation*).

Se puede ver que $C \preceq CS$ por el Lema 1 y $CS \not\preceq C$ dado el contraejemplo de la Tabla 3.7. $WB \not\preceq CS$ dado el contraejemplo de la Tabla 3.11, pero $CS \preceq WB$ por el Lema 2.

Es fácil ver también que $WB \preceq DB \preceq DBE \preceq BB$ utilizando el Lema 3. $DB \not\preceq WB$ dado el contraejemplo de la Tabla 3.12 y $DBE \not\preceq DB$ tal cómo muestra el contraejemplo de la Tabla 3.13.

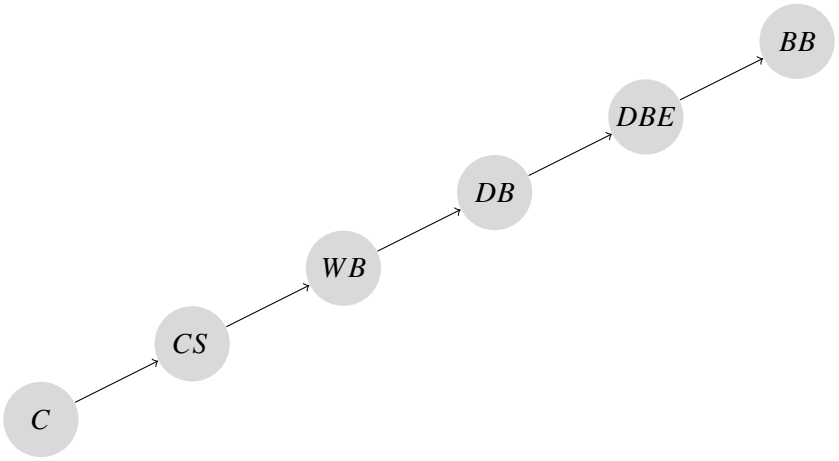
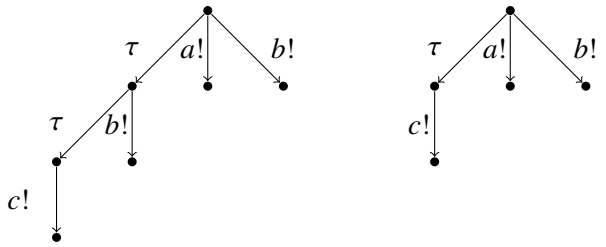
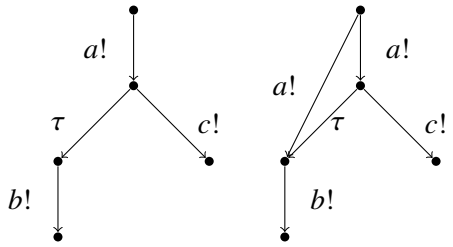


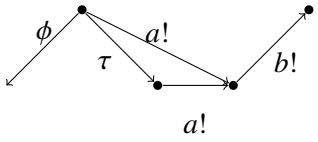
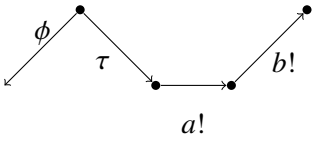
Figura 3.7: Parte del diagrama del poset de semánticas. Semánticas con $\neg$ o $\neg_d$.

Equivalente por: CS	
	
Diferenciable por	Contrajemplo
WB	$\varepsilon(\neg(a!\varepsilon T)\bigwedge(b!\varepsilon T)\bigwedge(c!\varepsilon T))$

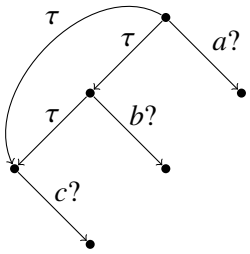
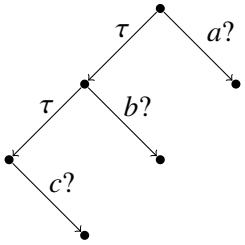
Cuadro 3.11

Equivalente por: WB	
	
Diferenciable por	Contrajemplo
DB	$a!\neg(c!T)$

Cuadro 3.12

Equivalente por: *DB*	
	
Diferenciable por	Contrajemplo
DBE	$(\phi)a!(T)$

Cuadro 3.13

Equivalente por: *DBE*	
	
Diferenciable por	Contrajemplo
BB	$a?\tau\neg(b?T)$

Cuadro 3.14

Por último, $BB \not\leq DBE$ dado el contraejemplo de la Tabla 3.14.

En principio, nos detenemos en este punto dada la complejidad de las demostraciones que conlleva agregar nuevas semánticas a su lugar correspondiente en el poset. Creemos que la estructura de ésta parte del poset (Fig 3.7) se mantiene igual cuando se agrega un elemento de $\{F, R, FT, RT\}$ a todas las semánticas. También, suponemos que dichas cuatro estructuras conforman una estrctura de rombo, de forma similar a como sucedia en la primer parte del poset que hemos presentado.

3.4. Semánticas bajo un enfoque relacional

El conjunto de observaciones que genera una interfaz bajo cualquier noción de observabilidad es usualmente infinito. Luego el conjunto de fórmulas de observación que un proceso genera no siempre ofrece un marco adecuado para comparar distintos procesos. Por esta razón, introducimos una caracterización relacional para algunas nociones de observabilidad. En otras palabras, dadas dos interfaces S y P y una noción de observabilidad N, se satisface $O_N(S) \subseteq O_N(P)$ si y sólo si existe una cierta relación R_N tal que $S \ \mathsf{R}_N \ P$. Esta relación se llamara *N-simulación*.

Definición 12. *Sean* $S = \langle Q_1, s_0, A^I, A^O, A^H \rangle$ *y* $P = \langle Q_2, p_0, A^I, A^O, A^H \rangle$ *dos AI y sea N una noción de observabilidad que incluya a* $\bigwedge$. *Sea* $Q = Q_1 \cup Q_2$. *Una* N-simulación *es una relación binaria* $\mathsf{R} \subseteq Q \times Q$ *que satisfaga las cláusulas de la Tabla 3.15 para cada tipo de observación en N, tal que* $s_0 \ \mathsf{R} \ p_0$. *Las cláusulas* t_x *y* $t_{x,y}$ *indican los casos en que* $\{t, x\} \subseteq N$ *o* $\{t, x, y\} \subseteq N$, *respectivamente.*

Notar que por como hemos definido nuestro modelo, ha sido necesario diferenciar cláusulas de acciones de salida, interacción y de η, en la Tabla 3.15. Las primeras dos, por la existencia de escenario de alerta o de canales encubiertos. La última varía lógicamente ante la presencia de acciones de entrada, asi como ante la presencia de escenario de alerta. La variación es mínima, pero necesaria para mantener coherencia en el modelo.

Teorema 1. *Dados dos IA* S *y* S', *y una noción de observabilidad N tal que* $\bigwedge \in N$, *entonces*

$$O_N(S) \subseteq O_N(S') \iff \textit{existe una N-simulación tal que } S \ \mathsf{R}_N \ S'. \tag{3.1}$$

Demostración. Primero demostremos que $O_N(S) \subseteq O_N(S')$ implica que se satisfacen las cláusulas para una relación R_N. Dados $p \in Q_S$, $q \in Q_{S'}$, definamos R_N como $p \ \mathsf{R}_N \ q \iff O_N(p) \subseteq O_N(q)$.

$a^O \in N$ Si $p \xrightarrow{a} p'$ y $a \in A^O$ entonces se debe demostrar que existe q' tal que $q \Rightarrow\!\!\xrightarrow{a}\!\!\Rightarrow q'$ y $O_N(p') \subseteq O_N(q')$. Supongamos que este no es el caso. Sea $S = \{q' \mid q \Rightarrow\!\!\xrightarrow{a}\!\!\Rightarrow q'\}$. Entonces para cada $q' \in S$ existe una formula $\phi_{q'} \in O_N(p') - O_N(q')$. Luego la fórmula $\bigwedge_{q' \in S} \phi_{q'} \in O_N(p') - O_N(q')$ no puede ser generada por ningún estado en S. Por lo tanto $a\varepsilon \bigwedge_{q' \in S} \phi_{q'} \in O_N(p) - O_N(q)$ lo que contradice la hipótesis $O_N(p) \subseteq O_N(q)$.

Las variantes $a^O_{!!}, a^O_{\odot}$ se demuestran de forma análoga, así como los tipos $\rightleftarrows$, $\rightleftarrows_{IE}$, ε, δ, 0, $\odot$, T y sus respectivas variantes.

$RT \in N$ Supongamos $I(p) = X$, entonces debemos demostrar que existe q' tal que $q \Rightarrow q'$, $I(q') = X$ y $O_N(p) \subseteq O_N(q')$. Supongamos por el absurdo que este no es el caso. Luego para todo q' tal que $q \Rightarrow q'$, $I(q') \neq X$ o existe $\phi_{q'} \in O_N(p) - O_N(q')$. Sea $S = \{q' \mid q \Rightarrow q', O_N(p) - O_N(q') \neq \emptyset, I(q') = X\}$, es fácil ver que este conjunto no es vacio dado que $O_N(p) \subseteq O_N(q)$. Luego, utilizando los $\phi_{q'}$ correspondientes a los elementos de S, tenemos que $\bigwedge_{q' \in S} \phi_{q'} \in O_N(p)$ y por RT, $X \bigwedge_{q' \in S} \phi_{q'} \in O_N(p)$. Por suposición, $X \bigwedge_{q' \in S} \phi_{q'} \in O_N(q)$, luego por (τ) existe q'' tal que $I(q'') = X$ y $\bigwedge_{q' \in S} \phi_{q'} \in O_N(q'')$, pero si $I(q'') = X$ necesariamente $q'' \in S$ y por consiguiente $\phi_{q''} \notin O_N(q'')$ lo cual contradice $\bigwedge_{q' \in S} \phi_{q'} \in O_N(q'')$.

(0)	pRq y $p \overset{a}{\not\to}, \forall a \in A \implies q \overset{a}{\not\to}$
(a^O)	pRq y $p \overset{a}{\to} p'$ con $a \in A^O \implies \exists q' : q \Rightarrow \overset{a}{\to} \Rightarrow q'$ y $p'Rq'$
$(a^O_{!!})$	pRq y $p \overset{a}{\to} p'$ y $a \in A^O \implies \exists q' : q \Rightarrow \overset{a}{\to} q'$ y $p'Rq'$
$(a^O_{\odot})$	pRq y $p \overset{a}{\to} p'$ con $a \in A^O \implies \exists q' : q \overset{a}{\to} q'$ y $p'Rq'$
(τ)	pRq y $p \overset{a}{\to} p'$ con $a \in A^H \implies \exists q' : q \Rightarrow q'$ y $p'Rq'$
$(\odot)$	pRq y $p \overset{a}{\to} p'$ con $a \in A^H \implies \exists q', a' \in A^H : q \overset{a'}{\to} q'$ y $p'Rq'$
$(\rightleftarrows)$	pRq y $p \overset{a}{\to} p'$ con $a \in A^I \implies \exists q' : q \Rightarrow \overset{a}{\to} \Rightarrow q'$ y $p'Rq'$
$(\rightleftarrows_{!!})$	pRq y $p \overset{a}{\to} p'$ con $a \in A^I \implies \exists q' : q \Rightarrow \overset{a}{\to} q'$ y $p'Rq'$
$(\rightleftarrows_{\odot})$	pRq y $p \overset{a}{\to} p'$ con $a \in A^I \implies \exists q' : q \overset{a}{\to} q'$ y $p'Rq'$
$(\rightleftarrows_{IE})$	pRq y $(p \overset{a}{\not\to} \wedge\ p = p') \vee p \overset{a}{\to} p'$ con $a \in A^I \implies \exists q' : (q \Rightarrow \overset{a}{\not\to} \Rightarrow q' \bigvee q \Rightarrow \overset{a}{\to} \Rightarrow q') : p'Rq'$
$(\rightleftarrows_{IE_{!!}})$	pRq y $(p \overset{a}{\not\to} \wedge\ p = p') \vee p \overset{a}{\to} p'$ con $a \in A^I \implies \exists q' : (q \Rightarrow q' \overset{a}{\not\to} \bigvee q \Rightarrow \overset{a}{\to} q') : p'Rq'$
$(\rightleftarrows_{IE_{\odot}})$	pRq y $(p \overset{a}{\not\to} \wedge\ p = p') \vee p \overset{a}{\to} p'$ con $a \in A^I \implies (\exists q' : q \overset{a}{\to} q' : p'Rq')$ ó $(q \overset{a}{\not\to}$ y $p'Rq)$
(F)	pRq y $p \overset{a}{\not\to}$ con $a \in A^I \implies \exists q' : q \Rightarrow q' \overset{a}{\not\to}$
(R)	pRq y $p \overset{X}{\to}$ con $X \subseteq A^I \implies \exists q' : q \Rightarrow q' \overset{X}{\to}$
(FT)	pRq y $p \overset{a}{\not\to}$ con $a \in A^I \implies \exists q' : q \Rightarrow q' \overset{a}{\not\to}$ y pRq'
(RT)	pRq y $p \overset{X}{\to}$ con $X \subseteq I(p) \implies \exists q' : q \Rightarrow q' \overset{X}{\to}$ y pRq'
(η)	pRq y $p \overset{a}{\to} p'$ con $a \in A^O (\cup A^I$ si $\rightleftarrows$ esta en la noción$) \implies \exists q', q'' : q \Rightarrow q'' \overset{a}{\to} \Rightarrow q' pRq''$ y $p'R$
$(\eta_{!!})$	pRq y $p \overset{a}{\to} p'$ con $a \in A^O (\cup A^I$ si $\rightleftarrows$ esta en la noción$) \implies \exists q', q'' : q \Rightarrow q'' \overset{a}{\to} q', pRq''$ y $p'Rq$
$(\eta_{\rightleftarrows_{IE}})$	pRq y $(p \overset{a}{\to} p'$ con $a \in A^O \cup A^I$ o $p \overset{a}{\not\to}, a \in A^I, p = p') \implies$ $\exists q', q'' : (q \Rightarrow q'' \overset{a}{\to} \Rightarrow q'$ o $q \Rightarrow q'' \overset{a}{\not\to}$ con $q'' = q')$ y se da que: $pRq'', p'Rq'$
$(\eta_{\rightleftarrows_{IE},!!})$	pRq y $(p \overset{a}{\to} p'$ con $a \in A^O \cup A^I$ o $p \overset{a}{\not\to}, a \in A^I, p = p') \implies$ $\exists q', q'' : (q \Rightarrow q'' \overset{a}{\to} q'$ o $q \Rightarrow q'' \overset{a}{\not\to}$ con $q'' = q')$ y se da que: $pRq'', p'Rq'$
(b)	pRq y $p \overset{a}{\to} p'$ con $a \in A^H, \implies \exists c \in A^H, q', q'' : q \Rightarrow q'', (q'' = q'$ o $q'' \overset{c}{\to} q'), pRq'', p'Rq'$
$(\neg)$	$pRq \implies qRp$
$(\neg_d)$	$pRq \implies \exists q' : q \Rightarrow q' \wedge q'Rp$
(Δ)	pRq y $p \overset{\tau}{\to} p_1 \overset{\tau}{\to} p_2 ...$ con $p_i \overset{a}{\not\to}$ para $a \in A^I \implies q \overset{\tau}{\to} q_1 \overset{\tau}{\to} q_2 ...$ con $q_i \overset{a}{\not\to}$ para $a \in A^I$
(λ)	pRq y $(p \not\to \vee\ p \overset{\tau}{\to} p_1 \overset{\tau}{\to} p_2 ...$ con $p_i \overset{a}{\not\to}$ para $a \in A^I) \implies (q \not\to \vee\ q \overset{\tau}{\to} q_1 \overset{\tau}{\to} q_2 ...$ con $q_i \overset{a}{\not\to}$ para a
(δ)	pRq o $p \overset{a}{\not\to}$ para $a \in A^{OH} \implies \exists q' : q \Rightarrow q' \wedge q' \overset{a}{\not\to}$ para $a \in A^{OH}$

Cuadro 3.15: Relational characterization conditions

Los tipos F, R y FT se demuestran de forma análoga.

$\eta \in N$ Supongamos $p \xrightarrow{a} p', a \in A^I$ debemos demostrar que existen q', q'' tal que $q \Rightarrow q'' \xrightarrow{a} q', O_N(p) \subseteq O_N(q'')$ y $O_N(p') \subseteq O_N(q')$. Supongamos por el absurdo que este no es el caso. Sean $Q'' = \{q'' \mid q \Rightarrow q'' \xrightarrow{a}, \exists \phi_{q''} \in O_N(p) - O_N(q'')\}$ y $Q' = \{q' \mid q \Rightarrow \xrightarrow{a} q', \exists \psi_{q'} \in O_N(p') - O_N(q')\}$. Luego $(\bigwedge_{q'' \in Q''} \phi_{q''}) \bigwedge (a \bigwedge_{q' \in Q'} \psi_{q'}) \in O_N(p)$ y como $O_N(p) \subseteq O_N(q)$, $(\bigwedge_{q'' \in Q''} \phi_{q''}) \bigwedge (a \bigwedge_{q' \in Q'} \psi_{q'}) \in O_N(q)$ lo cual llevaría a una contradicción a través de τ, $\bigwedge$, de forma similar al caso de RT.

Las variantes de η y el b se demuestran de forma similar.

$\neg \in N$ Supongamos $\phi \notin O_N(p)$, entonces $\neg\phi \in O_N(p) \subseteq O_N(q)$, por lo tanto $\phi \notin O_N(q)$.

Finalmente los tipos Δ y λ son directos, al igual que $\neg_d$.

La demostración de que si existe una *N-simulación* R_N tal que $p \; \mathsf{R}_N \; q$ luego $O_N(p) \subseteq O_N(q)$ no presenta mayores dificultades. La demostración procede por inducción en la estructura de la observación. A continuación la prueba para los casos casos a^O y $\neg$. La prueba para todos los otros tipos de observación sigue el lineamiento del primer caso.

$a^O \in N$ Sea $a\phi \in O(p)$. Luego existe p' tal que $p \Rightarrow \xrightarrow{a} p'$ y $\phi \in O(p')$. Por la cláusula $a^O \in N$ de la relación R_N debe existir q' tal que $q \Rightarrow \xrightarrow{a} \Rightarrow q'$ y $p' \; \mathsf{R}_N \; q'$. Por hipótesis inductiva $\phi \in O(q')$ y por lo tanto $a\phi \in O(q)$.

$\neg \in N$ $\neg\phi \in O(p)$ por lo tanto $\phi \notin O(p)$. Si $\phi \in O(q)$ luego por la cláusula $(\neg)$ y la hipótesis inductiva, $\phi \in O(p)$, absurdo. Por lo tanto $\phi \notin O(q)$ y $\neg\phi \in O(q)$.

□

Capítulo 4

Observaciones finales

4.1. Trabajos relacionados

En este trabajo se han estudiado semánticas para sistemas de transiciones concretos con acciones de entrada, de salida e internas. Para esto se utilizó el enfoque basado en nociones de observabilidad de [4] el cual permite definir las semánticas a partir de suposiciones individuales del modelo. La misma tarea se realizo parcialmente en [5] y [6]. En este trabajo, además de incluir nuevos tipos de observación, realizamos un estudio parcial sobre el preorden entre las nociones de observabilidad que nunca fue realizado hasta el momento.

Por otro lado, las semánticas presentadas en el trabajo están relacionadas con el marco desarrollado por la teoría de testing basada en la relación **ioco** (*Input Output Conformance*) [9]. En este contexto, las especificaciones y las implementaciones son modeladas por sistemas de transiciones con acciones de entrada y de salida. Como una implementación no tiene control sobre las acciones de entrada que recibe desde su ambiente estos sistemas deben ser input enabled, i.e. para toda acción de entrada $a?$ y estado s existe un estado s' tal que $s \xrightarrow{a?} s'$. Es natural que una implementación no reaccione en todo momento a todos los posibles estímulos; la falta de reacción ante un estímulo $a?$ se modela definiendo $s' = s$. Esta restricción no está presente en las especificaciones dado que son descripciones del comportamiento deseado. Esto es equivalente a la intuición detrás de los autómatas de interfaz. Además, en ambos tipos de sistemas se modela la necesidad del estímulo del ambiente para que el sistema progrese. Esto se realiza mediante la inclusión de una transición con una acción especial δ que no es ni de entrada, ni de salida y que indica que se está en presencia de un estado inactivo (*"quiescent state"*). Entonces todo estado s, tal que $s \not\xrightarrow{q}$ con a una acción de salida o interna, ejecuta una transición visible $s \xrightarrow{\delta} s$. En este contexto, el comportamiento de cada modelo está representado por sus trazas.

Las interfaces de autómatas y las nociones de observabilidad presentadas subsumen los modelos introducidos por esta teoría de testing. El comportamiento que describe la especificación puede describirse con una interfaz P_s a la cual se le aso-

cia una noción de observabilidad $sp = \{a^O, T, \bigvee, \rightleftarrows, \delta, \tau, \varepsilon\}$. Notemos que la forma de interacción está definida por el tipo de observación $\rightleftarrows$, el cual especifica que las acciones de entrada se ejecutan sólo si están habilitadas en el estado actual. Por otro lado, el comportamiento de la implementación puede modelarse por una interfaz P_i y una noción de observabilidad $im = \{a^O, T, \bigvee, \rightleftarrows, \delta, \tau, \varepsilon, \rightleftarrows_{IE}\}$. En este caso la forma de interacción está definida por los tipos de observación $\rightleftarrows$ y $\rightleftarrows_{IE}$. Luego todas las acciones de entrada pueden ejecutarse en todo momento pero si una de estas acciones es ejecutada en un estado dónde no se encuentra habilitada, ésta será descartada por el sistema. Este enfoque permite separar claramente la parte funcional de los modelos, expresada por los sistemas de transición, de las suposiciones sobre el modelo, expresada por los tipos que componen las nociones de observabilidad.

La relación **ioco** establece una relación de *conformidad* entre la implementación y la especificación, i.e. la implementación cumple con la especificación. La misma establece que la implementación cumple con la especificación si cada comportamiento posible en la implementación, el cual haya sido descripto por la especificación, no produce una acción de salida o δ que no haya sido especificado. Esta condición puede definirse utilizando las observaciones de cada sistema: P_i **ioco** P_s sii[1]

$$\forall \phi \in O_{sp}(P_s) : \phi\phi' \in O_{im}(P_i), \phi' \in \{a\varepsilon : a \in A^O\} \cup \{\delta\} \implies \phi\phi' \in O_{sp}(P_s)$$

4.2. Conclusiones

En este tesis se han estudiado distintas semánticas para sistemas con acciones de entrada y de salida. Para esto se aplico un enfoque basado en tipos de observaciones y nociones de observabilidad. Un aspecto sumamente positivo de este enfoque es que una vez definidos los tipos de observaciones y las nociones de observabilidad el espectro de semánticas queda totalmente definido. La parte fundamental para realizar esto es definir primero las suposiciones que rigen el modelo. Notar que esto fue esencial para justificar la diferencia con los sistemas de transiciones que no diferencian entre los dos tipos de acciones.

Cabe destacar que el conjunto de tipos de observaciones aquí presentados difiere con el conjunto presentado en [4]. Por ejemplo, no hemos presentado distintas versiones de los operadores para estados estables; tampoco se ha estudiado la posibilidad de observar branching infinitos. Más aún, también podrían agregarse variantes a los tipos aquí presentados, entre ellos podemos destacar los tipos F, R, FT y RT pero para acciones de salida. Notar que éstos sólo tendrían sentido en un contexto donde el observador cuente con un dispositivo que le permita observar las posibles acciones (de salida) que podría ejecutar el sistema.

Con respecto a los resultados prensentados en [6], se agregó el tipo de observación $\neg_d$. Esto no presento mayores problemas. Además se refino la Def. 7 (noción

[1] Se omiten los T al final de las observaciones ϕ y $\phi\phi'$.

de observabilidad). La definición presentada en [6] no presentaba la condición 6. Cómo ya se menciono, el estudio sobre el preorden de las nociones de observabilidad es original a este trabajo.

El objetivo principal de este trabajo fue comprender en profundidad los resultados presentados en [11]. Luego, extender los resultados similares para sistemas interactivos de [5] y [6]. La tarea fue sumamente compleja dado que el marco utilizado es complejo y presenta muchas sutilezas. Esto sucede principalmente al momento de definir el orden parcial entre las nociones de observabilidad, por este razón, el estudio no abarco el total de las semánticas. Más allá de esto, el objetivo consideramos haber logrado el objetivo con creces.

Bibliografía

[1] J. Bakker, J. Kok, J.-J. Meyer, E.-R. Olderog, and J. Zucker. Contrasting themes in the semantics of imperative concurrency. In J. Bakker, W. Roever, and G. Rozenberg, editors, *Current Trends in Concurrency*, volume 224 of *Lecture Notes in Computer Science*, pages 51–121. Springer Berlin Heidelberg, 1986.

[2] L. de Alfaro and T. Henzinger. Interface-based design. In M. Broy, J. Grünbauer, D. Harel, and T. Hoare, editors, *Engineering Theories of Software Intensive Systems*, volume 195 of *NATO Science Series*, pages 83–104. Springer, 2005.

[3] L. de Alfaro and T. A. Henzinger. Interface automata. In *Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering*, ESEC/FSE-9, pages 109–120, New York, NY, USA, 2001. ACM.

[4] R. J. v. Glabbeek. The linear time - branching time spectrum II. In E. Best, editor, *CONCUR*, volume 715 of *Lecture Notes in Computer Science*, pages 66–81. Springer, 1993.

[5] M. Lee and P. R. D'Argenio. Semantics for interactive sequential systems and non-interference properties. *CLEI Electron. J.*, 14(3), 2011.

[6] M. D. Lee. Semántica de procesos para sistemas interactivos y sistemas probabilísticos. Master's thesis, Universidad Nacional de Córdoba, Córdoba, 2013.

[7] J. Tretmans. Conformance testing with labelled transition systems: Implementation relations and test generation. *Computer Networks and ISDN Systems*, 29(1):49–79, 1996.

[8] J. Tretmans. Formal methods and testing. chapter Model based testing with labelled transition systems, pages 1–38. Springer-Verlag, Berlin, Heidelberg, 2008.

[9] J. Tretmans. Model based testing with labelled transition systems. In R. M. Hierons, J. P. Bowen, and M. Harman, editors, *Formal Methods and Testing*,

volume 4949 of *Lecture Notes in Computer Science*, pages 1–38. Springer, 2008.

[10] R. van Glabbeek. The linear time-branching time spectrum i - the semantics of concrete, sequential processes. In *Handbook of Process Algebra, chapter 1*, pages 3–99. Elsevier.

[11] R. van Glabbeek. The linear time-branching time spectrum ii - the semantics of sequential systems with silent moves, 1993.

[12] D. Walker. Automated analysis of mutual exclusion algorithms using CCS. *Formal Aspects of Computing*, 1:273–292, 1989.

[13] P. Wegner. Why interaction is more powerful than algorithms. *Commun. ACM*, 40(5):80–91, May 1997.