

Cuarta Fase

Para añadir movimiento al sistema estático definido en la tercera fase tenemos que simplificarlo. Así que esta cuarta fase de la práctica consiste, primero, en reducir el modelo para, a continuación, definir procedimientos que permitan hacer que el sistema evolucione.

Simplificación del modelo de la tercera fase

Para simplificar el sistema **elimina** (basta con que los comentarios) o **modifica** las relaciones y los hechos del sistema de la tercera fase que se enumeran a continuación:

1. Elimina las relaciones **destino** y **puertoEntrada** de la signatura **Regla**
2. Modifica la relación **reenviar** para que asocie cada **Regla** con a lo sumo un **Puerto**
3. Elimina las relaciones **origen** y **destino** de la signatura **Paquete**
4. Elimina la relación **puertosAlcanzablesDesde** en la signatura **Nodo**
5. Elimina los siguientes hechos de la segunda fase:
 - a) los hosts origen y destino de cada paquete deben ser distintos.
 - b) para cada paquete existe un *camino* en la red desde su posición actual hasta el nodo destino. Un camino es un recorrido a través de los puertos conectados en la red.
6. Elimina los siguientes hechos de la tercera fase:
 - a) el puerto de entrada de una regla no puede ser uno de los puertos por los que hay que reenviar el paquete.
 - b) no hay ciclos en la red, es decir, las reglas de los switches deben asegurar que ningún paquete que ha visitado un switch vuelve al switch antes de llegar a su destino.
7. Modifica los siguientes hechos de la tercera fase:
 - a) cada regla pertenece **a lo sumo** a una tabla.
 - b) los puertos de salida de cada regla son del switch en los que se encuentra la regla (si está en un switch).

Extensión con tiempo del modelo simplificado

Una vez simplificado el sistema realiza la siguientes modificaciones sobre el mismo:

1. Define una signatura **Tiempo** para representar distintos instantes de tiempo
2. Modifica las relaciones **posicion** y **tabla** de manera que puedan cambiar con el tiempo
3. Modifica los hechos del sistema que tienen que ver con las nuevas relaciones **posicion** y **tabla** para que sean consistentes con su nueva definición (tienen que tener en cuenta el tiempo).
4. Define un predicado **pred configuracion()** que permita configurar el sistema con 2 **Switches** y 3 **Hosts**.
5. Define un predicado **pred inicio(t:Tiempo)** que defina el estado inicial en el instante **t**: todas las tablas de los switches están vacías y los paquetes no se encuentran en ningún puerto inicialmente.

6. Define un predicado `pred nuevaReglaControlador(t,t':Tiempo,s:Switch,p:Paquete)` que modela el envío del paquete de control `p` con una nueva regla del controlador al switch `s`.
7. Define un predicado `pred nuevaReglaSwitch(t,t':Tiempo,s:Switch,p:Paquete)` que modela que el switch `s` procesa el paquete de control `p`, añadiendo a su tabla la nueva regla que lleva el paquete.
8. Define un predicado `pred paqueteHostSwitch(t,t':Tiempo,p:Paquete,h:Host,s:Switch)` que modela la transmisión de un paquete del host `h` al switch `s` a través de sus puertos. Nota: el paquete podría no estar en ningún puerto del host.
9. Define un predicado `paqueteSwitchEnviar(t,t':Tiempo,paq:Paquete,s:Switch)` que modela que el switch `s` reenvía el paquete `paq` (que no es un paquete de control) utilizando una de sus reglas (se supone que el switch tiene una regla con el tipo de `paq`).
10. Utilizando la sentencia `open util / ordering [Tiempo]` que ordena los instantes de tiempo, define el predicado `pred traza()` que configura e inicializa el sistema en el instante inicial y luego itera sobre los instantes de tiempo aplicando de manera indeterminista los predicados `nuevaReglaControlador`, `nuevaReglaSwitch`, `paqueteHostSwitch` y `paqueteSwitchEnviar`.

FIN DE LA SEGUNDA PARTE DE LA PRÁCTICA

Se deben respetar los nombres de las firmas y relaciones del enunciado. Si algún estudiante lo ve estrictamente necesario puede añadir más firmas/relaciones, pero explicando claramente su necesidad y uso. Se pueden añadir las funciones y predicados que se vea necesario con el objetivo de simplificar los asertos.

Recuerda que esta es una práctica que debéis hacer en parejas. Los dos componentes deben participar IGUALMENTE en el desarrollo de la práctica, y además la documentación entregada DEBE ser el fruto del trabajo EXCLUSIVO de cada pareja. Cualquier anomalía que se detecte en este sentido se tendrá en cuenta en la evaluación de la práctica.

Documentos a entregar

- D1. Cuatro ficheros `.als` que se correspondan con cada una de las fases en las que se ha dividido la práctica. En cada fichero, debe estar claro qué hechos (`fact`), predicados, o funciones pertenecen a cada una de las fases. Cada restricción, predicado y función debe estar acompañada de un comentario que explique su funcionalidad.
- D2. Ejemplos representativos de instancias para cada una de las cuatro fases de la práctica.

Recuerda que esta es una práctica que debéis hacer en parejas. Los dos componentes deben participar IGUALMENTE en el desarrollo de la práctica, y además la documentación entregada DEBE ser el fruto del trabajo EXCLUSIVO de cada pareja. Cualquier anomalía que se detecte en este sentido se tendrá en cuenta en la evaluación de la práctica.