



## MÉTODOS FORMALES PARA INGENIERÍA DE SOFTWARE

### Examen Parcial

17 de Octubre de 2024

Los repositorios en la nube traen aparejados algunos problemas de sincronización sobre la información almacenada en ellos. Considere el repositorio “*ProximaNube*”. El repositorio lleva un registro de los pedidos de actualización que vayan llegando.

#### ***Estrategia nuevo átomo:***

Considere el modelo dado en el anexo 1. Asumiendo que el modelo estático es correcto y que la política de aplicación de las actualizaciones es aplicar una prioritaria, en caso de que haya alguna, y en caso contrario tomar al azar una de las solicitudes pendientes, resuelva los siguientes incisos.

**Importante: Resuelva el tercer inciso en un archivo separado.**

1. Valide con al menos dos comandos de éxito y no éxito el predicado `AplicarActualizacion[a1,a2: Archivo]`.
2. Provea una nueva versión del predicado `AplicarActualizacion`, llámela `AplicarActualizacionCorregido[a1,a2: Archivo]`. En esta nueva versión corrija las irregularidades encontradas en el inciso anterior. Valide este predicado con al menos dos comandos de éxito y dos de no éxito.
3. Como se está representando la dinámica para las actualizaciones de un solo archivo, ¿puede definirse la signatura `Archivo` como unitaria? Justifique su respuesta. Muestre el efecto de este cambio utilizando el analizador.

#### ***Estrategia nuevo patrón:***

Considere ahora el modelo dado en el anexo 2, pensado para modelar la dinámica del escenario planteado a través de la noción de estado.

En este caso asuma que el repositorio tiene como política de sincronización permitirle el acceso a la última solicitud de actualización, a menos que solo haya una única solicitud de actualización pendiente o haya alguna solicitud identificada como prioritaria.

Tenga en cuenta que las únicas acciones posibles, para establecer cambios de estado, son las siguientes:

- **Llegada de una solicitud de actualización:** Al llegar una solicitud de actualización, la misma debe registrarse en la relación *ultimaActualización*. En caso que haya una solicitud previa en dicha relación, la solicitud previa debe moverse a *actualizacionesPendientes*.
- **Aplicación de la última solicitud de actualización:** Es posible realizar la operación `AplicarUltimaActualizacion` si la solicitud almacenada en la relación *ultimaActualizacion* es prioritaria, o bien si no lo es pero no hay actualizaciones prioritarias pendientes. En caso de que la actualización aplicada sea prioritaria, se eliminan todas las actualizaciones pendientes. En caso de que la actualización aplicada no sea prioritaria, se coloca como *ultimaActualizacion* alguna de las actualizaciones que estaban pendientes.
- **Aplicación de una solicitud de actualización prioritaria:** La operación `AplicarActualizacionPrioritaria` efectúa una actualización prioritaria que se encuentre pendiente. Al aplicarla se deben eliminar todas las actualizaciones que estaban pendientes al momento de registrar en la relación *ultimaActualizacion* la actualización aplicada.

Teniendo en cuenta todo lo anterior, resuelva los siguientes incisos. **Importante: Resuelva los incisos en archivos separados.**

1. Evalúe si el modelo, estático, dado en el archivo anexo `NubeEstadosEstatico.als` es adecuado. Deje comentarios sobre los comandos utilizados y los resultados obtenidos/problemas encontrados. **No modifique el modelo en este inciso.**
2. Corrija los errores detectados en el inciso anterior y verifique que el modelo sea correcto estáticamente. Puede modificar el modelo, agregar, eliminar y corregir facts según considere necesario. Deje expresado en comentarios lo modificado justificando los cambios realizados (haga mención al comentario del punto anterior que da lugar a la situación).
3. Muestre las modificaciones que sería necesario efectuar en el modelo, estático, para considerar la existencia de dos átomos de la signatura *Archivo*. ¿Como se deberían generalizar los cambios para considerar cualquier cantidad de átomos de la signatura *Archivo*? Muestre ambas definiciones en un mismo archivo `.als`
4. Añadir al archivo resultante del inciso 2 la definición de la parte que involucra dinámica, que se encontraba originalmente comentada en el archivo del anexo. Validar el predicado `AplicarUltimaActualizacion[e1,e2: Estado]`. Asumir que los otros dos predicados de dinámica ya fueron validados y son correctos.
5. Corregir el predicado para subsanar las falencias identificadas en el inciso anterior. Revalide el predicado. Agregue un comando para validar si es posible aplicar, haciendo uso de dicho predicado, una actualización no prioritaria habiendo actualizaciones prioritarias pendientes. Dejar expresado en un comentario cuál es la respuesta del analizador. En caso de que sea posible, corrija el modelo para evitarlo.
6. Verificar si se cumple la siguiente propiedad para todos los estados:

*Si se aplica una solicitud no prioritaria, no quedan solicitudes pendientes en el próximo estado.*

Dejar expresado en un comentario cuál es la respuesta del analizador y su interpretación sobre la misma.