Verificación de Sistemas Reactivos
Introducción

Alejandro Sánchez

Departamento de Informática
Universidad Nacional de San Luis

Maestría en Ingeniería de Software
Maestría en Calidad del Software
Especialización en Ingeniería de Software
25-26 Octubre 2013

Estructura de la presentación

1. Caracterización de sistemas reactivos
2. Enfoque para la verificación de sistemas reactivos
3. Objetivo y agenda del curso

Estructura de la presentación

Definición

Sistema reactivo

Un sistema que computa
al reaccionar a estímulos de su ambiente

Dos tipos de sistemas - Ejemplos

Computacionales

- Compilador (ej. Java, C++)
- Programa de liquidación de haberes
- Algoritmo de cálculo del trayecto más corto
- Consulta a una base de datos

Reactivos

- Sistemas operativos
- Protocolos de comunicación (ej. servicio web)
- Programas de control (ej. elevador, central nuclear)
- Software ejecutando en sistemas embebidos (ej. teléfonos móviles)

¿Diferencias?

Dos tipos de sistemas - Ejemplos

Computacionales

- Compilador (ej. Java, C++)
- Programa de liquidación de haberes
- Algoritmo de cálculo del trayecto más corto
- Consulta a una base de datos

Reactivos

- Sistemas operativos
- Protocolos de comunicación (ej. servicio web)
- Programas de control (ej. elevador, central nuclear)
- Software ejecutando en sistemas embebidos (ej. teléfonos móviles)

¿Diferencias?

Dos tipos de sistemas - Contraste

Computacionales

Transforman una entrada en una salida

- Terminación deseable
- Resultado único
- Secuenciales sin interacciones
- Semántica dada por función parcial $S \hookrightarrow S$

Reactivos

Computan al reaccionar a estímulos de su ambiente

- Terminación no deseable
- Resultados (si hay) puede no ser único
- Concurrentes con interacciones
- ?

¿Cómo desarrollar sistemas reactivos que “funcionen”?
¿Cómo analizar y verificar tales sistemas?

Dos tipos de sistemas - Contraste

Computacionales

Transforman una entrada en una salida

- Terminación deseable
- Resultado único
- Secuenciales sin interacciones
- Semántica dada por función parcial $S \hookrightarrow S$

Reactivos

Computan al reaccionar a estímulos de su ambiente

- Terminación no deseable
- Resultados (si hay) puede no ser único
- Concurrentes con interacciones
- ?

¿Cómo desarrollar sistemas reactivos que “funcionen”?
¿Cómo analizar y verificar tales sistemas?

Dos tipos de sistemas - Contraste

Computacionales

Transforman una entrada en una salida

- Terminación deseable
- Resultado único
- Secuenciales sin interacciones
- Semántica dada por función parcial $S \hookrightarrow S$

Reactivos

Computan al reaccionar a estímulos de su ambiente

- Terminación no deseable
- Resultados (si hay) puede no ser único
- Concurrentes con interacciones
- ?

¿Cómo desarrollar sistemas reactivos que “funcionen”?
¿Cómo analizar y verificar tales sistemas?

Dos tipos de sistemas - Contraste

Computacionales

Transforman una entrada en una salida

- Terminación deseable
- Resultado único
- Secuenciales sin interacciones
- Semántica dada por función parcial $S \hookrightarrow S$

Reactivos

Computan al reaccionar a estímulos de su ambiente

- Terminación no deseable
- Resultados (si hay) puede no ser único
- Concurrentes con interacciones
- ?

¿Cómo desarrollar sistemas reactivos que “funcionen”?
¿Cómo analizar y verificar tales sistemas?

Dos tipos de sistemas - Contraste

Computacionales

Transforman una entrada en una salida

- Terminación deseable
- Resultado único
- Secuenciales sin interacciones
- Semántica dada por función parcial $S \hookrightarrow S$

Reactivos

Computan al reaccionar a estímulos de su ambiente

- Terminación no deseable
- Resultados (si hay) puede no ser único
- Concurrentes con interacciones
- ?

¿Cómo desarrollar sistemas reactivos que “funcionen”?
¿Cómo analizar y verificar tales sistemas?

Dos tipos de sistemas - Contraste

Computacionales

Transforman una entrada en una salida

- Terminación deseable
- Resultado único
- Secuenciales sin interacciones
- Semántica dada por función parcial $S \hookrightarrow S$

Reactivos

Computan al reaccionar a estímulos de su ambiente

- Terminación no deseable
- Resultados (si hay) puede no ser único
- Concurrentes con interacciones
- ?

¿Cómo desarrollar sistemas reactivos que “funcionen”?

¿Cómo analizar y verificar tales sistemas?

Dos tipos de sistemas - Contraste

Computacionales

Transforman una entrada en una salida

- Terminación deseable
- Resultado único
- Secuenciales sin interacciones
- Semántica dada por función parcial $S \hookrightarrow S$

Reactivos

Computan al reaccionar a estímulos de su ambiente

- Terminación no deseable
- Resultados (si hay) puede no ser único
- Concurrentes con interacciones
- ?

¿Cómo desarrollar sistemas reactivos que “funcionen”?
¿Cómo analizar y verificar tales sistemas?

Estructura de la presentación

Teoría de sistemas reactivos

Se necesita una teoría de sistemas reactivos y sus aplicaciones que soporte:

- Diseño (desarrollo)
- Especificación (requerimiento)
- Análisis
- Verificación (automática y composicional)

Diseño

¿Cómo modelar un sistema reactivo?

Un proceso que realiza una acción
y se convierte en otro proceso

Labelled transition system (LTS)

Process algebra

$$A = set.B$$
$$B = alarm.B + reset.A$$

Diseño

¿Cómo modelar un sistema reactivo?

Un proceso que realiza una acción
y se convierte en otro proceso

Labelled transition system (LTS)

Process algebra

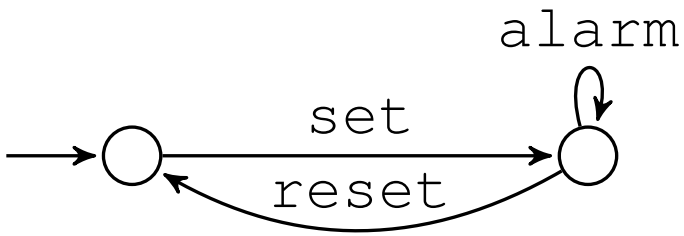
$$A = set.B$$
$$B = alarm.B + reset.A$$

Diseño

¿Cómo modelar un sistema reactivo?

Un proceso que realiza una acción
y se convierte en otro proceso

Labelled transition system (LTS)



Process algebra

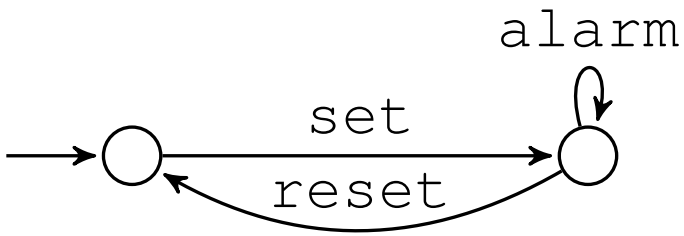
$$A = set.B$$
$$B = alarm.B + reset.A$$

Diseño

¿Cómo modelar un sistema reactivo?

Un proceso que realiza una acción
y se convierte en otro proceso

Labelled transition system (LTS)



Process algebra

$$A = set.B$$
$$B = alarm.B + reset.A$$

Especificación

¿Cómo especificar requerimientos para un sistema reactivo?

- Modelo
  - Especificar un modelo de los requerimientos
  - Comparar (refinamiento) con el modelo de la implementación
- Propiedades
  - Especificar propiedades (en una lógica)
  - Establecer si el modelo las satisface

Especificación

¿Cómo especificar requerimientos para un sistema reactivo?

- Modelo
  - Especificar un modelo de los requerimientos
  - Comparar (refinamiento) con el modelo de la implementación
- Propiedades
  - Especificar propiedades (en una lógica)
  - Establecer si el modelo las satisface

Especificación

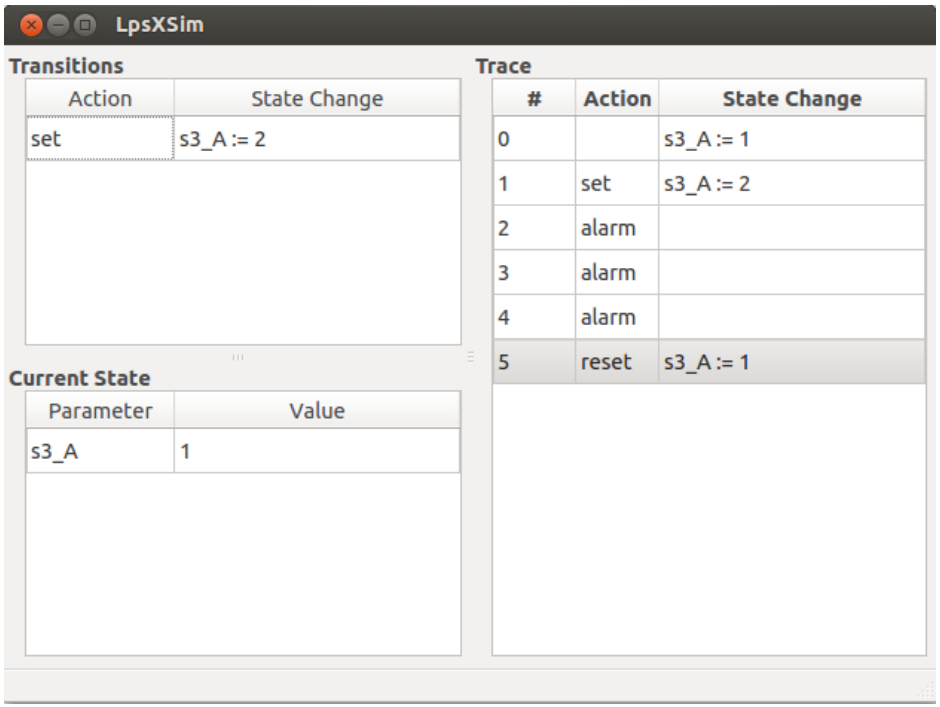
¿Cómo especificar requerimientos para un sistema reactivo?

- Modelo
  - Especificar un modelo de los requerimientos
  - Comparar (refinamiento) con el modelo de la implementación
- Propiedades
  - Especificar propiedades (en una lógica)
  - Establecer si el modelo las satisface

Análisis

¿Cómo estudiar un sistema reactivo?

- Visualización
- Animación
- Refinamiento (implementación correcta)
- Equivalencia
  - transformar un modelo
  - distinguir entre modelos



Verificación

¿Cómo verificar un sistema reactivo?

- Especificar propiedades utilizando una lógica (modal)
- Verificar si el modelo (álgebra de procesos) las satisface
  - automaticamente
  - composicionalmente

Estructura de la presentación

Objetivo

Presentar una teoría general de sistemas reactivos y sus aplicaciones que soporte el diseño, la especificación, el análisis y la verificación de los mismos.

Que el alumno pueda:

- Modelar y especificar sistemas reactivos en un marco preciso
- Analizar y verificar sistemas reactivos

Agenda

- 25 de Octubre
  - Introducción
  - Labelled Transiftion Systems (LTSs)
  - Álgebras de Procesos
    - Calculus of Communicating Systems (CCS)
    - mCRL2
- 26 de Octubre
  - Introducción a las lógicas modales
  - Arquitectura de software de sistemas reactivos

Bibliografía I

- Alessandro Aldini, Marco Bernardo, and Flavio Corradini, A Process Algebraic Approach to Software Architecture Design, vol. 54, Springer London, 2010.
- Robert Allen and David Garlan, A formal basis for architectural connection, ACM Trans. Softw. Eng. Methodol. **6** (1997), no. 3, 213–249.
- Luca Aceto, Anna Ingólfsdóttir, Kim Guldstrand Larsen, and Jiri Srba, Reactive systems: Modelling, specification and verification, Cambridge University Press, New York, NY, USA, 2007.
- J. C. M. Baeten, T. Basten, and M. A. Reniers, Process algebra: Equational theories of communicating processes, Cambridge University Press, 2010.

Bibliografía II

- J. F. Groote, A. Mathijssen, M. Reniers, Y. Usenko, and M. van Weerdenburg, The formal specification language mCRL2, Methods for Modelling Software Systems: Dagstuhl Seminar 06351, 2007.
- mCRL2 web site, `http://www.mcrl2.org`, 2013.
- R. Milner, Communicating and mobile processes: the π-calculus, Cambridge University Press, 1999.
- Alejandro Sanchez, Luis S. Barbosa, and Daniel Riesco, A language for behavioural modelling of architectural patterns, Proceedings of the Third Workshop on Behavioural Modelling (New York, NY, USA), BM-FA '11, ACM, 2011, pp. 17–24.

Bibliografía III


Richard N. Taylor, Nenad Medvidovic, and Eric M. Dashofy, Software architecture : foundations, theory, and practice, Wiley, January 2009.