

# Riordino intelligente dei farmaci tramite modelli MDP

M.Biagi   **T.Papini**

STLab, Dipartimento d'Ingegneria dell'Informazione, Università degli Studi di Firenze, Italia,  
{marco.biagi,tommaso.papini}@unifi.it

31 Marzo 2017

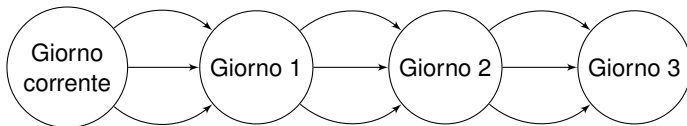
- Tecniche basate su MDP
  - modellazione del reparto
  - attualizzazione del modello
  - sperimentazione

# Panoramica

- 1 Il modello generale
  - Struttura del modello
  - Specifiche e limitazioni
  
- 2 Utilizzo del modello
  - Strumenti utilizzati
  - Calcolo della strategia ottima
  
- 3 Sperimentazione

## Struttura del modello

- Modellazione di più giorni consecutivi
  - giorno corrente (fine giornata)
  - 3 giorni successivi
- probabilità e non-determinismo
  - evoluzione probabilistica di ogni giornata
  - scelte non-deterministiche sull'ordine



## Specifiche e limitazioni

### ■ Reparto

- un reparto con posologia fissa
- capienza del reparto fissata
- capienza del magazzino fissata

### ■ Farmaco

- un solo tipo di farmaco

### ■ Caratterizzazione probabilistica

- pazienti in arrivo (programmati/pronto soccorso)
- pazienti in uscita
- consumo di farmaci per i pazienti in reparto

### ■ Scelte non-deterministiche

- se e in che quantità riordinare  $\{0, 10, 20, 30, 40\}$

### ■ Funzione di costo

- costo dell'ordine di ogni unità di farmaco
- costo della giacenza di ogni farmaco in magazzino
- costo di farmaci mancanti da somministrare

## Strumenti utilizzati

- Componente Java

- simulazione
- attualizzazione
- estrazione dell'avversario
- sperimentazione

- Prism

- Implementazione del modello attualizzato
- Calcolo della strategia ottima

## Calcolo della strategia ottima

Ogni giorno, il modello viene aggiornato sulla base dei dati attuali.

- **Simulatore**: genera dati verosimili e consistenti
  - implementa le stesse distribuzioni di probabilità definite nel modello Prism
- **Template**: rappresenta un generico modello Prism non aggiornato
  - aggiornato tramite il template engine Apache Velocity

Sul modello aggiornato, Prism può calcolare la strategia ottima

- Da componente Java
  - viene estratto l'avversario
  - che indica la scelta ottima

La scelta ottima suggerita da Prism **minimizza** il costo **medio** sui 3 giorni successivi!

# Sperimentazione

**state: num patients = 20**

