

## Introduzione

Realizzazione di un generico risponditore telefonico. Si inizia con il pensare al modello che rappresenta tale risponditore e le funzionalità che implementerebbe, si potrebbe considerare come una macchina a stati finiti.

## Contesto Ideale

Il contesto è quello della **facilitazione dell'interazione** al telefono delle persone anziane (e non solo).

Supponiamo di avere un sistema che consente ad una persona anziana di poter chiedere a voce (senza alzare la cornetta), ad un sistema che è in ascolto nel luogo dove si trova, delle richieste di servizio.

Tali richieste che potrebbero attivare una chiamata ad una certa persona, magari riconosciuta da una certo *alias* (nome e cognome).

L'idea è quella di **simulare**, un sistema che rappresenta la funzionalità del risponditore telefonico, tramite Model Checker, che esplori quali sono gli stati in cui si può trovare e verificare se partendo da uno stato iniziale si arriverà o meno ad uno stato indesiderato (di fallimento).

## Considerazioni

L'obiettivo è quello di concepire tale sistema che possa sostituire in modo **assoluto** la cornetta e la composizione di un numero. In modo più specifico per realizzarlo ci si pongono le seguenti considerazioni:

- Come si dovrebbe attivare?
- Quali frasi dovrebbero essere emesse al sistema per dire alla persona anziana, in che fase del protocollo si trova?
- Quali sono le fasi invece, da parte dell'anziano per attivare il servizio?
- Cosa prevede il protocollo?
- Come dovrebbe attivarsi questo dialogo?
- Potrebbe esserci il rischio di un loop infinito?
- Situazione dalla quale non si possa uscire più? Perché non c'è più una frase in cui sbloccare il sistema? (Deadlock)
- Oppure situazione dove il sistema non ha altre alternative che riproporre una situazione che non ha più senso o già riproposta?

## Contesto Reale

C'è un grosso bisogno di migliorare le condizioni di vita degli anziani, soprattutto nei casi più delicati.

Le RSA chiedono in generale come migliorare la loro performance con le persone anziane.

L'obiettivo reale è rendere più facile l'interazione con l'entourage familiare, che è fuori dall'RSA e anche come migliorare il monitoraggio per gli stessi.

## Introduzione al Problema Tecnico

Il problema tecnico deriva dall'utilizzo dei device collegati in wi-fi con vari sensori, che studiando il mondo esterno riescono a capire se la persona è a letto, sulla sedia, se sta chiedendo aiuto, se è in bagno, ecc..

Questi device infatti diventano **sorgenti di eventi** che il sistema dovrà gestire e provvedere a creare un quadro chiaro, in tempi accettabili, alla persona umana che monitora cosa sta succedendo nell'RSA.

Deve facilitare un operatore se nell'RSA sta andando tutto bene.

## Problema Tecnico

Quanto illustrato evidenzia che tutti questi eventi sono **concorrenti**: andrebbero in qualche maniera gestiti e messi a sistema.

Il bisogno è quello di **progettare sistemi concorrenti che siano robusti**, (se una persona va in bagno e *non chiude la porta*, questo non dovrebbe mettere in crisi il sistema provocando una starvation o un deadlock ).

## Riflessioni Finali

I Model Checker sono stati utilizzati per verificare protocolli di qualunque tipo, dagli ascensori, ai protocolli per la sicurezza e sono adatti anche in questo contesto.

Implementano il concetto di logica temporale così da inglobare anche il campo dell'IA.

Ci sono tanti strumenti per la verifica dei sistemi concorrenti, SPIN è sicuramente il più famoso e storico.

Dall'IA sappiamo che fare ricerca nello spazio degli stati è computazionalmente problematico, costoso, a meno che non si vada in *depth first* in modo da contenere linearmente la complessità spaziale pur rimanendo quella temporale.

Con spin appunto si può andare sia in depth first (*profondità*) che in breadth first (ampiezza) nella ricerca dello spazio degli stati.

In ogni caso se gli stati sono tanti, è evidente che anche con un sistema potente come SPIN che potrebbe sfruttare anche il multi core potrebbe crollare.

Nella realtà, per non mettere in ginocchio SPIN, esso viene applicato a modelli semplificati ( anche nell'IA si ha un tecnica analoga, quando si parla di euristiche da problemi rilassati ) sul dispositivo che si progetta: infatti più variabili ci sono dentro, più valori possono assumere questi variabili, più gli stati ammessi dal sistema sono tanti, più diventa complesso per un computer potente fare una verifica che non sia raggiungibile dallo stato iniziale uno stato indesiderato.

## Obiettivo Reale del Progetto

Prendere confidenza sui Model Checker e ragionare su **SPIN** apprendendo il suo utilizzo concretamente.

Progetto con 5/6 stati al massimo da verificare con il Model Checker SPIN.

Lo scopo non sarà quello di realizzare un sistema vero e proprio come descritto contestualmente per le RSA ma quello di imparare ad utilizzarlo, partendo dalle slides del corso, poi approfondendolo con il testo di Holzmann, sul web, fino ad applicarlo in pratica.