Sviluppo di un contatore elettrico intelligente

Stefano Antonio Labianca

22 Gennaio 2024

matricola: 758364 **email:** s.labianca10@studenti.uniba.it

Tabella dei contenuti

1 Introduzione

Nell'arco della nostra giornata, usiamo diversi dispositivi elettronici e, alle volte, anche per diverse ore della giornata o addirittura per tutto il giorno.

Per chi abita nelle zone di campagna, o in abitazioni singole, usare molti dispositivi elettronici contemporaneamente, specialmente se hanno alti consumi o possiedono una classe energetica bassa, fa scattare il salvavita.

1.1 Dispositivo salvavita

Il "salvavita", o più propriamente detto interruttore differenziale, è un dispositivo che arresta il flusso di energia elettrica dal contatore di un'abitazione, proteggendo persone e animali.

Figura 1: Esempio di contatore differenziale

Questi interruttori, monitorano la differenza di corrente in entrata e in uscita dal dispositivo e, quando la differenza di corrente in entrata e in uscita supera una certa soglia, allora l'interruttore scatta togliendo l'alimentazione al circuito.

1.2 Obiettivo del progetto

Il progetto si pone l'obiettivo di sviluppare un programma in grado di svolgere i seguenti task:

1. Determinare da una lista di dispositivi, quali possono tenere accesi contemporaneamente senza che salti il salvavita.
2. Tenere traccia dei dispositivi elettronici e del loro consumo in Watt.
3. Ottenere tutti quei dispositivi che rispettano certi vincoli di consumo energetico.

2 Progetto

2.1 Struttura del progetto

All'intero del progetto, possiamo trovare le seguenti cartelle:

- `/appliance`: Qui è possibile trovare la classe Appliance, che definisce un elettrodomestico, insieme ad una serie di metodi di suppporto, contenuti in `appliances_controller.py`
- `/cli`: Troviamo una classe che incapsula tutta la logica legata agli input e all'output del terminale
- `/csp_problem`: Questa cartella contiene tutti i file legati all'argomento del CSP.
  Infatti è possibile trovare la rappresentazione delle variabili e dei vincoli, fatta rispettivamente usando le classi Variable e Constraint.
  Inoltre è presente anche la classe CSP usata come wrapper per rappresentare un generico problema di questa categoria.
  Infine è presenta la cartella `/algorithm` che contiene le realizzazioni degli algoritmi DFS e GAC usati per risolvere i problemi legati al CSP.
- `/knowledge_base`: Contiene una classe usata per rappresentare un Sistema Esperto.
- `/ontology`: Questa cartella contiene una classe che permette di manipolare l'ontologia contenuta all'intero del file `appliance_ontology.rdf`.
- `/test`: Contiene tutti quei file contenente vari test fatti al programma.
- `/utils`: Contiene file di utilità che facilitano alcune operazioni interne al programma. Per esempio, il file `pagination.py` viene usato per impaginare l'output del programma.

2.2 Inizializzare il progetto

2.3 Avviare il progetto