

# Progetto di Ragionamento Automatico

Elia Piacentini - VR448249

23 gennaio 2020

## Indice

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	Consegna del progetto . . . . .	1
1.2	Scelta del linguaggio per lo sviluppo . . . . .	1
<b>2</b>	<b>L'idea alla base</b>	<b>1</b>
	Introduzione	

## 1 Consegna del progetto

Implementare in un linguaggio di programmazione a scelta la procedura di decisione, basata su chiusura di congruenza, della soddisfacibilità di un insieme di letterali nel frammento senza quantificatori dell'unione delle teorie dell'uguaglianza, delle liste non vuote possibilmente cicliche, e degli array senza estensionalità. Si implementi l'algoritmo di chiusura di congruenza su grafo diretto aciclico, considerando le varianti viste in classe (e.g., lista proibita, scelta non arbitraria del rappresentante della classe generata da union, aggiornamento del campo find di tutti i nodi di una classe).

## 2 Scelta del linguaggio per lo sviluppo

Ottenuta la consegna, il primo dubbio da risolvere era relativo al linguaggio in cui implementare il solutore; l'unico requisito per il linguaggio era che fosse eseguibile su Linux. Degli aspetti da non sottovalutare erano la velocità di sviluppo (visto il tempo a disposizione) e la velocità di esecuzione (che sarebbe potuta essere un parametro di valutazione). I linguaggi presi in considerazione erano tre:

- Java
- Python
- Dart

JAVA, il linguaggio che conosco meglio, è abbastanza veloce in esecuzione ma avrei dovuto spendere troppo tempo per strutturare il codice. Dart, simile a JAVA come prestazioni, avrebbe potuto rappresentare una scelta interessante: la compatibilità sarebbe stata garantita tramite un'interfaccia web e questo mi avrebbe anche agevolato nel creare una UI; purtroppo la mia conoscenza di Dart era troppo bassa per cimentarmi in un progetto come questo. L'ultimo rimasto, Python, non è certo il linguaggio più veloce, ma sicuramente uno dei più versatili. La velocità di sviluppo e di debug sono discretamente alte ed esistono parecchie librerie per gestire la parte grafica. Ho quindi scelto di sviluppare il progetto in Python, nello specifico usando la versione 3.7. Analisi del solutore

## 3 L'idea alla base