

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE



CORSO DI LAUREA IN INFORMATICA

Implementazione di una procedura di decisione per Binding-Fragments in Vampire

Relatore
Prof. Massimo Benerecetti

Correlatore
Prof. Fabio Mogavero

Candidato
Matteo Richard Gaudino

Matricola
N86003226

Anno Accademico 2022 - 2023

Indice

Introduzione	5
1 Prerequisiti	6
1.1 Logica Proposizionale	6
1.1.1 Formule	6
1.1.2 Assegnamenti	8
1.1.3 Forme Normali	8
1.1.4 Naming	8
1.2 Logica del primo ordine	8
1.2.1 Termini	8
1.2.2 Formule	8
1.2.3 Semantica	8
1.2.4 Forme Normali	8
1.2.5 Skolemizzazione	8
1.2.6 Unificazione	8
1.3 Soddisfacibilità e Validità	8
1.4 Resolution	8
1.5 Il formato TPTP	8
2 Binding-Fragments	9
2.1 Classificazione	9
2.2 Algoritmo Astratto	9
3 Vampire	10
3.1 I Termini	10
3.2 Formule e Clausole	10
3.3 Unificazione e Substitution Trees	10
3.4 Preprocessing	10
3.5 Saturazione e Refutazione	10
3.6 Il SAT-Solver	10
3.7 Misurazione dei Tempi	10

3.8	Opzioni	10
4	Progettazione	11
4.1	Algoritmo di Classificazione	11
4.2	Preprocessing	11
4.2.1	Boolean Top Formula	11
4.2.2	Forall-And	11
4.2.3	SAT-Clausification	11
4.3	Procedura di Decisione	11
4.3.1	Implicants Sorting	11
4.3.2	Maximal Unifiable Subsets	11
4.3.3	Algoritmo Finale	11
5	Implementazione	12
5.1	TODO	12
6	Sperimentazione	13
6.1	La libreria TPTP	13
6.2	Analisi dei risultati	13
6.3	Ottimizzazioni	13
6.4	Conclusioni e Possibili Sviluppi futuri	13

Introduzione

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Capitolo 1

Prerequisiti

In questo capitolo verranno descritte le nozioni di base necessarie per comprendere il lavoro svolto. In particolare, verranno introdotti i concetti di logica proposizionale e del primo ordine, definita come estensione della prima. Nell'ultimo paragrafo del capitolo verrà descritto in che modo le formule di logica del primo ordine possono essere rappresentate in un formato di file, per poi essere processate come input da un theorem prover.

1.1 Logica Proposizionale

1.1.1 Formule

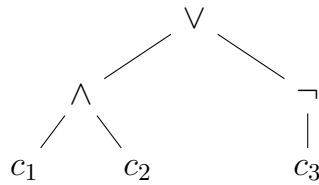
Sia $\Sigma_c = \{c_1, c_2, \dots\}$ un insieme di simboli di costante, $\Sigma = \{\wedge, \vee, \neg, (,), \top, \perp\} \cup \Sigma_c$ è detto alfabeto della logica proposizionale. Con queste premesse possiamo definire come formule della logica proposizionale il linguaggio generato dalla grammatica Context Free seguente:

$$\varphi := \top \mid \perp \mid C \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi)$$

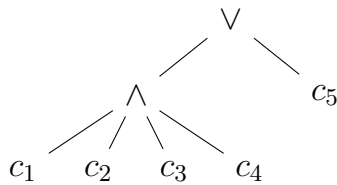
Dove $C \in \Sigma_c$ è un simbolo di costante. Vengono inoltre introdotti i seguenti simboli come abbreviazioni:

- $(\gamma \Rightarrow \kappa)$ per $(\neg\gamma \vee \kappa)$
- $(\gamma \Leftrightarrow \kappa)$ per $((\gamma \Rightarrow \kappa) \wedge (\kappa \Rightarrow \gamma))$
- $(\gamma \oplus \kappa)$ per $\neg(\gamma \Leftrightarrow \kappa)$

È possibile rappresentare una qualunque formula attraverso il proprio albero di derivazione. Questo albero verrà chiamato in seguito anche *albero sintattico* della formula. Ad esempio, la formula $(c_1 \wedge c_2) \vee \neg c_3$ può essere rappresentata dal seguente albero sintattico:



La radice dell'albero è detta *connettivo principale*. Per compattezza, grazie alla proprietà associativa di \wedge e \vee , è possibile omettere le parentesi, es. $(c_1 \wedge (c_2 \wedge (c_3 \wedge c_4))) \vee c_5$ può essere scritto come $(c_1 \wedge c_2 \wedge c_3 \wedge c_4) \vee c_5$. Allo stesso modo, nell'albero sintattico della formula è possibile compattare le catene di \wedge e \vee come figli di un unico nodo:



Questa è una caratteristica molto importante, in quanto non solo permette di risparmiare inchiostro, ma consente di vedere \wedge e \vee non più come operatori binari ma come operatori n-ari. A livello implementativo, ciò si traduce in un minor impatto in memoria, visite all'albero più veloci e algoritmi di manipolazione più semplici.

- 1.1.2 Assegnamenti
- 1.1.3 Forme Normali
- 1.1.4 Naming
- 1.2 Logica del primo ordine
 - 1.2.1 Termini
 - 1.2.2 Formule
 - 1.2.3 Semantica
 - 1.2.4 Forme Normali
 - 1.2.5 Skolemizzazione
 - 1.2.6 Unificazione
- 1.3 Soddisfacibilità e Validità
- 1.4 Resolution
- 1.5 Il formato TPTP

Capitolo 2

Binding-Fragments

2.1 Classificazione

2.2 Algoritmo Astratto

Capitolo 3

Vampire

3.1 I Termini

3.2 Formule e Clausole

3.3 Unificazione e Substitution Trees

3.4 Preprocessing

3.5 Saturazione e Refutazione

3.6 Il SAT-Solver

3.7 Misurazione dei Tempi

3.8 Opzioni

Capitolo 4

Progettazione

L'algoritmo di decisione, la classificazione, Il preprocessing

4.1 Algoritmo di Classificazione

4.2 Preprocessing

4.2.1 Boolean Top Formula

4.2.2 Forall-And

4.2.3 SAT-Clausification

4.3 Procedura di Decisione

4.3.1 Implicants Sorting

4.3.2 Maximal Unifiable Subsets

4.3.3 Algoritmo Finale

Capitolo 5

Implementazione

5.1 TODO

Capitolo 6

Sperimentazione

6.1 La libreria TPTP

6.2 Analisi dei risultati

6.3 Ottimizzazioni

6.4 Conclusioni e Possibili Sviluppi futuri