



UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Anton Galitch

Director:

Codirector:

Buenos Aires, 2014

Resumen

tes etst set set set set s

Palabras claves: test

Resumen

tes etst set set set set s hace falta esto?

Keywords: test

Índice general

1..	Introducción	1
1.1.	Motivación	1
1.2.	Objetivos Especificos	1
1.2.1.	Extensión de Heterogenius	1
1.2.2.	True Heterogeneity	1
1.2.3.	Ampliación del árbol de análisis	1
2..	Preliminares	3
3..	Aportes	5
3.1.	Extendiendo Heterogenius con Herramientas de Lógica de Primer Orden . .	5
3.1.1.	Herramientas Usadas	5
3.1.2.	Cálculo de secuentes	6
3.1.3.	Búsqueda de contraejemplos	6
3.1.4.	Integración con Heterogenius	6
3.2.	Extensión del concepto de heterogeneidad	7
3.2.1.	Operaciones para el manejo de fórmulas	7
3.2.2.	Extensión de las traducciones Rho	8
3.3.	Expansión del concepto de árbol de análisis	9
3.3.1.	Caminos alternativos en una demostración	9
4..	Caso de Estudio	11
5..	Conclusiones y Trabajos Futuros	13
6..	Referencias	15

1. INTRODUCCIÓN

1.1. Motivación

Analisis de software.
Heterogeneidad.
Tool support: lightweight y heavyweight.
Usabilidad e interfaces.

1.2. Objetivos Especificos

1.2.1. Extensión de Heterogenius

El lenguaje de lógica de primer orden TPTP-FOF, al ser muy difundido en la comunidad de investigadores de demostradores automáticos de teoremas es soportado por numerosas herramientas. Entre ellas EProver y SPASS, demostradores automáticos de teoremas; EProver y Mace4, buscadores de modelos.

Para permitir el uso de todas éstas herramientas y otras, se decidió integrar TPTP-FOF con Heterogenius mediante la implementación de una ρ -translation desde el lenguaje *PDOCFA*.

1.2.2. True Heterogeneity

Heterogenius en su primera versión permitió realizar demostraciones heterogeneas mediante traducciones de secuentes. Cada secuente tenía fórmulas de un mismo lenguaje haciendo que los secuentes sean en realidad homogeneos.

Se decidió ampliar el concepto de heterogeneidad expandiendolo también a los secuentes. De esta forma se permitió tener mayor flexibilidad en las demostraciones al poder soportar fórmulas de distintos lenguajes en el mismo secuente.

1.2.3. Ampliación del árbol de análisis

Con el objetivo de permitir documentar todo el proceso de demostración (camino alternativo tomados, decisiones que no produjeron ningún resultado exitoso, etc), se decidió ampliar el concepto de árbol de analisis. Para esto se incluyó un nuevo tipo de ramificación, que permite soportar ramas de demostraciones alternativas y ramas de decisiones no exitosas.

[TODO: tal vez algun grafico mostrando las diferentes ramas.]

2. PRELIMINARES

3. APORTES

3.1. Extendiendo Heterogenius con Herramientas de Lógica de Primer Orden

Existen numerosas herramientas que funcionan con el lenguaje de lógica de primer orden *TPTP-FOF*. Para permitir la integración de estas herramientas con Heterogenius y abrir el camino para la interacción con las futuras tecnologías basadas en éste lenguaje, se agregó *TPTP-FOF* al motor de Heterogenius como un lenguaje de análisis.

Junto a la integración de *TPTP-FOF*, se incorporaron los siguientes mecanismos para poder usar las herramientas correspondientes:

- Se permitió la carga de especificaciones escritas puramente en *TPTP-FOF* mediante la adaptación del *TPTP-Parser* [TODO: citar a Andrei Tchaltsev {tchaltsev AT itc.it} and Alexandre Riazanov {alexandre.riazanov AT gmail.com}.]
- Se agregó una ρ -translation desde las formulas *PDOCFA* a *TPTP-FOF*. [TODO: referenciar la seccion que explica esto en detalle].

Teniendo el soporte de *TPTP-FOF* por parte del motor de cálculo de secuentes de Heterogenius, integramos algunas de las herramientas mas difundidas en el ámbito de demostradores automáticos de teoremas para el lenguaje *TPTP-FOF*: *E-Prover* y *SPASS* como calculadores de secuentes; *E-Prover* y *Mace4* como buscadores de contraejemplos.

3.1.1. Herramientas Usadas

//TODO: revisar todo esto y explicar con mas detalle:

E-Prover

Es un demostrador automático de teoremas de lógica de primer orden basado en el calculo por superposición. Además realiza búsquedas de modelos por lo cual también se usa como un buscador de contraejemplos.

SPASS

Es un demostrador automático de teoremas de lógica de primer orden con igualdad desarrollado por el Instituto Max Planck.

Desde el 2000, tanto *SPASS* como *E-Prover* ocupan los primeros lugares en la competencia anual de demostradores de teoremas *CASC* (CADE ATP System Competition).

Mace4

Es un buscador de modelos finitos y contraejemplos para lógica de primer orden.

3.1.2. Cálculo de secuentes

TODO: ver como escribir bien esta parte:

Sea Σ la especificación y

$$\frac{\alpha_1, \dots, \alpha_n}{\beta_1, \dots, \beta_m}$$

el secuente que se quiere analizar.

Se arma una nueva fórmula $\varphi : \bigwedge_{i=1}^n \alpha_i \Rightarrow \bigvee_{j=1}^m \beta_j$.

Se aplica un demostrador automático para ver si $\Sigma \vdash \varphi$.

Dos reglas de calculo:

$$\frac{\Sigma \vdash \varphi}{true} \text{ (si vale)}$$

$$\frac{\Sigma \not\vdash \varphi}{false} \text{ (si no vale)}$$

TODO: otra posibilidad es que tire timeout. Hay que tener una regla para esto???

3.1.3. Búsqueda de contraejemplos

Tanto *Mace4* como *E-Prover* se usan para buscar contraejemplos de los secuentes *TPTP-FOF*. Como las dos herramientas son buscadores de modelos, lo que se hace es armar una teoría tomando en cuenta la especificación y el secuente que se quiere analizar.

Sea Σ la especificación y

$$\frac{\alpha_1, \dots, \alpha_n}{\beta_1, \dots, \beta_m}$$

el secuente que se quiere analizar.

Se arma una nueva teoría Γ tal que:

- $\varphi \in \Gamma$ si $\varphi \in \Sigma$.
- $\gamma \in \Gamma$
con $\gamma : \bigwedge_{i=1}^n \alpha_i \wedge \bigwedge_{j=1}^m \neg \beta_j$

Luego se realiza una búsqueda de un modelo para la teoría construida Γ . En caso de encontrar un modelo que satisfaga la teoría Γ se termina la búsqueda y se reporta que existe por lo menos un contraejemplo para el secuente procesado.

3.1.4. Integración con Heterogenius

TODO: explicar la arq. TODO: agregar algun diagrama.

3.2. Extensión del concepto de heterogeneidad

Otro de los objetivos fue extender el concepto de heterogeneidad para lograr tener demostraciones verdaderamente heterogeneas en lugar de demostraciones homogeneas en un árbol de análisis heterogeneo.

La diferencia principal radica en que con la implementación actual los secuentes pueden soportar fórmulas de diferentes lenguajes. Así un secuyente puede ser de tipo homoganeo o heterogeneo. En el primer caso todas las fórmulas del secuyente usan el mismo lenguaje; en el segundo las fórmulas son de lenguajes distintos.

La ventaja de los secuentes heterogeneos es que se puede combinar fórmulas (lemmas, propiedades, teoremas) provenientes de distintas especificaciones escritas en lenguajes diferentes. De éste forma nos podemos abstraer del lenguaje en el que están escritas las fórmulas y concentrarnos en el análisis.

La principal limitación de los secuentes heterogeneos es que las herramientas (calculadores de secuentes, buscadores de contraejemplos, demostradores automáticos) trabajan con secuentes escritos en un solo lenguaje, o sea secuentes homogeneos. Debido a ésto se proveen nuevas operaciones para el manejo de fórmulas dentro de un secuyente:

3.2.1. Operaciones para el manejo de fórmulas

Cada una de las siguientes operaciones puede cambiar o no la heterogeneidad de un secuyente. Dependiendo de los lenguajes de las fórmulas del resultado, el secuyente puede pasar a ser heterogeneo, homoganeo o mantener su tipo.

Proyección

Dado un secuyente, se selecciona un subconjunto de las fórmulas que se quieren proyectar y el nuevo secuyente se forma a partir de las fórmulas seleccionadas.

Dado un secuyente S

$$\frac{\alpha_1, \dots, \alpha_n}{\alpha_{n+1}, \dots, \alpha_m}$$

y un subconjunto $\mathcal{C} \subseteq \{1 \dots m\}$, el secuyente resultante S' :

$$\frac{\alpha_i \text{ con } i = 1 \dots n \text{ y } i \in \mathcal{C}}{\alpha_j \text{ con } j = n + 1 \dots m \text{ y } j \in \mathcal{C}}$$

Introducción de antecedentes desde una fuente externa

Ésta operación permite cargar desde un archivo de especificación, ya sea *Alloy* o *FOF* axiomas e introducirlos como antecedentes del secuyente analizado.

Dado un secuyente S

$$\frac{\alpha_1, \dots, \alpha_n}{\alpha_{n+1}, \dots, \alpha_m}$$

y un conjunto de fórmulas nuevas $\{\beta_1 \dots \beta_k\}$. El nuevo secuyente es:

$$\frac{\alpha_1, \dots, \alpha_n, \beta_1, \dots, \beta_k}{\alpha_{n+1}, \dots, \alpha_m}$$

Traducción

Se extendió el concepto de traducciones ρ para que se puedan traducir fórmulas por separado. El seciente resultante contendrá las fórmulas del seciente analizado en el lenguaje seleccionado.

Dado un seciente S

$$\frac{\alpha_1^{l_1}, \dots, \alpha_n^{l_n}}{\alpha_{n+1}^{l_{n+1}}, \dots, \alpha_m^{l_m}}$$

con $\alpha_i^{l_i}$ fórmula en el lenguaje l_i .

y una relación $\mathcal{T} : Formula \times Lenguaje$ que indica el lenguaje seleccionado para cada fórmula del seciente S , el seciente resultante S' es:

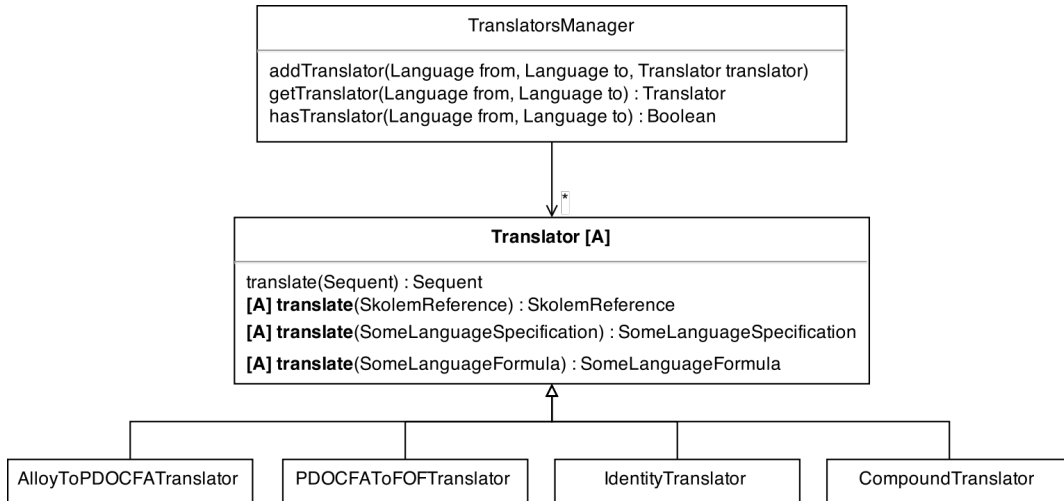
$$\frac{\beta_1^{l_1}, \dots, \beta_n^{l_n}}{\beta_{n+1}^{l_{n+1}}, \dots, \beta_m^{l_m}}$$

con $\beta_i^{l_i} = \alpha_i^{\mathcal{T}(\alpha_i, l_i)}$.

3.2.2. Extensión de las traducciones Rho

Debido a todos éstos cambios fue necesario refactorizar el diseño de la infraestructura que soportaba las traducciones ρ . Lo primero que se hizo fue tener un *TranslationsManager*, un objeto encargado de manejar todas las traducciones soportadas por el sistema.

Por otro lado los traductores (subclases de *Translator*) deben implementar los tres métodos abstractos definidos en la clase padre. Cada uno de éstos métodos permite un control más fino de las traducciones al separar el seciente en sus partes, que son: una referencia de skolemización, una especificación y la fórmula analizada.



Se proveen los traductores de *Alloy* a *PDOCFa*, de *PDOCFa* a *TPTP-FOF* así como el *CompoundTranslator* que permite componer los traductores para lograr traducciones transitivas, por ejemplo de *Alloy* a *TPTP-FOF*.

3.3. Expansión del concepto de árbol de análisis

3.3.1. Caminos alternativos en una demostración

4. CASO DE ESTUDIO

5. CONCLUSIONES Y TRABAJOS FUTUROS

.....

6. REFERENCIAS

