Mejora en la eficiencia de la demostración automática de teoremas en lógica temporal usando implicantes e implicados.

P. Cordero, M. Enciso, I. P. de Guzmán

E.T.S.I. Informática. Universidad de Málaga.
Campus de Teatinos. 29071 Málaga, Spain
{pcordero,enciso}@uma.es guzman@ctima.uma.es

Resumen

Una simple mirada a la bibliografía, nos permite comprobar el gran interés que existe en la actualidad en el área de la demostración automática en lógica temporal (ver, como ejemplo, las recientes publicaciones [3], [4], [16], [17], [18]).

En la mayoría de estos trabajos se desarrollan métodos de deducción automática clausales y, entre aquellos que son no clausales, no se ha conseguido hasta ahora introducir las ventajas en eficiencia que se consiguen con el aprovechamiento de la información aportada por los implicantes e implicados. En nuestra opinión el modo de abordar el trabajo sin un referente único en la metateoría, es la causa de que esta técnica no haya podido ser aplicada aún en la Lógica Temporal.

Nuestro planteamiento ha sido desarrollar el estudio teórico de los conjuntos de implicantes e implicados en Lógica Temporal. Dicho trabajo, inédito hasta el momento, servirá de base para introducir mejoras en la eficiencia de los métodos de deducción temporal de cualquier tipo.

Palabras Claves: Logica Temporal, Demostradores de Teoremas, Teoría de la Computación.

1 Introducción

El interés de la lógica temporal en Computación reclama seguir realizando aportaciones hasta alcanzar el papel que le corresponde para incorporar capacidad deductiva en tareas frente a las que la lógica clásica hace patente sus deficiencias, tareas en las que los aspectos temporales son esenciales. Hoy en día está ampliamente aceptado que la dificultad de cálculo con las fórmulas temporales es una de las causas principales que entorpece la consecución del objetivo de que la lógica temporal juegue el papel que le corresponde en las Ciencias de la Computación. La riqueza del desarrollo de la metateoría para la lógica temporal, no se corresponde con los resultados que conducen a su efectiva aplicación.

En definitiva, este trabajo se enmarca en el área de la demostración automática de teoremas para lógicas temporales, siguiendo un nuevo paradigma de demostración denominado TAS e introducido por nuestro grupo de investigación.[1] El trabajo pionero fue la tesis doctoral de D. Francisco Sanz [15]; desde entonces, los métodos TAS desarrollados para la lógica clásica [8], para lógicas multivaluadas [9, 7] y lógicas temporales [8, 11, 9], han ido confirmando uno de los objetivos de la nueva metodología: su flexibilidad para extenderse a las lógicas no clásicas.

Los métodos TAS son métodos por refutación cuyas ventajas principales con respecto a otros métodos son las siguientes:

- A diferencia de los métodos de resolución [3], son no clausales, es decir, no requieren la transformación a una forma clausal de la fórmula a analizar. Esta característica era obligada con el objetivo final de obtener métodos TAS en lógicas no clásicas para las que no existen formas clausales adecuadas.

  Por contra, dicha exigencia de los métodos de Resolución les hace no tener competencia en comparación con un método tipo Gentzen a la hora de su ejecución. En este sentido, el método TAS puede entenderse como una nueva vía en deducción automática (como afirma R. Hahnler en [13, 14]) o como un conjunto de estrategias eficientes que pueden ser incorporadas a cualquier método de deducción auto-

[1] Su nombre se debe a que los métodos TAS hasta aquí desarrollados trabajan mediante Transformaciones de Árboles Sintácticos de la fórmula cuya validez se desea analizar.

mática para mejorar su eficiencia (como se ha hecho en [10, 11] con los métodos basados en Gentzen).

- Mientras que los métodos de tablas semánticas y de disolución [12, 19, 4] se basan en la busqueda de ramas insatisfacibles en el proceso de ramificación (encargado de la distribución de $\wedge$ con respecto a $\vee$ y del avance temporal, responsable del coste exponencial en el peor caso de los demostradores automáticos de teoremas), TAS utiliza técnicas dirigidas de modo dinámico por la fórmula sobre las que ejecutan para evitar distribuciones.

Básicamente, *TAS determina y manipula eficientemente conjuntos de implicados e implicantes unitarios de la fórmula de entrada.* Esta manipulación nos permite sustituir fórmulas por constantes (incorporando las estrategias utilizadas por los métodos de tablas semánticas y disolución) y eliminar, total o parcialmente, ocurrencias de un átomo en una fórmula. Estas transformaciones, llamadas *reducciones*, juegan un papel destacado en el paradigma TAS y su importancia radica en que permiten disminuir el tamaño de la fórmula de entrada *antes* de la ramificación.

El diseño de las *reducciones* y, a posteriori, del demostrador automático para una lógica dada, requiere cuatro etapas bien definidas:

- La elección adecuada de las fórmulas unitarias o literales.
- El anális del conjunto de literales con la relación de orden "implicación lógica", que permita definir subconjuntos con las propiedades máxima información–mínimo coste en el estudio de la satisfacibilidad en la lógica objetivo. En definitiva, la teoría formal de implicantes e implicados unitarios y el diseño de algoritmos que permitan su manipulación eficiente.
- La definición de una forma normal que mantenga las "buenas" propiedades de la *forma normal negativa* en la lógica clásica proposicional y de un algoritmo eficiente que permita la transformación de toda fórmula en un tal forma normal equivalente.
- La definición de *reducciones* que preservan el significado y de *reducciones* que preservan la satisfacibilidad.

El punto de partida más cercano de este trabajo es [11], donde los autores presentan un demostrador automático de teoremas para la lógica temporal lineal con tiempo discreto e infinito (lógica FNext). En dicho trabajo, el tratamiento de implicantes e implicados no gozaba de la propiedad antes citada de disponer de *la máxima información con el mínimo coste.*

En este trabajo hemos buscado la consecución de dicho objetivo, extendiendo además la lógica original con las correspondientes conectivas de pasado, dando lugar a la lógica FNext$\pm$. Por otra parte, el estudio de FNext facilita la comprensión del estudio posterior para FNext$\pm$, de mayor dificultad, por lo que ha sido necesario desarrollar este trabajo, en una primera etapa para la lógica FNext, cubriendo en una segunda fase la más compleja tarea correspondiente a la lógica FNext$\pm$.

Concretamente, nuestro trabajo cubre los tres primeros puntos arriba mencionados siguiendo el siguiente esquema: En la sección 3, se presenta el conjunto de literales para la lógica FNext$\pm$. A continuación, se presentan los conceptos de conjunto cerrado de literales, conjunto generador de conjuntos cerrados y bases de dichos conjuntos. Estas bases son los conjuntos que disponen de la propiedad deseada (máxima información - mínimo coste). Para la aplicación a la demostración automática de teoremas en general, y a la metodología TAS en particular, de estos resultados, es necesaria además la introducción de la definición de una forma normal negativa temporal que permita disponer de la información sobre implicantes e implicados de modo eficiente.

En este trabajo, hemos obviado una exposición detallada de las definiciones intermedias y hemos omitido las pruebas formales de los resultados presentados. Dicho contenido puede ser consultado en [6] para FNext y en [5] para FNext$\pm$.

2 La lógica FNext$\pm$

FNext$\pm$ es la Lógica Temporal Proposicional con flujo de tiempo lineal, discreto e infinito y las conectivas $\neg$ (negación), $\wedge$ (conjunción), $\vee$ (disyunción), $\rightarrow$ (implicación material), F (alguna vez en el futuro), G (siempre en el futuro), $\oplus$ (mañana), P (alguna vez en el pasado), H (siempre en el pasado), $\ominus$ (ayer) y los símbolos $\top$ (verdad) y $\bot$ (falsedad).

$\mathcal{V}$ denota el conjunto de variables proposicionales $p, q, r, \ldots$ (posiblemente con subíndices). Asumimos que este conjunto está completamente ordenado lexicográficamente, es decir, $p_n \leq q_m$ para todo n, m y $p_n \leq p_m$ si y sólo si $n \leq m$.

Definición 2.1 *Las fórmulas bien formadas en FNext$\pm$ (fbfs) son las generadas por las reglas de construcción de la lógica clásica proposicional junto con la siguiente regla:*

si A es una fórmula bien formada (fbf), $\oplus A$, FA, GA, $\ominus A$, PA y HA son fbfs .

3 Literales Temporales

En esta sección definimos el conjunto de fórmulas unitarias o *literales* y un orden parcial en él.

Definición 3.1 *Definimos una relación binaria, $\trianglelefteq$, del siguiente modo: Si A y B son fbfs ,*

$$A \trianglelefteq B \text{ si y sólo si } \models A \to B$$

Y definimos una relación binaria de equivalencia, $\equiv$, del siguiente modo: Si A y B son fbfs ,

$$A \equiv B \text{ si y sólo si } A \trianglelefteq B \text{ y } B \trianglelefteq A$$

Obsérvese que $\trianglelefteq$ no es una relación de orden. Sin embargo, la relación inducida por $\trianglelefteq$ en el conjunto de clases de equivalencia $FNext \pm /_{\equiv}$, tambien denotada por $\trianglelefteq$, es un orden parcial.

Definición 3.2 *Dado $p \in \mathcal{V}$, las fbfs p y $\neg p$ recibirán el nombre de literales clásicos en p. En el resto del trabajo, ℓ_p denotará a un literal clásico en p y $\mathcal{V}^{\pm}$ al conjunto de literales clásicos.*

Sea FNext$\pm^{mon}$ el conjuto de *fbfs* que no tienen conectivas binarias, es decir, *fbfs* de la forma $A = \gamma_1 \dots \gamma_n \ell_p$ donde $\ell_p \in \mathcal{V}^{\pm}$ y, para todo $1 \leq i \leq n$, se tiene que $\gamma_i \in \{F, G, \oplus, P, H, \ominus, \neg\}$.

El par $(FNext \pm^{mon} /_{\equiv}, \trianglelefteq)$ es un conjunto parcialmente ordenado (**poset**[2]).

Presentamos, a continuación, las leyes que nos permiten seleccionar un representante canónico de cada clase en FNext$\pm^{mon}/_{\equiv}$.

- $\neg\neg A \equiv A$, $\neg \oplus A \equiv \oplus\neg A$, $\neg FA \equiv G\neg A$, $\neg GA \equiv F\neg A$, $\neg \ominus A \equiv \ominus\neg A$, $\neg PA \equiv H\neg A$, $\neg HA \equiv P\neg A$.
- $FFA \equiv F \oplus A$; $\quad GGA \equiv G \oplus A$; $PPA \equiv P \ominus A$; $\quad HHA \equiv H \ominus A$.
- $HFA \equiv F \ominus A$; $\quad PGA \equiv G \ominus A$; $GPA \equiv P \oplus A$; $\quad FHA \equiv H \oplus A$.
- $\oplus \ominus A \equiv \ominus \oplus A \equiv A$; $\quad HGA \equiv GHA$; $\quad PFA \equiv FPA$
- Si $\gamma \in \{F, G, P, H\}$ entonces $\oplus\gamma A \equiv \gamma \oplus A$ y $\ominus\gamma A \equiv \gamma \ominus A$.
- Si $\gamma \in \{FG, GF, PH, HP, FP, GH\}$ entonces $\gamma \oplus A \equiv \gamma A$, $\gamma \ominus A \equiv \gamma A$, $F\gamma A \equiv \gamma A$; $G\gamma A \equiv \gamma A$; $P\gamma A \equiv \gamma A$ y $H\gamma A \equiv \gamma A$.

Ejemplo 1.- En este ejemplo, obtenemos el representante canónico de la *fbf* $A = \neg P \oplus GF \oplus \oplus\neg G\neg \ominus p$ en $FNext\pm$.

$$\neg P \oplus GF \oplus \oplus\neg G\neg \ominus p \equiv HFGG \oplus \oplus \oplus \ominus\neg p \equiv FG \oplus \oplus\neg p \equiv FG\neg p$$

[2]Partially ordered sets.

Los elementos de $FNext \pm^{mon} /_{\equiv}$ recibirán el nombre de literales temporales y serán representados por su forma canónica. En la siguiente definición se describen estos literales:

Definición 3.3 *Dado un literal clásico $\ell_p \in \mathcal{V}^{\pm}$, el conjunto de* literales temporales [3] *en ℓ_p, denotado $Lit(\ell_p)$, es:*

$$\begin{aligned} Lit(\ell_p) = \{\top, \bot\} \cup \\ \cup\{FG\ell_p, GF\ell_p, PH\ell_p, HP\ell_p, FP\ell_p, GH\ell_p\} \\ \cup\{\odot^k \ell_p, F \odot^k \ell_p, G \odot^k \ell_p, P \odot^k \ell_p, H \odot^k \ell_p \mid k \in \mathbb{Z}\} \end{aligned}$$

donde entendemos $\odot^k \ell_p$ como: $\oplus \overset{k}{\dots} \oplus \ell_p$ si $k > 0$, ℓ_p si $k = 0$ y $\ominus \overset{k}{\dots} \ominus \ell_p$ si $k < 0$.

Denotaremos por Lit al conjunto de todos los literales temporales en la lógica en la que estemos trabajando:

$$Lit = \bigcup_{\ell_p \in \mathcal{V}^{\pm}} Lit(\ell_p)$$

El conjunto de literales en FNext es la restricción al fragmento de futuro de este conjunto:

$$\begin{aligned} Lit(\ell_p) = \{\top, \bot\} \cup \{FG\ell_p, GF\ell_p\} \cup \\ \cup\{\oplus^k \ell_p, F \oplus^k \ell_p, G \oplus^k \ell_p \mid k \in \mathbb{N}\} \end{aligned}$$

El par $(Lit, \trianglelefteq)$ en FNext es un retículo. Sin embargo, en FNext$\pm$, $(Lit, \trianglelefteq)$ no lo es, ya que existen pares de literales que no tienen supremo o ínfimo. Por ejemplo, el conjunto de cotas superiores de $\{\ell_p, \oplus\ell_p\}$ tiene dos elementos minimales, $P \oplus^2 \ell_p$ y $F \ominus \ell_p$. En la figura 1 mostramos el retículo $(Lit(\ell_p), \trianglelefteq)$ de FNext y, en la figura 2, el poset $(Lit(p), \trianglelefteq)$ de FNext$\pm$.

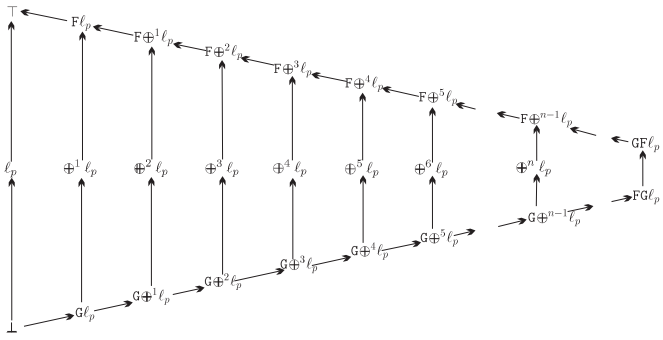

Figura 1: Retículo $Lit(\ell_p)$ en FNext

[3]Como sólo trabajaremos con literales temporales, en el resto del trabajo omitiremos el adjetivo *temporal*.