Aplicación de la Lógica Temporal Proposicional para la Representación y Deducción de Conocimientos Temporales

Bettina Kemme
Universidad de Erlangen
e-mail: kemme@algebra.us.es
bakemme@cip.informatik.uni-erlangen.de
Joaquin Borrego Díaz
Universidad de Sevilla
e-mail: jborrego@obelix.cica.es

In Computer Science, propositional temporal logics are mainly used for the verification of concurrent systems. Formulae are interpreted over state sequences, where each state is labeled with a set of propositions. A state sequence can be regarded as a discrete passage of time and the temporal logic provides the formalization and deduction of temporal properties.
On the other hand, in artificial intelligence research temporal knowledge is often represented by relations between intervals, and reasoning about incomplete information is a main task.
Our aim is to use the mathematical and technical machinery of temporal logics to reason about temporal intervals. Therefore we present how state sequences can represent relations between intervals and how properties of the intervals can be expressed by temporal formulae. We use decision procedures and model-checker algorithms to find scenarios that are consistent with the information provided, and for queries about given knowledge.

1 Introducción

Las lógicas temporales proposicionales (LTP) facilitan la formalización del concepto *tiempo* y permiten deducir sobre él. Se interpretan las fórmulas sobre sucesiones de estados etiquetados por conjuntos de proposiciones. Las sucesiones representan transcursos del tiempo y las fórmulas ofrecen la formalización de propiedades de este transcurso. Con algoritmos de satisfactibilidad se pueden construir efectivamente un modelo de una fórmula y se tienen algoritmos de *model-checking* para verificar eficientemente si las sucesiones de un grafo de transiciones son modelos de una fórmula. Las LTP desempeñan un gran papel en la verificación de programas paralelos.

En Inteligencia Artificial (IA), también se representan conocimientos temporales por relaciones temporales entre intervalos, que describen acciones o sucesos. Muy importante es la deducción sobre conocimientos incompletos.

En este trabajo nos interesamos en expresar los conceptos del tiempo desarrollados en la IA con las LTP. En la sección 2 presentamos la lógica temporal lineal considerada (LTL), sus modelos y los algoritmos de satisfactibilidad y model-checking. En la sección 3 exponemos la representación de intervalos temporales y sus relaciones por la lógica de intervalos. En la sección 4 representamos relaciones entre intervalos por los modelos de la LTL y traducimos fórmulas de la sección 3 a la LTL. Aplicamos el algoritmo de la construcción de un modelo para encontrar un o todos los modelos de fórmulas de la lógica de intervalos y describimos como se puede contestar a consultas (sección 5). Al final damos en sección 6 una perspectiva a las próximas investigaciones.

2 Lógica Temporal Proposicional y Lineal

2.1 Interpretación

La lógica temporal lineal (LTL) considera el tiempo como un único carril de tiempo, en el que cada instante tiene un instante siguiente inequívocamente determinado. La interpretación de una fórmula temporal lineal debe de representar adecuadamente este transcurso lineal del tiempo.
Formalmente una interpretación sobre un conjunto de variables proposicionales VP es una *estructura lineal* $\mathcal{M} = (S, \mathcal{I}, x)$, en la cual S representa un conjunto de estados, $\mathcal{I}$ asocia a cada estado s un subconjunto de VP, las variables verdaderas en s, y $x = (s_0, s_1, \ldots)$, o sea $x = (s_0, s_1, \ldots, s_m)$ en el caso finito, es una sucesión de estados que representa el transcurso del tiempo.

2.2 Sintaxis

Partiendo del conjunto de variables proposicionales VP, se construyen las fórmulas empleando las proposiciones y los operadores lógicos y temporales como sigue:

$$A := (p) | \neg A_1 | (A_1 \wedge A_2) | \bigcirc A_1 | \bigcirc_\forall A_1 | (A_1 \mathcal{U} A_2)$$

Se usa las abreviaturas usuales: $(A \vee B)$ para $\neg(\neg A \wedge \neg B)$ y $(A \rightarrow B)$ para $(\neg A \vee B)$, $\Diamond A$ para $(true\mathcal{U}A)$, $\Box A$ para $\neg\Diamond\neg A$ o $(A\mathcal{W}false)$ y $(A\mathcal{W}B)$ para $(\Box A \vee (A\mathcal{U}B))$.

2.3 Semántica

Sea $\mathcal{M} = (S, \mathcal{I}, x)$ una interpretación. Sea $x^i = (s_i, s_{i+1}, \ldots)$ (o sea $x^i = (s_i, \ldots, s_m)$, si $|\mathcal{D}_x| = m$) sufijo de x, comenzando en s_i. El valor de verdad de una fórmula A de la LTL en una interpretación $\mathcal{M}$ respeto de la sucesión x, que escribiremos $\mathcal{M}, x \models A$, se define recursivamente de la siguiente manera:

- $\mathcal{M}, x \models p$ ssi p variable y $p \in \mathcal{I}(s_o)$
- $\mathcal{M}, x \models \neg A$ ssi $\mathcal{M}, x \not\models A$
- $\mathcal{M}, x \models (A \wedge B)$ ssi $\mathcal{M}, x \models A$ y $\mathcal{M}, x \models B$
- $\mathcal{M}, x \models \bigcirc A$ ssi $\mathcal{M}, x^1 \models A$
- $\mathcal{M}, x \models \bigcirc_\forall A$ ssi si s_0 tiene un instante siguiente s_1, entonces $\mathcal{M}, x^1 \models A$
- $\mathcal{M}, x \models (A\mathcal{U}B)$ ssi $\exists i : \mathcal{M}, x^i \models B$ y $\forall k < i : \mathcal{M}, x^k \models A$

2.4 Algoritmos de satisfactibilidad y construcción de un modelo

Con el algoritmo de satisfactibilidad de [2] podemos construir fácilmente un modelo de una fórmula A en el caso de que A tiene un modelo finito. El algoritmo se base en el método del tablero, conocido de la lógica proposicional. Se construye un tablero descomponiendo la fórmula A en pasos sucesivos. El tablero consiste en un grafo dirigido con una raíz, en que los nodos tienen conjuntos de subfórmulas de A como etiquetas. La ampliación referente a la lógica proposicional consiste en el tratamiento adecuado de los operadores temporales. En el tablero hay nodos especiales que representan los estados del modelo.
Si la fórmula A tiene un modelo lineal finito el tablero tiene una hoja *abierta* que está etiqueteado sólamente con literales no contradictorios y fórmulas del tipo $\bigcirc_\forall A$.
Obtenemos el modelo transformando todos los nodos especiales del camino desde la raíz hasta la hoja y la hoja a estados.

Podemos considerar un grafo finito de transiciones $\mathcal{G}$, cuyos estados están etiqueteados con conjuntos de proposiciones, como un conjunto de estructuras lineales finitas. El algoritmo de model-checking descrito en [8] comprueba si hay en $\mathcal{G}$ una sucesión que es modelo de una fórmula dada.Otros algoritmos de este tipo se pueden encontrar en [3, 4, 6, 7, 10, 12].

3 La Lógica de Intervalos

La forma más habitual de representar conocimientos temporales dentro de la inteligencia artificial es la lógica temporal de J. F. Allen.
En [1] Allen utiliza intervalos densos para construir una estructura temporal. Representa sucesos y acciones por intervalos y relaciona estos intervalos con 13 relaciones básicas (vea tabla 1).

Relación	**Símb.**	**Símb. inv.**	**Ejemplo**
X igual que Y	=	=	XXX YYY
X antes de Y	<	>	XXX YYY
X da con Y	m	mi	XXXYYY
X superp. Y	o	oi	XXX YYY
X dentro de Y	d	di	XXX YYYYYY
X arranca Y	s	si	XXX YYYYY
X termina Y	f	fi	 YYYYY

Tabla 1 Relaciones temporales básicas de Allen

Si no sabemos la relación exacta entre dos intervalos, es decir si el conocimiento sobre los intervalos es *incompleto*, no podemos describirlo con una de las relaciones básicas.
Sin embargo podemos expresar cada relación posible por una disjunción sobre un subconjunto de las relaciones básicas.

Definición 3.1 *Llamamos lógica de intervalos a la lógica del primer orden sobre los símbolos de relaciones binarias* $\{=, <, >, m, mi, o, oi, f, fi, s, si, d, di\}$ *y un conjunto de variables proposicionales de intervalos.*
Llamamos fórmulas de intervalos a las fórmulas del la lógica de intervalos si están en forma normal conjuntiva sin negación y en cada disjunción se encuentran exactamente dos proposiciones de intervalos.
Una fórmula de intervalos es completa, si tiene una disjunción para cada dos proposiciones de intervalos. De lo contrario es incompleta.

Ejemplo $(AdiB \wedge B < C \wedge (A < C \vee AmC)$ es una fórmula de intervalo completa.

Un *escenario consistente* es una disposición temporal de los intervalos de una fórmula, que ilustra una de las relaciones posibles expresadas por esta fórmula.

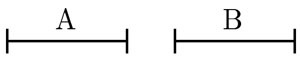

Figura 1 Escenario consistente para $A < B$

Allen indica reglas de inferencia, que podemos considerar como reglas de transitivitad sobre el conjunto de relaciones. Estas reglas posibilitan la deducción temporal.

Ejemplo Sean A, B y C tres intervalos y valga, que A superpone B y B superpone C, entonces A está antes de C o superpone C o da con C:

$$AoB \wedge BoC \Longrightarrow A < C \vee AoB \vee AmB.$$

Véase la tabla de todas las reglas de transitividad en [1].

Los problemas de encontrar un escenario consistente y encontrar todas las relaciones posibles entre dos intervalos son problemas $\mathcal{NP}$-completos.

Otra forma de representar relaciones entre intervalos consiste en indicar las posiciones temporales de los puntos iniciales y finales de los intervalos (véase [14]). Sean a^+ punto final del intervalo A

y b^- punto inicial de B entonces $(a^+ < b^-)$ representa la relación básica $A < B$. Relaciones posibles son $\{<, \leq, =, >, \geq, \neq\}$. La lógica correspondiente se llama *Algebra de puntos*.

4 Representación de la Lógica de Intervalos por la LTL

¿Cómo podemos representar los intervalos y sus relaciones con la LTL? Los intervalos temporales de Allen son continuos. Para poder expresar esta propiedad con la representación discreta de la estructura $\mathcal{M} = (S, \mathcal{I}, x)$ nos imaginamos cada instante discreto como un periodo continuo que, de momento, no podemos descomponer, es decir, que no tenemos información sobre la estructura interna de este periodo. Si representamos cada intervalo por una proposición entonces el conjunto de instantes (periodos) en que esa proposición es cierto representa la duración total del intervalo.

Definición 4.1 *Llamamos un* a-intervalo *a una subsucesión finita* $x^{m-n} = (s_m, s_{m+1}, \ldots, s_n)$ *de una estructura lineal* $\mathcal{M} = (S, \mathcal{I}, x)$, *si*
$\forall i, m \leq i \leq n:$
$a \in \mathcal{I}(s_i) \wedge a \notin \mathcal{I}(s_{m-1}) \wedge a \notin \mathcal{I}(s_{n+1})$.

Definición 4.2 *Una estructura lineal* $\mathcal{M} = (S, \mathcal{I}, x)$ *es una* interpretación de intervalos, *si es finita, es decir,* $\mathcal{D}_x$ *finito, y cada a-intervalo en* $\mathcal{M}$ *es único.*

Lema 4.3 *Para cada fórmula de intervalos que es satisfactible existe una interpretación de intervalos, que es un modelo para esta fórmula.*

Demostración Si una fórmula de intervalos es satisfactible entonces hay un escenario consistente. Podemos transformar este escenario a un modelo. En cada instante en que tiene lugar en el escenario un cambio de los intervalos válidos pasamos en la interpretación de intervalos a un nuevo estado, en que son ciertas exactamente aquellas proposiciones, cuyos intervalos son ciertos en el escenario después del cambio.

Definición 4.4 *Una interpretación de intervalos es un modelo mínimo de intervalos* $\mathcal{M}_{min}$ *de una fórmula de intervalos, si representa un escenario consistente y no hay ningun modelo de intervalos* $\mathcal{M}$, *que represente el mismo escenario consistente con* $|\mathcal{D}_x| < |\mathcal{D}_{x_{min}}|$.

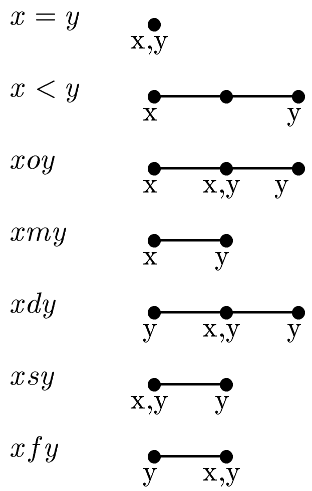

Figura 2 Modelos mínimos de intervalos de las relaciones básicas

Interpretaciones de intervalos son un subconjunto de las interpretaciones de la LTL y nos preguntamos si podemos traducir fórmulas de intervalos a fórmulas de la LTL. Con la lógica temporal que presentamos en 2 sólo podemos hablar del futuro. Pero como para cada relación básica que define un intervalo o una parte de un intervalo *antes* de otro intervalo existe una relación inversa, que expresa la relación en forma futura y con las proposiciones de intervalos cambiados, no parece necesario ampliar la lógica con operadores temporales para que podamos hablar del pasado.

En tabla 2 proponemos para cada relación básica una fórmula de la LTL, llamada *fórmula básica*, que tiene como modelo el modelo mínimo de esa relación. Se abrevia $(x \wedge y)$ por xy, $(\neg x \wedge y)$ por $\neg x$y, $(x \wedge \neg y)$ por $x \neg y$ y $(\neg x \wedge \neg y)$ por $\neg x \neg y$.

X antes Y: $(\neg x\neg y\mathcal{U}(x\neg y\wedge((x\neg y\mathcal{U}\neg x\neg y)\mathcal{U}\neg xy)))$
X igual Y: $(\neg x\neg y\mathcal{U}(xy\wedge(xy\mathcal{W}\neg x\neg y)))$
X da con Y: $(\neg x\neg y\mathcal{U}(x\neg y\wedge(x\neg y\mathcal{U}\neg xy)))$
X superpone Y: $(\neg x\neg y\mathcal{U}(x\neg y\wedge(x\neg y\mathcal{U}(xy\wedge(xy\mathcal{U}\neg xy)))))$
X dentro de Y: $(\neg x\neg y\mathcal{U}(\neg xy\wedge(\neg xy\mathcal{U}(xy\wedge(xy\mathcal{U}\neg xy)))))$
X arranca Y: $(\neg x\neg y\mathcal{U}(xy\wedge(xy\mathcal{U}\neg xy)))$
X termina Y: $(\neg x\neg y\mathcal{U}(\neg xy\wedge(\neg xy\mathcal{U}(xy\wedge(xy\mathcal{W}\neg x\neg y)))))$

Tabla 2 Las relaciones básicas de Allen como fórmulas de la LTL

Lema 4.5 *Sea ArB una fórmula de intervalos que describe una relación básica. Sea F_r la fórmula básica correspondiente de la LTL.*
Una interpretación de intervalos $\mathcal{M}=(S,\mathcal{I},x)$ es un modelo de F_r ssi representa un escenario consistente para ArB.

DEMOSTRACIÓN Realizamos la demostración sólo para la relación *superpone*. Las demostraciones para las fórmulas demás son análogas.

Sea $\mathcal{M}=(S,\mathcal{I},x)$ una interpretación de intervalos con $x=(s_0,\ldots,s_m)$ y $F_o\equiv(\neg a\neg b\mathcal{U}(a\neg b\wedge(a\neg b\mathcal{U}(ab\wedge(ab\mathcal{U}\neg ab)))))$.
$\Longrightarrow$:
De $\mathcal{M},x\models F_o$ resulta, que
$\exists k_1:\mathcal{M},x^{k_1}\models(a\neg b\mathcal{U}(ab\wedge(ab\mathcal{U}\neg ab)))$
$\wedge a\in\mathcal{I}(s_{k_1})\wedge b\notin\mathcal{I}(s_{k_1})\ \wedge\forall i,i<k_1:a,b\notin\mathcal{I}(s_i)$.
Esto significa, que
$\exists k_2>k_1:\mathcal{M},x^{k_2}\models(ab\mathcal{U}\neg ab)\ \wedge a,b\in\mathcal{I}(s_{k_2})$
$\wedge\forall i,k_1\leq i<k_2:a\in\mathcal{I}(s_i)\wedge b\notin\mathcal{I}(s_i)$.
Así resulta, que
$\exists k_3>k_2:a\notin\mathcal{I}(s_{k_3})\wedge b\in\mathcal{I}(s_{k_3})$
$\wedge\forall i,k_2\leq i<k_3:a,b\in\mathcal{I}(s_i)$.
De esto resulta, que
$x=(s_0,\ldots s_{k_1},\ldots,s_{k_2},\ldots s_{k_3},\ldots,s_m)$,
$0\leq k_1<k_2<k_3\leq m$, con
$\forall i,i<k_1:a,b\notin\mathcal{I}(s_i)\wedge$
$\forall i,k_1\leq i<k_2:a\in\mathcal{I}(s_i)\wedge b\notin\mathcal{I}(s_i)\wedge$
$\forall i,k_2\leq i<k_3:a,b\in\mathcal{I}(s_i)\wedge$
$a\notin\mathcal{I}(k_3)\wedge b\in\mathcal{I}(s_{k_3})$.
El hecho que $\mathcal{M}$ es una interpretación de intervalos garantiza que
$\exists j,k_3\leq j\leq m:\quad\forall k\leq j:a\notin\mathcal{I}(s_k)\wedge b\in\mathcal{I}(s_k)$ y
$\forall k>j:a,b\notin\mathcal{I}(s_k)$,
porque cada a-intervalo es único.
En particular es cierto que en efecto existen un a- y un b-intervalo en $\mathcal{M}$.
$\Longleftarrow$:
Representamos un escenario consistente de AoB por una estructura lineal, cuyo sucesión de estados
$x=(s_0,\ldots,s_{k_1},\ldots,s_{k_2},\ldots s_{k_3},\ldots,s_{k_4},\ldots s_m)$,
$0\leq k_1<k_2<k_3\leq k_4\leq m$ cumple las siguientes condiciones:
$\forall i,i<k_1:a,b\notin\mathcal{I}(s_i)$,
$\forall i,k_1\leq i<k_2:a\in\mathcal{I}(s_i)\wedge b\notin\mathcal{I}(s_i)$,
$\forall i,k_2\leq i<k_3:a,b\in\mathcal{I}(s_i)$,
$\forall i,k_3\leq i<k_4:a\notin\mathcal{I}(s_i)\wedge b\in\mathcal{I}(s_i)$ y
$\forall i,k_4\leq i\leq m:a,b\notin\mathcal{I}(s_i)$.
En particular s_{k_3} puede ser el último estado de la sucesión.
Es cierto $\mathcal{M},x\models F_o$, porque $\mathcal{M},x^{k_3}\models(\neg a\wedge b)$ y con eso
$\forall i,k_2\leq i<k_3:\quad\mathcal{M},x^i\models(ab\wedge((a\wedge b)\mathcal{U}\neg ab))$ y con eso
$\forall i,k_1\leq i<k_2:$
$\mathcal{M},x^i\models(a\neg b\wedge(a\neg b\mathcal{U}(ab\wedge(ab\mathcal{U}\neg ab))))$ y con eso
$\forall i,i<k_1:\mathcal{M},x^i\models(\neg a\neg b\mathcal{U}(a\neg b\wedge((a\neg b\mathcal{U}\neg a\neg b)\mathcal{U}\neg ab)))$
y con eso en particular
$\mathcal{M},x^0\models F_o$ %

Pero las fórmulas F_r tienen tambien modelos finitos, que no son interpretaciones de intervalos. Por eso tenemos que exigir que exista exactamente un a-intervalo en el modelo para cada intervalo A de la fórmula de intervalos.

Definición 4.6 *Llamamos a la fórmula*

$$F_a\equiv\Box((a\wedge\bigcirc_\forall\neg a)\rightarrow\bigcirc_\forall\Box\neg a)$$

fórmula única de intervalos a.

Se demuestra fácilmente:

Lema 4.7 *Sea $\mathcal{M} = (S, \mathcal{I}, x)$ una estructura lineal. Sea $F_a \equiv \Box((a \wedge \bigcirc_\forall \neg a) \rightarrow \bigcirc_\forall \Box \neg a)$.*
Es cierto: $\mathcal{M}, x \models F_a$ $l \iff$ existe a lo más un a-intervalo en $\mathcal{M}$.

Definición 4.8 *Llamamos a una fórmula de la lógica temporal lineal F_{FI} una LTL-fórmula de intervalos, si la recibimos de una fórmula de intervalos FI, sustituiendo cada símbolo de relación y sus proposiciones de intervalos por la fórmula básica correspondiente, y añadiendo por una conjunción para cada proposición A la fórmula única F_a.*

Teorema 4.9 *Los modelos finitos de una LTL-fórmula de intervalos F_{FI} son exactamente las interpretaciones de intervalos que representan escenarios consistentes de la fórmula de intervalos correspondiente FI.*

Demostración Cada modelo finito $\mathcal{M} = (S, \mathcal{I}, x)$ de una LTL-fórmula F_{FI} es una interpretación de intervalos. Como hemos añadido para cada intervalo A la fórmula única F_a tenemos la garantía de que hay a lo más un a-intervalo en $\mathcal{M}$.
$\mathcal{M}$ es un modelo de F_{FI} ssi representa un escenario consistente: Se prueba por inducción de la longitud de la fórmula. El caso básico resulta de 4.5 y de lo anterior. %

Las reglas de transitividad se mantienen trivialmente en esta nueva representación.

La algebra de puntos no considera los intervalos enteros sino sólamente los puntos iniciales y finales. Por lo tanto las proposiciones de la LTL representan puntos. La traducción de las relaciones es evidente. Por ejemplo podemos traducir $a^+ < b^-$ por $\Diamond(a^+ \wedge \bigcirc \Diamond b^-)$. Por la introducción de $(\Diamond p \wedge \Box(p \rightarrow \bigcirc_\forall \Box \neg p))$ para cada proposición garantizamos que cada punto es cierto en exactamente un estado del modelo.

5 Aplicación de Algoritmos de la LTL para la Lógica de Intervalos

5.1 Busqueda de un escenario consistente

En 2.4 presentamos un algoritmo que construye un modelo finito de una fórmula de la LTL. Podemos emplear este algoritmo directamenta para construir un modelo de una LTL-fórmula de intervalos, porque estas fórmulas tienen por lo menos un modelo finito. Como las fórmulas pueden ser extensas, es oportuno expresar conocimientos incompletos lo más breve posible. Si buscamos un escenario consistente no hacen falta LTL-fórmulas de intervalo completos. Además es posible ampliar el algortimo de 2.4 para que construya sólamente caminos que no lesionan la singularidad de los intervalos. Por lo tanto no hacen falta las fórmulas únicas.

5.2 Busqueda de todos los escenarios consistentes

Si una LTL-fórmula de intervalos es completa podemos obtener todos los modelos minimales de la fórmula por medio del mismo algoritmo. Para hacer esto construimos el tablero completamente. Todos los caminos componiéndose de nodos especiales desde la raíz hasta las hojas abiertas son modelos de la fórmula. El conjunto de estos caminos es un grafo dirigido, finito y sin ciclos. Para generar sólamente modelos minimales el algoritmo tiene que garantizar las siguientes condiciones:

- Estados iniciales y finales tienen que estar etiquetados por lo menos con un literal positivo.
- Dos estados sucesivos no deben tener los mismos literales en sus etiquetas.

La entrada de una fórmula completa garantiza que el algoritmo construye todos los modelos minimales posibles.

5.3 Consultas

Si tenemos un conjunto de conocimientos temporales, sea en forma de fórmulas, sea como escenarios consistentes, podemos hacer consultas.
Sean A y B intervalos, sobre cuales hemos almacenado conocimientos temporales. Consultas típicas serán:

1. ¿Hay un escenario consistente en que A está antes de B?
2. ¿Se verifica siempre que A está antes de B?
3. ¿Es posible que A sea un intervalo sin duración?

Según que forma de almacenamiento de representar que hemos elegido podemos responder a las preguntas de diferentes maneras. Consideramos aquí sólamento la lógica de intervalos y no la algebra de puntos.

5.3.1 Representación de conocimientos por fórmulas

Sea F una fórmula de intervalos que representa conocimientos temporales.

1. Si tenemos las relaciones posibles entre A y B en forma de una disjunción entonces se verifica:
   Si $(A < B)$ es una componente de la disjunción, sustituye la disjunción por $(A < B)$ y prueba si la LTL-fórmula correspondiente es satisfactible. Si $(A < B)$ no es una componente de la disjunción, no hay un escenario consistente.
   Si no hay una disjunción para A y B, prueba si la LTL-fórmula correspondiente a $F \wedge (A < B)$ es satisfactible.
2. Prueba si la LTL-fórmula correspondiente a $F \wedge \neg(A < B)$ es satisfactible. Es cierto que A siempre está antes de B si la fórmula no es satisfactible.
3. Sea F' la LTL-fórmula correspondiente a F. Prueba si $F' \wedge \Box(a \rightarrow \bigcirc_{\forall}\Box\neg a)$ es satisfactible.

5.3.2 Representación de conocimientos por escenarios consistentes

Como describimos en 5.2, un grafo dirigido y sin círculos puede representar todos los escenarios consistentes de una fórmula de intervalos. Podemos aplicar los algoritmos de model-checking a ese grafo. Sea $\mathcal{G}$ un grafo dirigido acíclico que represente los conocimientos temporales.

1. Prueba si existe en $\mathcal{G}$ un camino x en el cual $\mathcal{M}, x \models (\neg a \neg b \mathcal{U}(a \neg b \wedge ((a \neg b \mathcal{U} \neg a \neg b)\mathcal{U} \neg ab)))$
2. Prueba si existe en $\mathcal{G}$ un camino x en el cual $\mathcal{M}, x \models \neg(\neg a \neg b \mathcal{U}(a \neg b \wedge ((a \neg b \mathcal{U} \neg a \neg b)\mathcal{U} \neg ab)))$
   A siempre está antes de B si no hay un camino.
3. Prueba si existe en $\mathcal{G}$ un camino x con lo cual $\mathcal{M}, x \models \Box(a \rightarrow \bigcirc_{\forall}\Box\neg a)$

6 Conclusión

Hemos traducido la representación de conocimientos temporales desarrollados en inteligencia artificial a la lógica temporal lineal y hemos presentado como podemos emplear los algoritmos de satisfactibilidad y de model-checking para deducir sobre intervalos. En [11] se encuentran más aplicaciones de estos algoritmos. Indicamos tambien que la lógica temporal ramificada parece adecuada para la consultas porque se puede expresar en una forma directa propiedades existenciales y universales. Además la complejidad del model-checking de la lógica ramificada es sólo polinomial.

Una proxima etapa puede ser representar conocimientos cuantitativos (como "el intervalo A termina 10 minutos antes del comienzo de B"). Base de eso pueden ser las lógicas temporales que se aplica en la verificación de real-time-systems y que disponen de operadores temporales cuantitativos. La traducción de limitaciones temporales, como se encuentran en [5], parece posible con estas lógicas.

References

[1] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] M. Ben-Ari. *Math. Logic for Computer Science*, chapter 5, pages 200–243. Prentice Hall, 1993.

[3] M. Ben-Ari, Z. Manna, and A. Pnueli. The temporal logic of branching time. In *Proc. 8th Ann. Symp. on Principles of Programming Languages*, pages 164–176, 1981.

[4] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications: A practical approach. In *Proc. 10th Ann. ACM Symp. on Principles of Progamming Languages*, Austin, Texas, 1983.

[5] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[6] Pierre Marie Delpech, Alix Potet, and Claudette Sayettat. Démonstration automatique et logiques temporelles. *Technique et Science Informatiques*, 6(6):541–556, 1987.

[7] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 995–1072. Elsevier Science Publisher, 1989.

[8] Michael Fisher. A model checker for linear time temporal logic. *Formal Aspects of Computing*, 4:299–319, 1992.

[9] Christian Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54:199–227, 1992.

[10] Pascal Gribomont and Pierre Wolper. Logique temporelle. In André Thayse, editor, *Approche Logique de l'Intelligence Artificielle*, pages 179–258. Bordas, 1989.

[11] B. Kemme. Algoritmos de satisfactibildad y model-checking para la lógica temporal proposicional: Comparación y aplicación para la representación de conocimientos temporales. Master's thesis, Departamento de Algebra, Computación, Geometría y Topología , Universidad de Sevilla, Lehrstuhl für Künstliche Intelligenz, Friedrich-Alexander-Universität Erlangen-Nürnberg, 1994.

[12] Orna Lichtenstein and Amir Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *Proc. 12th Ann. ACM Symp. on Principles of Progamming Languages*, pages 97–107, 1894.

[13] Peter van Beek. Reasoning about qualitative temporal information. *Artificial Intelligence*, 58:297–326, 1992.

[14] M. Vilain and H.A. Kautz. Constraint propagation algorithms for temproal reasoning. In *Proceedings AAAI-86*, pages 132–144, Philadelphia, 1986.