LÓGICA MODAL EN MAS: Lógica Epistémica y Temporal

VÍCTOR RAMOS GONZÁLEZ

Síntesis, Verificación y Razonamiento de Agentes Inteligentes

Máster Universitario en Lógica, Computación e Inteligencia Artificial.

Universidad de Sevilla

Curso 2021-2022

¿Dónde estamos?

- Usos de MAS: formalizar diversos problemas en la informática, la inteligencia artificial, la teoría de los juegos, la teoría de la elección social y otras disciplinas relacionadas.
- Habitualmente, los componentes de estos sistemas son autónomos, inteligentes, activos e incluso proactivos, con objetivos y conocimiento sobre el mundo. Así, hemos de proveer a los agentes de cierto sentido racional, para lo cual, los métodos basados en la Lógica suelen ser los más adecuados.
- Hasta ahora, hemos tratado con la síntesis de sistemas multiagentes, presentando los sistemas multiagente casi como un paradigma para la computación, la programación y/o el diseño.
- Hoy lo dedicaremos a introducir **verificación**.

¿Para qué usar la Lógica en MAS?

- Comportamiento: BDI (KD4, KD, KD)
- Background Knowledge: S5
- Estudio temporal de sistemas: LTL, CTL, CTL*
- Coaliciones, estrategias y teoría de juegos: ATL y ATL*
- Comunicacición, Normas, Argumentación
- Programación de agentes
- Verificación de agentes

El porqué de la Lógica Modal en MAS

En el ámbito del MAS consideramos que los agentes están gobernados por un sentido racional en su comportamiento, inducido por su conocimiento, sus creencias, sus deberes, el tiempo, acciones, estrategias, etc.

Consideremos por ejemplo que tenemos que:

Alicia cree que a Benito le gusta el curso de MAS.

¿Cómo formalizar esto?

- Haciendo uso de FOL: $Cree(Alicia, Gusta(Benito, MAS))$
- Esa expresión ni siquiera es correcta sintácticamente.
- Podríamos introducir términos que representasen $Gusta(Benito, MAS)$, pero resulta muy poco elegante y dificulta enormemente el razonamiento.

Lógica Modal Normal. Aspectos formales (I)

Sintáxis

Formalmente, la Lógica Modal es una extensión de las Lógicas Clásicas (PL, FOL, HOL). Sintácticamente, siguen las reglas y operadores de las Lógicas Clásicas, pero incorporan nuevos operadores. Originalmente dos (duales):

$$\Box \text{ (\textit{necesidad})} \qquad \text{y} \qquad \Diamond \text{ (\textit{posibilidad})}$$

Semántica

(Def. 1) Marcos y Modelos de Kripke

Un **modelo Kripke** es un par $\langle F = \langle \mathcal{W}, \mathcal{R} \rangle, \mathcal{V} \rangle$ donde:

- $\mathcal{W} \subseteq \mathcal{P}(\mathcal{PV})$: conjunto no vacío de posibles mundos.
- $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$: relación de accesibilidad.
- $\mathcal{V} : \mathcal{PV} \to \mathcal{P}(\mathcal{W})$: conjunto no finito de posibles mundos.
- $F = \langle \mathcal{W}, \mathcal{R} \rangle$: marco de Kripke.

Lógica Modal Normal. Aspectos formales (II)

(Def. 2) Verdad local (Satisfactibilidad)

Dado un modelo Kripke $M = \langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$ y un mundo $w \in \mathcal{W}$, el valor de verdad asociado a una fórmula de la Lógica Modal corresponde a:

$$
\begin{array}{lcl}
M, w \models p \in \mathcal{PV} & \Longleftrightarrow & w \in \mathcal{V}(p) \\
M, w \models \neg\varphi & \Longleftrightarrow & M, w \not\models \varphi \\
M, w \models \varphi \wedge \psi & \Longleftrightarrow & M, w \models \varphi \wedge M, w \models \psi \\
 & \vdots & \\
M, w \models \Box\varphi & \Longleftrightarrow & \textit{para cada } w' \in \mathcal{W} \textit{ tal que } w\mathcal{R}w' \textit{ se tiene } M, w' \models \varphi \\
M, w \models \Diamond\varphi & \Longleftrightarrow & \textit{para algún } w' \in \mathcal{W} \textit{ tal que } w\mathcal{R}w' \textit{ se tiene } M, w' \models \varphi
\end{array}
$$

- φ es satisfactible si lo es para algún par (M, w)
- φ es insatisfactible si no lo es para ningún par (M, w)

Lógica Modal. Aspectos formales (III)

(Def. 3) Verdad global (Validez débil)

Dado un modelo Kripke $M = \langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$, una fórmula φ se dice verdadera en M, $M \models \varphi$, si es localmente verdadera para todo mundo $w \in \mathcal{W}$ del modelo M.

(Def. 4) Validez

Dado un marco Kripke $F = \langle \mathcal{W}, \mathcal{R} \rangle$, una fórmula φ se dice:

- válida en un mundo w, $F, w \models \varphi$, si para todo modelo $M = \langle F, \mathcal{V} \rangle$, $M, w \models \varphi$;
- válida en F, $F \models \varphi$, si para todo mundo $w \in \mathcal{W}$ se tiene $F, w \models \varphi$;
- válida en en una clase de marcos/modelos $\mathcal{F}/\mathcal{M}$, $\mathcal{F}/\mathcal{M} \models \varphi$, si es válida para cualquier marco/modelo de la clase;
- válida, $\models \varphi$, si es válida para cualquier marco.

Lógica Modal. Inferencia

Teoría de la Correspondencia

- Aunque generalmente el establecimiento de valor de verdad de una fórmula conlleva la comprobación exhaustiva de las relaciones asociadas, en muchos casos existen propiedades en la relación de accesibilidad que pueden ser capturadas mediante axiomas utilizables en la inferencia (Teoría de la correspondencia).
- En concreto una lógica se dice normal si cumple dos axiomas:
  - K-axioma (axioma de distribución) : $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$.
  - Regla de necesidad : Si $\models \varphi$ entonces $\models \Box\varphi$
- Otros axiomas:

Axioma	Propiedad de $\mathcal{R}$	Caracterización
T : $\Box\varphi \to \varphi$	Reflexiva	$\forall w : w\mathcal{R}w$
D : $\Box\varphi \to \Diamond\varphi$	Serie	$\forall w \exists w' : w\mathcal{R}w'$
4 : $\Box\varphi \to \Box\Box\varphi$	Transitiva	$\forall w w' w'' : w\mathcal{R}w' \wedge w'\mathcal{R}w'' \to w\mathcal{R}w''$
5 : $\Box\varphi \to \varphi$	Euclídea	$\forall w w' w'' : w\mathcal{R}w' \wedge w\mathcal{R}w'' \to w'\mathcal{R}w''$

Corrección, Completitud y Computabilidad en Lógicas Modales

Corrección y completitud en la Lógica Modal Normal

Sea $\boldsymbol{X}$ cualquier subconjunto de $\{D, T, 4, 5\}$ y $\mathcal{X}$ el conjunto de propiedades de relación correspondientes a $\boldsymbol{X}$. Se tiene que $\boldsymbol{K} \cup \boldsymbol{X}$ es correcto y completo respecto del conjunto de modelos Kripke cuya relación de accesibilidad satisface $\mathcal{X}$

Computabilidad de la Lógica Modal Normal

La Lógica Modal K (y por ende sus extensiones $\{D, T, 4, 5\}$) cumple la Propiedad de Modelos Finitos (fuerte) y por tanto es decidible (computable).

Lógicas Modales

La Lógica Modal como marco genérico para formalizar la realidad

Conocimiento	$\rightsquigarrow$	**L. epistémica**	Acciones	$\rightsquigarrow$	**L. dinámica**
Creencias	$\rightsquigarrow$	**L. doxástica**	Tiempo	$\rightsquigarrow$	**L. temporal**
Obligaciones	$\rightsquigarrow$	**L. deóntica**	Estrategias	$\rightsquigarrow$	**L. estratégica**

- Lógica epistémica:
  - Lenguaje de la Lógica Epistémica.
  - Model Checking en Lógica Epistémica
- Lógica ~~dinámica~~ y temporal:
  - Lineal vs Branching (LTL vs CTL*)
  - Model Checking en CTL (y otros)

Lógica Epistémica. Aspectos formales (I)

- Lógica multimodal con i operadores modales (y sus correspondientes duales):

$K_i\varphi$: *el agente i conoce que* φ *es cierto*

- El marco es extendido como$\langle \mathcal{W}, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n \rangle$ en el que cada $\mathcal{R}_i$ corresponde a la relación de accesibilidad del agente i.
- Extensión natural de la semántica:

$$M, w \models K_i\varphi \iff \text{\textit{para cada} } w' \text{ \textit{tal que} } w\mathcal{R}_i w' \text{ \textit{se tiene} } M, w' \models \varphi$$

- Se pueden/suelen considerar operadores de conocimiento colectivo.

Lógica Epistémica. Aspectos formales (II)

Operadores de conocimiento colectivo

$$M, w \vDash Op_A \varphi \iff \textit{para cada } w' \in \mathcal{W} \textit{ tal que } w \mathcal{R}_{Op_A} w' \textit{ se tiene } M, w' \vDash \varphi$$

- Conocimiento mutuo - E_A: "todos en el grupo A saben que ..."

$$\mathcal{R}_{E_A} = \bigcup_{i \in A} \mathcal{R}_i$$

- Conocimiento común C_A: "todos en el grupo A saben que todos conocen que..."

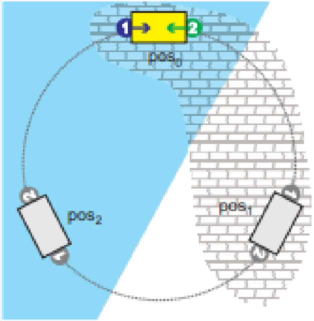
$$\mathcal{R}_{C_A} = CT(E_A)$$

- Conocimiento distribuido D_A: "si los agentes compartieran su información disponible, serían capaces de reconocer que..."

$$\mathcal{R}_{D_A} = \bigcap_{i \in A} \mathcal{R}_i$$

Lógica Epistémica. Robots y Carro I

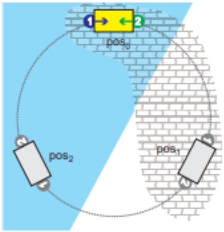
Dos robots empujan un carro desde lados opuestos. Luego, el carro puede moverse en el sentido de las agujas del reloj, en sentido contrario, o permanecer en el mismo lugar. El robot 1 sólo es capaz de observar el color de la superficie sobre la que está que se encuentra, y el robot 2 sólo percibe la textura.

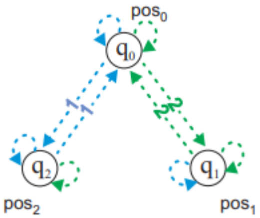


Lógica Epistémica. Robots y Carro II

Luego en el modelo Kripke correspondiente se identifican:

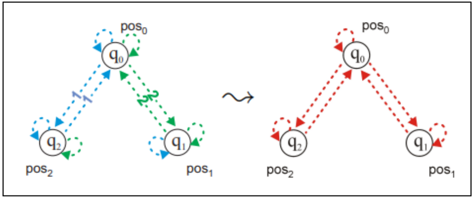
- 3 posiciones diferentes del carro, asociadas con los estados q_0, q_1 y q_2.
- Las proposiciones pos_0, pos_1, pos_2, para poder referirnos a la posición actual del carro en el lenguaje del objeto.
- Las relaciones correspondientes a la indistinguibilidad de los estados.

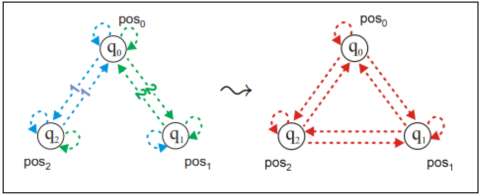


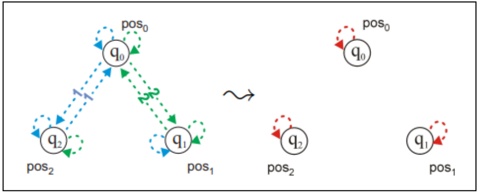


Lógica Epistémica. Robots y Carro III

¿Relaciones de conocimiento colectivo: $E_{1,2}$, $C_{1,2}$, $D_{1,2}$?







Lógica Epistémica. Sistemas $K - \{DT45\}$

¿Cómo interpretar los axiomas D, T, 4 y 5 en ámbito del conocimiento?

- Axioma D : los conocimientos son siempre consistentes (no contradictorios)

$$K_i\varphi \rightarrow \neg K_i\neg\varphi$$

- Axioma T : Todo aquello que conoce un agente ha de ser cierto. (Conocer $\neq$ Creer)

$$K_i\varphi \rightarrow \varphi$$

- Axioma 4 : (Instrospección positiva) un agente sabe que conoce todo lo que conoce.

$$K_i\varphi \rightarrow K_iK_i\varphi$$

- Axioma 5 : (Instrospección negativa) un agente sabe que no conoce todo lo que no conoce.

$$\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$$

- Normalmente $S5 = KDT45$ es tomada como Lógica de Conocimiento y $KD45$ como Lógica de Creencias.

Aplicaciones de Lógica Modal en MAS

- Apoyo al análisis de un dominio de problemas y el diseño de un nuevo sistema, Pros: estructuras conceptuales intuitivas y un enfoque sistemático para especificar propiedades deseables del sistema. Contras: No implementable directamente.
- Verificación y exploración de un MAS existente mediante la comprobación de modelos, la demostración de teoremas o la prueba de corrección.
- Generación automática de comportamientos mediante la programación con especificaciones ejecutables y la planificación automática.
- Inferencia - intratable en la práctica para la mayoría de las lógicas modales.

Comprobación de Modelos (Model Checking)

Model Checking - Verificación de Sistemas

Problema de decisión asociado a la pertenencia al conjunto:

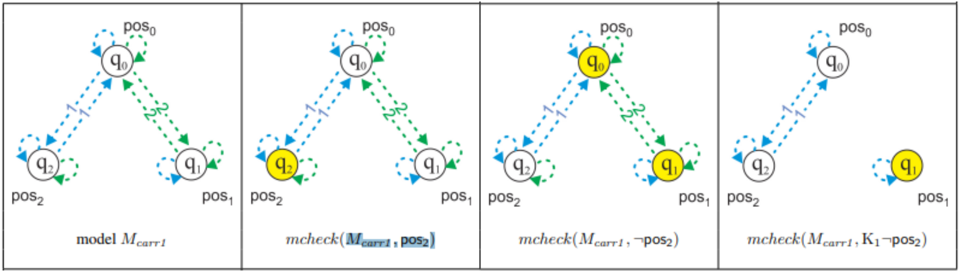
$$MC(\mathcal{L}, \mathcal{S}, \vDash) = \{((M, w), \varphi) \in \mathcal{S} \times \mathcal{L} \,|\, M, w \vDash \varphi\}$$

- Versión Local: Dado $\mathcal{M}$, w y φ : Chequear si $M, w \vDash \varphi$.
- Versión Global: Dado $\mathcal{M}$ y φ : Obtener el conjunto de estados $w \in \mathcal{W}$ tal que $M, w \vDash \varphi$.

¿Eficiencia? $\Longrightarrow$ Complejidad $\boldsymbol{P}$

Comprobación de Modelos - Ejemplo

$$mcheck(M_{carr1}, K_1 \neg pos_2)$$



Lógica Temporal

Hasta ahora hemos tratado:

- Varios operadores K_i que definen las relaciones entre los posibles mundos.
- Descripción de sistemas estáticos

$$\Downarrow$$

Problema: LOS MAS SON SISTEMAS DINÁMICOS

- Modelización del tiempo. Liveness, Safety, Fairness
- Comportamientos en el tiempo $\longrightarrow$ LTL/CTL

Lógica Temporal

La lógica temporal es, en origen, desarrollada para la representación en el marco del lenguaje natural. Sin embargo, en el marco de las Ciencias de la Computación, ha alcanzado un desarrollo importante, sobre todo, en las tareas de especificación y verificación de sistemas.

Operadores temporales

$\bigcirc\varphi$ / $\boldsymbol{X}\varphi$	φ se hace cierto en el siguiente instante de tiempo
$\Box\varphi$ / $\boldsymbol{G}\varphi$	φ se hace cierto en todos los instantes futuros
$\Diamond\varphi$ / $\boldsymbol{F}\varphi$	φ será eventualmente cierto (en el futuro)
$\varphi\mathcal{U}\psi$ / $\varphi\boldsymbol{U}\psi$	φ será cierto en todo momento hasta que ocurra ψ

Modelización del tiempo en la Lógica Modal

Propiedades del tiempo

- Liveness : *Algo que sea bueno para el sistema ocurrirá* (objetivos de logro) - Operador $\boldsymbol{F}\varphi$.
- Safety: *lo bueno se mantendrá y lo malo se evitará* (Objetivos de mantenimiento) - Operador $\boldsymbol{G}\varphi$.
- Fairness : *Toda petición es satisfecha* (Objetivos de mantenimiento) - Operador $\boldsymbol{G}(\varphi \rightarrow \boldsymbol{F}\psi$ y GF.

Varios modelos del mundo:

- Mundo determinsta: Linea temporal Lineal (LTL).
- Mundo no determinista: Linea temporal ramificada (CTL, CTL*)

Determinismo: Linear Temporal Logic (LTL)

Lógica LTL

Extensión de la Lógica Proposicional añadiendo los operadores temporales X y U, a partir de los cuales pueden definirse otros: (eventualmente cierto) $F\varphi \equiv \top U \varphi$, (en adelante siempre cierto) $G\varphi \equiv \neg F \neg\varphi$, ($U$ débil) $\varphi W \psi \equiv (\varphi U \psi) \vee G\varphi$, (*release*) $\varphi R \psi \equiv \psi W(\varphi \wedge psi)$.

Modelos LTL

Un modelo LTL es una secuencia de estados a través del tiempo, que representan un camino λ (no necesariamente finito).

Notación:

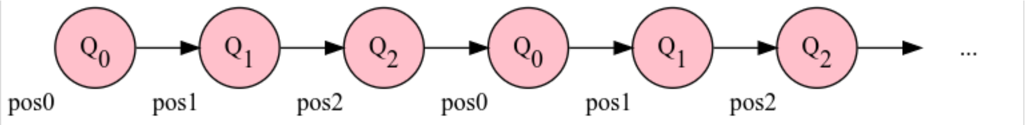
- $\lambda[i]$: i-ésimo momento.
- $\lambda[i..j]$: momentos de tiempo entre el i y el j.
- $\lambda[i..]$: todo momento a partir de i.

Determinismo: Linear Temporal Logic (LTL)

Semántica LTL

- $\lambda \models p \iff \lambda[0] \in \mathcal{V}(p)$, con p una fórmula atómica y V una función que asigna a cada estado un conjunto de fórmulas atómicas (etiquetas) verdaderas en él.
- $\lambda \models \neg\varphi \iff \lambda \not\models \varphi$
- $\lambda \models \varphi \wedge \psi \iff \lambda \models \varphi \wedge \lambda \models \psi$
- Similar para el resto de conectivas proposicionales
- $\lambda \models X\varphi \iff \lambda[1...] \models \varphi$
- $\lambda \models \varphi U \psi \iff \exists i \geq 0 | \lambda[i...] \models \psi \wedge \forall j < i : \lambda[j..] \models \varphi$
- $\lambda \models F\varphi \iff \exists i \geq 0 | \lambda[i...] \models \varphi$
- $\lambda \models G\varphi \iff \forall i \geq 0 : \lambda[i...] \models \varphi$
- $\lambda \models \varphi W \psi \iff \forall i = 0, \ldots, j-1 : \lambda[i...] \models \varphi$ con j el primer momento tal que $\lambda[j...] \models \psi$ e.o.c. $j = \infty$
- $\lambda \models \varphi R \psi \iff \forall i = 0, \ldots, j : \lambda[i...] \models \psi$ con j el primer momento tal que $\lambda[j...] \models \varphi$ e.o.c. $j = \infty$

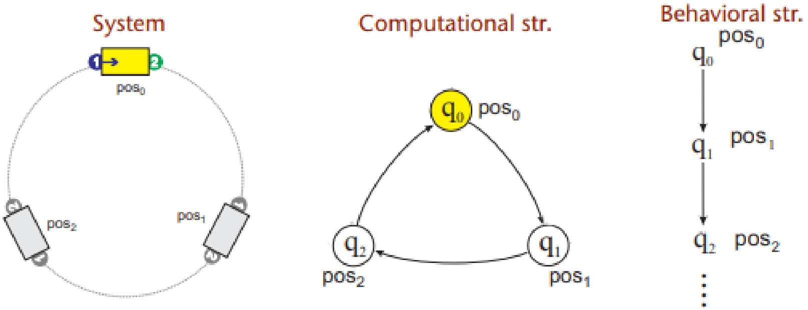
Determinismo: Linear Temporal Logic (LTL)



$$¿\lambda \vDash pos_0?,\ ¿\lambda \vDash F pos_2?,\ ¿\lambda \vDash GF pos_2?,\ ¿\lambda \vDash FG pos_2?$$

PROBLEMA: ¿VERIFICACIÓN AUTOMÁTICA EN CAMINOS INFINITOS?
PROBLEMA: ¿COMPLEJIDAD MODEL CHECKING EN LTL?

Estructura computacional vs estructura de comportamiento

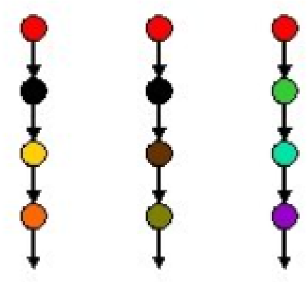


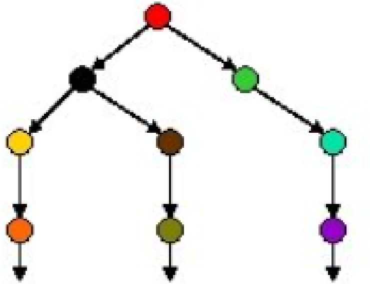
PROBLEMA: ¿COMPLEJIDAD MODEL CHECKING EN LTL?

$\Downarrow$

P-SPACE completa

Determinsmo vs No Determinismo





No determinismo: Branching Temporal Logic (CTL*/CTL)

Lógica CTL*

Extensión de la Lógica CTL* amplia la Lógica LTL con dos operadores duales: E y A que expresan la existencia de un camino en el que se cumpla cierta fórmula y el cumplimiento de la fórmula en todo camino

Modelos CTL/CTL*

Un modelo CTL corresponde a un árbol de posible computaciones en el que cada una de las ramas es una secuencia completa de estados a través del tiempo, que representan un camino λ (no necesariamente finito).

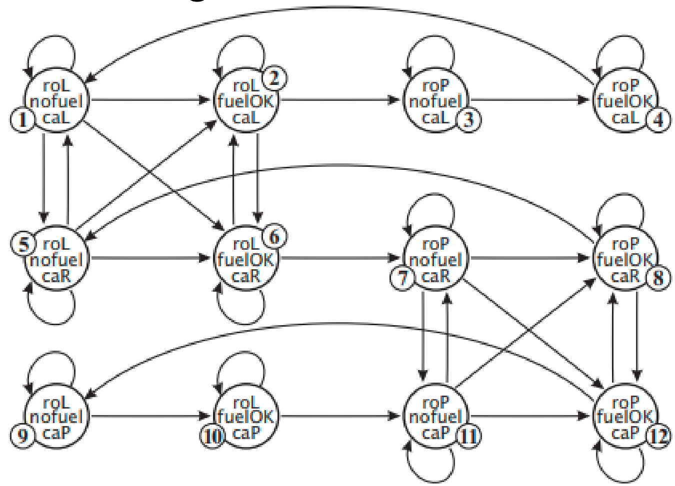
Lógica CTL

Limitación de la Lógica CTL en el que cada operador temporal está inmediatamente precedido de un operador de cuantificación. Ventaja: El razonamiento en CTL es automatizable.

Ejemplo: Rocket & Cargo

La figura presenta un modelo de Kripke M' para una variante sencilla del dominio del cohete que solía ser popular en la comunidad de planificación de STRIPS. Un cohete puede estar en su base en Londres (denotado por la proposición roL) o en el aeródromo de París (roP). El cohete también puede desplazarse entre los aeródromos si tiene combustible ($fuelOK$), pero el vuelo gasta todo el combustible y vacía el depósito de combustible ($nofuel$) hasta que se rellene. Una carga puede estar en la base de Londres (caL), en la base de París (caP), o dentro del cohete (caR). Cuando está fuera, puede cargarse en el cohete si están en la misma ubicación. Cuando está dentro, se puede descargar.

Ejemplo: Rocket & Cargo



Ejemplo: Rocket & Cargo

Se puede comprobar que las siguientes fórmulas son ciertas en todo estado del modelo M':

- $EFcaR$: Hay algún camino en el que eventualmente la carga se encontrará en el cohete.
- $AG(roL \vee roP)$: En todos los estados alcanzables el cohete está en Londres o el cohete está en París
- $roL \rightarrow AX(roP \rightarrow noFuel)$: Si en un estado el cohete está en el aeródromo de Londres entonces para todo estado inmediatamente siguiente en el que el cohete esté en el aeródromo de París, el cohete estará sin combustible.

Model Checking - CTL

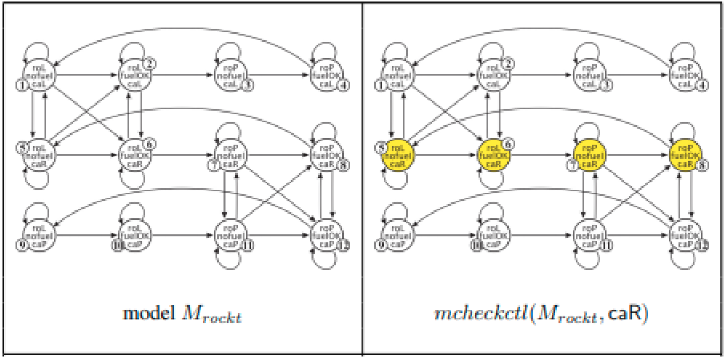
Apoyándose en el μ-cálculo y algunos resultados de Knaster-Tarsi y Kleen.

function $mcheck(\mathcal{M}, \varphi)$.
  Model checking formulae of CTL.
  Returns the exact subset of Q for which formula φ holds.

case $\varphi \equiv p$ **:** return $\{q \in Q \mid p \in \pi(q)\}$
case $\varphi \equiv \neg\psi$ **:** return $Q \setminus mcheck(\mathcal{M}, \psi)$
case $\varphi \equiv \psi_1 \wedge \psi_2$ **:** return $mcheck(\mathcal{M}, \psi_1) \cap mcheck(\mathcal{M}, \psi_2)$
case $\varphi \equiv \mathsf{E}\bigcirc\psi$ **:** return $pre(mcheck(\mathcal{M}, \psi))$
case $\varphi \equiv \mathsf{E}\Box\psi$ **:**
  $Q_1 := Q; \quad Q_2 := Q_3 := mcheck(\mathcal{M}, \psi);$
  while $Q_1 \not\subseteq Q_2$ **do** $Q_1 := Q_1 \cap Q_2; \; Q_2 := pre(Q_1) \cap Q_3$ **od;**
  return Q_1
case $\varphi \equiv \mathsf{E}\psi_1 \mathcal{U} \psi_2$ **:**
  $Q_1 := \emptyset; \quad Q_2 := mcheck(\mathcal{M}, \psi_2); \quad Q_3 := mcheck(\mathcal{M}, \psi_1);$
  while $Q_2 \not\subseteq Q_1$ **do** $Q_1 := Q_1 \cup Q_2; \; Q_2 := pre(Q_1) \cap Q_3$ **od;**
  return Q_1
end case

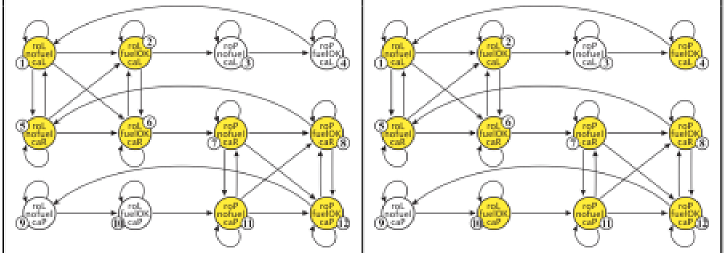
Model Checking - CTL

$$EFcaR$$



Model Checking - CTL

$$EFcaR$$

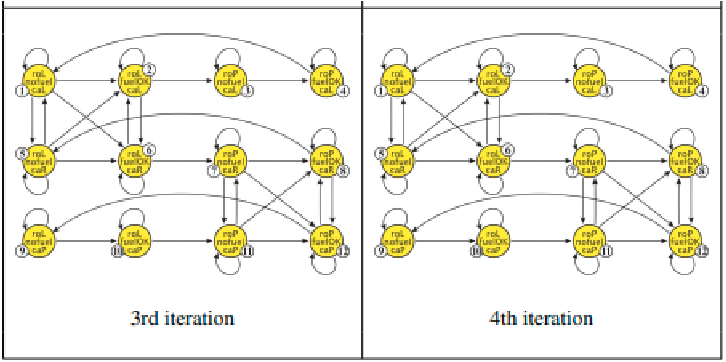


$mcheckctl(M_{rockt}, \mathrm{EFcaR})$: 1st iteration

2nd iteration

Model Checking - CTL

$$EFcaR$$



Notas & Conclusiones

¿Para qué podemos usar la Lógica en los MAS?

- Modelización de sistemas y especificación de propiedades de los sistemas
- Verificación de corrección de sistemas
- Otros: Programación con especificaciones ejecutables, Planificación automática, caracterización de actitudes de los agentes, estudio de agentes racionales, ...

Algunas notas:

- Existen muchos tipos de Lógicas Modales, en varios órdenes y con propósitos muy variados: conocimiento, deber, tiempo, estrategia (teoría de juegos), acciones (PDL), combinaciones (BDI)...
- Con la Lógica se demuestra la necesidad de la comunicación para el ámbito del conocimiento.
- La Lógica epistémica permite expresar condiciones de conocimiento y creencias, mientras que la temporal razona sobre el tiempo. NO SON INDEPENDIENTES y su combinación da lugar a otras lógicas: Lógicas de estrategia (ATEL).
- Uno de los usos principales es el model checking: SPIN, MOCHA, MCMAS, VERICS, ...