BIG BROTHER LOGIC

Samuel Hidalgo Berlanga
Paula Cuadra Ruiz
Sergio Camacho Marín

ÍNDICE

Introducción……………………………………………………………………...página 2
Preliminares……………………………………………………………………...página 4
Lógicas para agentes con cámaras de vigilancia fijas…………………….página 7
Expresividad y extensiones……………………………………………………página 9
Forma de BBL similar a PDL…………………………………………………..página 13
Bibliografía………………………………………………………………………página 15

<u>Introducción</u>:

El modelado y estudio de sistemas de múltiples agentes que involucran agentes inteligentes combinando habilidades perceptivas y de razonamiento es un campo importante de investigación con aplicaciones muy importantes en la inteligencia artificial (IA). Un aspecto importante a tratar es el procesamiento analítico de la percepción visual y la comunicación visual. Este aspecto presenta un problema y es que a día de hoy, es difícil encontrar un sistema que represente y razone lógicamente sobre la información visual que los agentes reciben del entorno y de los demás agentes. Este problema afecta no sólo al conocimiento de los agentes sobre el mundo que les rodea, sino sobre el propio conocimiento que es el que se encarga de guiar el comportamiento inteligente.

Este artículo trata estos problemas poniendo como contexto escenarios múltiples donde cada agente controla una cámara de vigilancia fija sobre un plano. Cada cámara tiene una posición y ángulo de visión fijos, pero tiene la capacidad de girar libremente alrededor de su eje. A través de estas cámaras, los agentes observan el mundo circundante y entre ellos.

Para abordar el tema, vamos a establecer unos supuestos básicos en estos escenarios:

- Los agentes son identificados con sus cámaras, por ello, podemos hablar sobre agentes que ven objetos y lo que las cámaras son capaz de conocer.
- Suponemos que las cámaras / agentes están colocados en puntos únicos en el plano y que son transparentes. También podemos considerar a las cámaras / agentes no transparentes y de tamaño real, lo cual no significa ninguna modificación importante en el problema a abordar.
- Asumimos que los únicos objetos de interés para cualquier agente son las demás cámaras/agentes.
- También asumimos que los agentes son estacionarios, a pesar de que en la realidad suelen ser móviles.
- Gracias a que son estacionarios, asumimos que cada agente conoce las posiciones exactas de todos los agentes / cámaras del plano, incluso de aquellas que no ve actualmente. Es un conocimiento común entre todos los agentes. Esta suposición está justificada si los agentes poseen algo de memoria, y en ese caso, podrían darse la vuelta para observar y memorizar las posiciones de los demás antes de seguir razonando y actuando por su cuenta.

De estas suposiciones surge una suposición: cómo se relacionan la visión y el conocimiento que tienen los agentes. A lo largo del informe, se va a demostrar que no sólo saben (los agentes) sobre el conocimiento de otros agentes, sino que existe un aspecto deductivo no trivial en el conocimiento analítico que pueden adquirir, surgiendo de un razonamiento geométrico suplementario.

Preliminares:

Modelos y lógicas epistémicas de agentes múltiples

Una estructura epistémica para un conjunto de agentes múltiples **Agt** es una tupla $\boldsymbol{S} = \langle \mathbf{Agt}, \mathbf{St}, \{\sim\mathbf{a} \mid \mathbf{a} \in \mathbf{Agt}\rangle$ donde:

- **St** es el conjunto de posibles estados (posibles mundos) y
- ~**a** es la relación epistémica de indistinguibilidad en **S** del agente **a**.

Dado un conjunto fijo de proposiciones atómicas ***PROP***, un modelo epistémico de agentes múltiples (**MAEM**) extiende una estructura epistémica (*S*) de agentes múltiples con una valoración de verdad *V* que asigna a cada $p \in \boldsymbol{PROP}$ un conjunto de estados *V(p)* donde se declara verdadero.

Consideramos lógicas epistémicas multiagente totalmente expresivas, y ampliando la lógica clásica proposicional agregamos los siguientes operadores:

- **Conocimiento individual**: K_a, que significa "El agente **a** sabe que…".
- **Conocimiento grupal**: K_A , que significa "El grupo de agentes **A** sabe que…".
- **Conocimiento Distribuido**: D_A, que significa "A partir del conocimiento de los agentes del grupo **A** se sabe que…".
- **Conocimiento común**: C_A , es similar al conocimiento grupal con la particularidad de que en este los agentes son conscientes de que el resto del grupo conoce la información.

Flatland

La interacción entre la percepción visual y el conocimiento se ha estudiado en un marco lógico basado en el siguiente lenguaje donde a y b son agentes:

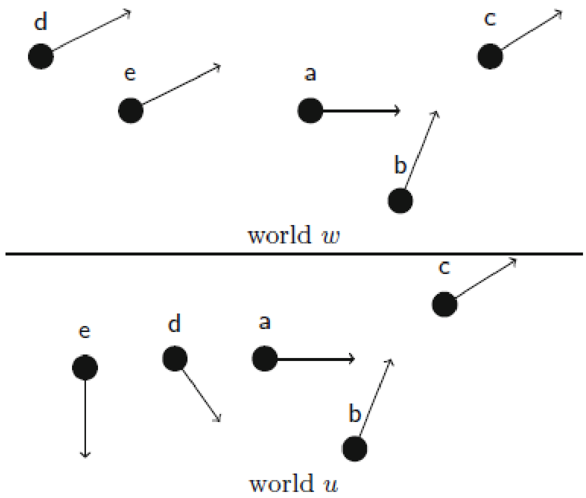
$(L_{\rhd}, K)\ \phi ::= \ a \rhd b \ \mid \ \neg\phi \ \wedge \ \phi \ \mid \ K_a\phi$

La preposición atómica $a \rhd b$ se lee "el agente **a** ve al agente **b**". El lenguaje $L_{\rhd}$, K no tiene proposiciones atómicas abstractas. Los modelos previstos primarios son modelos geométricos donde cada agente está ubicado en un punto del espacio y ve un medio espacio. Diferentes agentes están ubicados en diferentes puntos y cada agente **a** que pueda ver otro agente **b** también puede ver la dirección de visión de **b**.

Una semántica abstracta se basa en modelos de Kripke donde los mundos posibles son todos geométricos. La relación de indistinguibilidad ~**a** para el agente **a** en un modelo de Kripke se define de la siguiente forma:

Dos mundos ***w*** y ***u*** son tales que ***w*** $\sim a$ ***u*** sí y sólo sí los agentes vistos por el agente **a** son iguales y están ubicados de la misma manera en ambos mundos ***w*** y ***u***, es decir, que estén ocupando la misma posición y apunten en la misma dirección.

Se han estudiado dos variantes de este desarrollo, **Lineland** donde el espacio es unidimensional, y **Flatland**, donde el espacio es bidimensional, es decir, un plano.



Esta figura muestra dos mundos posibles en el modelo Flatland que son indistinguibles para el agente **a**, porque se ven iguales en las partes del plano que **a** puede ver. También podemos notar que el agente **a** ve a los agentes **b** y **c** en ambos mundos, por lo que **a** sabe que **b** es capaz de ver a **c**. Esto se expresaría como: $K_a(b \rhd c)$, que equivale a: $a \rhd b \ \wedge \ a \rhd c \ \wedge \ b \rhd c$. Además, **a** no puede ver ni al agente **d** ni al agente **e** en ninguno de los dos mundos, por lo que no sabe que es capaz de ver cada uno de ellos por separado. El agente **d** ve al agente **e** en el mundo ***w*** pero no lo ve en el mundo ***u***.

La verificación y las pruebas de satisfacción son PSPACE-completas para Lineland y PSPACE-duras y EXPSPACE para Flatland.

<u>Teoría de primer orden de los reales</u>

Considerando el siguiente lenguaje de primer orden para campos L_F:

$$\phi ::= \ (e \ > 0) \mid (e \ = \ 0) \mid \neg\phi \mid (\phi \ \wedge \ \phi) \mid \forall x\phi \mid \exists x\phi$$

donde:

- e ,es una expresión polinómica sobre las variables x, y…
- L_F,tiene una interpretación estándar en el campo de los reales $\mathbb{R}$.

El problema de comprobar si una fórmula cerrada de L_F es verdadera en $\mathbb{R}$ está en EXPSPACE.

Lógicas para agentes con cámaras de vigilancia fijas

Aquí es donde comenzamos a introducir la lógica básica para cámaras fijas, llamada Big Brother Logic o BBL. Consideraremos escenarios donde muchos agentes están posicionados en un plano, cada uno de ellos equipado con una cámara con un ángulo fijo de visión. Los agentes pueden girar libremente sus cámaras pero no pueden cambiar de posición (como mencionamos anteriormente). En todo momento un agente ve todo lo que se encuentra dentro de su sector de visión actual. En específico, asumimos por conveniencia técnica que las cámaras / agentes no tienen dimensiones y que están colocados en puntos únicos siendo transparentes (se puede ver a través de un agente). En el caso de que un agente no sea transparente y no se pueda ver a través de él, será considerado como un **obstáculo**.

Lenguajes

Vamos a fijar un conjunto de agentes **Agt** = {a1, … , ak}. El lenguaje básico utilizado en BBL incluye el operador **'ver'** (▹) mencionado anteriormente en el apartado *Flatland*. También incluye los operadores epistémicos de conocimiento $\{K_a\}$ a∈**Agt**, además de operador **'girando'** $\{\langle \overset{\frown}{a} \rangle\}$a∈**Agt** que significa que el agente **a** puede dar la vuelta a su posición para que ϕse mantenga como está.

Las fórmulas de BBL se definen como:

$$\phi ::= a \rhd b \mid \neg\phi \wedge \phi \mid K_a\phi \mid \langle \overset{\frown}{a} \rangle \phi$$

Además, también introduciremos las siguientes expresiones BBL:

- BBL_C : BBL añadiendo todos los operadores C_A para cada grupo de agentes **A**.
- BBL_D : BBL añadiendo todos los operadores D_A para cada grupo de agentes **A**.
- Combinaciones de las expresiones anteriores.

También escribiremos a $\not\rhd$ b como abreviación de ¬a ▹ b.
Algunos operadores adicionales que serán útiles a lo largo del desarrollo BBL:
- a ⋈ b , que significa que "**a** ve a **b** y **b** ve a **a**".
- B(abc), que significa "**b** está entre **a** y **c**".

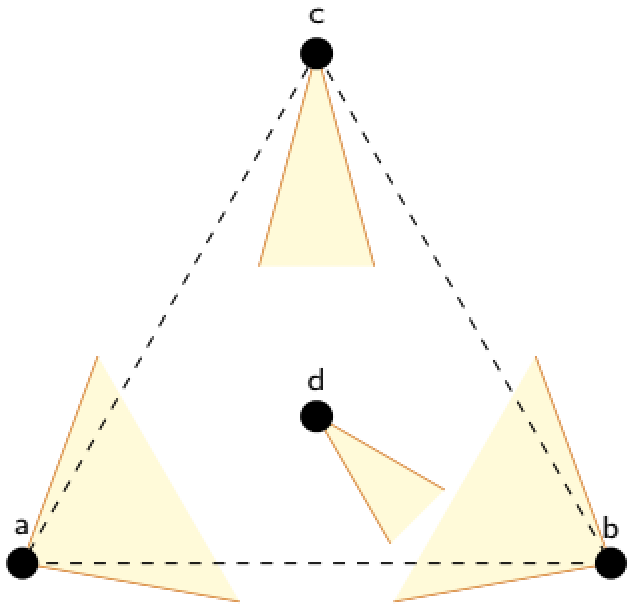
Semántica

Vamos a necesitar cuatro tipos distintos de modelos equivalentes.

- **Modelos geométricos**: son representaciones geométricas naturales de configuraciones de cámaras como puntos en el plano y de sus áreas de visión como sectores entre pares de rayos infinitos.
- **Estructuras geométricas de Kripke**: son estructuras de Kripke, siendo los mundos posibles los modelos geométricos posibles.
- **Abstracciones basadas en la visión**: también basado en estructuras de Kripke, que se obtienen del cociente finito de estructuras geométricas de Kripke. Servirá para la verificación algorítmica de modelos en dichas estructuras.
- **Modelos basados en la visión**: se construyen directamente a partir de modelos geométricos identificando sus espectros de visión (una tupla de familias de conjuntos de visión, una familia para cada agente, obtenida girando su cámara en un círculo completo para poder ver a los demás agentes). Básicamente son isomorfos a las abstracciones basadas en la visión y lo usaremos en los algoritmos de verificación de modelos reales.

Expresividad y extensiones

- **Expresando propiedades y especificaciones:** el lenguaje BBL puede expresar distintas especificaciones sobre un ámbito visual. Por ejemplo:
  - $\mathbf{a \triangleright b \rightarrow K_a[b\curvearrowright](a \triangleright b)}$: “Si **a** puede ver **b**, entonces, **a** sabe que cualquiera que sea la forma en que **b** se dé la vuelta, esto seguirá siendo cierto”.
  - $\mathbf{(b \triangleright a \wedge b \ntriangleright c) \wedge K_b \langle a\curvearrowright \rangle ((a \triangleright b \wedge a \triangleright c) \rightarrow K\ {}_b K_a(b \triangleright a \wedge b \ntriangleright c)}$: “Si **b** ve a **a** pero no a **c**, y sabe que **a** puede darse la vuelta para ver a ambos, **b** y **c,** entonces **b** considera que es posible que **a** sepa que **b** ve a **a**, pero no a **c**”.
  - $\mathbf{(a \bowtie b) \rightarrow C_{a,b}(a \bowtie b)}$:"Si **a** y **b** se ven, entonces esto es de conocimiento común entre ellos".
  - $\mathbf{K_{a,b}(\langle a\curvearrowright \rangle (a \triangleright b \wedge a \triangleright c) \wedge \langle b\curvearrowright \rangle (a \triangleright b \wedge a \triangleright c)) \rightarrow C_{a,b}(\langle a\curvearrowright \rangle \langle b\curvearrowright \rangle (a \bowtie b \wedge a \triangleright c \wedge b \triangleright c))}$
    “Si **a** y **b** saben que cada uno de ellos puede darse la vuelta para ver al otro, y a **c**, entonces es de conocimiento común entre ellos que ambos pueden volverse para verse y a **c**”.
  - $\mathbf{\langle a\curvearrowright \rangle (a \triangleright c \wedge a \triangleright d) \wedge \langle b\curvearrowright \rangle (b \triangleright d \wedge b \triangleright e) \rightarrow \langle a\curvearrowright \rangle \langle b\curvearrowright \rangle (D_{a,b} c \triangleright e \vee D_{a,b} c \ntriangleright e)}$: “Si **a** puede dar la vuelta para ver a ambos **c** y **d** y **e**, entonces a y b pueden darse la vuelta para convertirlo en un conocimiento distribuido entre ellos si **c** ve **e**”.
- **Definición de intermediación y colinealidad:** Para esto hay que suponer que el rango de visión de las cámaras está por debajo de 2π. Por lo que el lenguaje BBL puede expresar intermediación y colinealidad de la siguiente manera:
  - $\mathbf{[a\curvearrowright](a \triangleright b \rightarrow a \triangleright c)}$: es cierto cuando por ejemplo, c se encuentra en el rayo que comienza en a y pasa por b.
  - $\mathbf{[a\curvearrowright](a \triangleright b \rightarrow a \triangleright c) \wedge [b\curvearrowright](b \triangleright a \rightarrow b \triangleright c)}$:es cierto en un modelo geométrico precisamente cuando c se encuentra estrictamente entre a y b.



Entonces podemos definir,$\mathbf{[a\curvearrowright]((a \triangleright b \rightarrow a \triangleright c) \rightarrow a \triangleright d) \wedge [b\curvearrowright](b \triangleright a \rightarrow b \triangleright c) \rightarrow b \triangleright d)}$, como dis en el triángulo (a, b, c) pero solo cuando al menos 2 de los ángulos de visión de a, b, c sean lo suficientemente anchos para que cada uno de los agentes respectivos pueda darse la vuelta para ver a los otros dos.

- **Relaciones entre visión y conocimiento:** la vista y el conocimiento visual están estrechamente relacionados. Un agente sabe que ve todo lo que ve y sabe que no ve todo lo que no ve, pero hay más en el conocimiento visual del agente de lo que puede ver directamente. Por ejemplo, lo siguiente es válido para cualquier agente a, b, c1, c2, e, incluso si a no ve b:

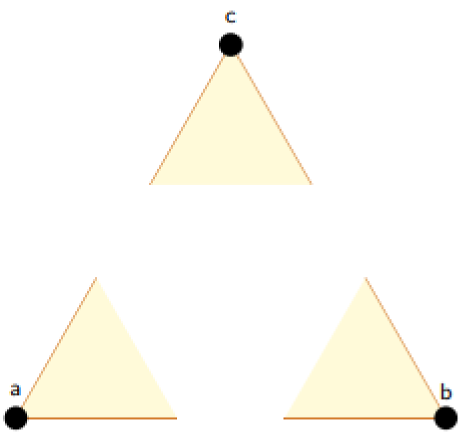
$$\mathbf{a \triangleright c_1 \wedge a \triangleright c_2 \wedge B(c_1 e c_2) \rightarrow K_a(b \triangleright c_1 \wedge b \triangleright c_2 \rightarrow b \triangleright e)}$$

- **Axiomas para cámaras con diferentes ángulos de visión:** el conjunto de fórmulas BBL válidas es muy sensible a los ángulos de visión admisibles de las cámaras de los agentes.

  Suponiendo que todos las cámaras tienen el mismo ángulo de visión, definimos lo siguiente:
  Dado un ángulo α tal que $0 \leq \alpha \leq \pi$, una fórmula $\varphi \in$ BBL es α-realizable si hay una estructura geométrica de Kripke M = (pos, ang, D, R, ~) donde ang(a) = α para cada uno a $\in$ Agt, que satisface ϕ .

  Considerando la fórmula:

  $$\mathbf{See\,(a\,,b\,,c\,) = a \triangleright b \wedge a \triangleright c \wedge b \triangleright a \wedge b \triangleright c \wedge c \triangleright a \wedge c \triangleright b}$$



  Considerando que las dos cámaras se ven entre sí, sólo puede satisfacerse esto en un plano euclidiano, en el cual las cámaras están sumando $\boldsymbol{\pi}$ y debido a que son triángulos de visión queda tal que $\boldsymbol{\alpha \geq \pi/3}$.

  Para esto último se considera la fórmula:

  $$\mathrm{See}(\mathrm{a}_1, \ldots, \mathrm{a}_n) = \bigwedge_{k=1}^{n} \bigwedge_{i=1, i \neq k}^{n} \mathrm{a}_k \triangleright \mathrm{a}_i$$

  Sacamos en conclusión que para que todos los agentes del conjunto ,{a1, ..., an}, puedan ver a todos los demás. Para que esa fórmula sea satisfechos en el plano euclidiano, todos los agentes en ese conjunto deben estar dispuestos en los vértices de un polígono convexo y ver el polígono completo, de ahí la suma de los ángulos de sus cámaras de la visión debe ser al menos **(n - 2) $\boldsymbol{\pi}$**. En particular, Ver (a 1, ..., a n) es α-realizable si y sólo si $\boldsymbol{\alpha \geq (n-2)\,\pi\,/\,n}$. Pero falla en estructuras de Kripke que tenga las cámaras un mismo ángulo que es al menos de **(n−2)$\boldsymbol{\pi}$/n**. Y válido para aquellas que no superan esto, son estrictamente por debajo.

- **Anuncios públicos:** en el framework de BBL, los agentes saben lo que perciben pero es necesario aportar información adicional para lograr su misión de vigilancia. Por ejemplo, si están reconociendo a varios objetivos, es posible que deban informarse entre sí de dónde ven realmente.

Forma de BBL similar a PDL

En este apartado tomaremos un punto de vista distinto usando un razonamiento PDL (Propositional Dynamic Logic) en el cual veremos cada secuencia de giros de cámara como un programa.

Todos los operadores de conocimiento contenidos en BBL pueden ser representados en términos de programa y además , ser traducidos a versiones PDL.

El lenguaje se divide en 2 partes, el BBL $_{PDL*}$, y el BBL $_{PDL.}$. El primero se encarga de definir el grupo acciones y fórmulas que podremos utilizar. Estas se dividen en dos conjuntos, φ y γ. Siendo el primero usado para fórmulas lógicas y el segundo para acciones del programa.
Un ejemplo de estos conjuntos es el siguiente:
$\varphi ::= a \triangleright b \mid \neg \phi \mid \phi \vee \phi \mid K_a\varphi \mid [\gamma]\varphi \mid [\varphi!]\varphi; \quad \gamma ::= a \mid \varphi? \mid \gamma ; \gamma \mid \gamma \cup \gamma \mid \gamma^*$

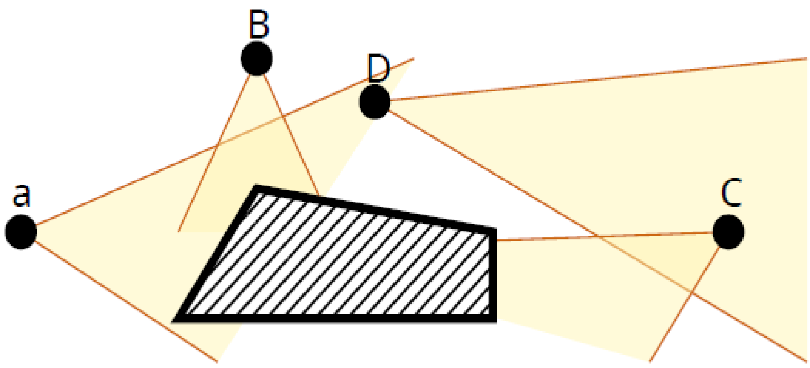
El BBL $_{PDL.}$ por su parte se encarga de utilizar lo que se especifica en BBL $_{PDL*}$ para realizar ejemplos concretos. Pa probar esto usaremos el siguiente enunciado como ejemplo sencillo: “Si **a** no ve a **b** entonces a se gira” que sería representado como: $\gamma = (\neg a \triangleright b)?; \overset{\curvearrowright}{a}$.

Ampliaciones de BBL

En esta sección vamos a hablar de distintos escenarios que se nos pueden presentar con la aplicación de la lógica BBL.

- **Agregar obstáculos**

En casos reales cotidianos, la visibilidad de las cámaras de los agentes (u objetos en casos reales) se puede ver afectada por obstáculos fijos o en movimiento.



En los modelos geométricos, estos se representan por sus formas colocadas en el plano.

Representaremos los obstáculos como un polígono, descrito formalmente por una lista que contiene sus vértices. Entonces, diremos que un segmento [pos(a), pos(b)] que va desde la posición del agente **a** hasta la posición del agente **b**, tiene intersección con el obstáculo si el segmento cruza por uno de los lados del polígono. Comprobar si esto ocurre o no, tiene una complejidad de tiempo polinómico. Además, el procedimiento de verificación del modelo en concreto se puede ajustar para hacer frente a los obstáculos, dando como resultado que el problema extendido de verificación del modelo se ejecute en espacio **PSPACE**.

Cuando agregamos obstáculos no podemos asumir automáticamente que los agentes conocen la posición de los demás agentes. Los algoritmos de comprobación de modelos y prueba de satisfacción se modifican en consecuencia.

- **<u>Cámaras en 3D</u>**

En casos reales, casi siempre tenemos cámaras en espacios tridimensionales. Esto agrega una sobrecarga técnica esencial, pero no crucial, sobre el cálculo de las áreas de visión de las cámaras. No obstante, los lenguajes lógicos y las partes epistémicas de su semántica permanecen inalteradas.
Los algoritmos de comprobación de modelos y prueba de satisfacción estarán en las mismas clases de complejidad que en los casos 2D.

- **<u>Mover cámaras y robots</u>**

En la vida real, los agentes y sus cámaras suelen ser móviles, pero también pueden ser fijos. La forma más sencilla de adaptarnos a estas situaciones es considerar los mismos lenguajes lógicos sobre modelos geométricos que involucran agentes con sus ángulos de visión pero no sus posiciones. Por lo tanto, para el problema de verificación de modelo para el caso de agentes móviles se reduce a un problema de prueba de satisfacción para modelos con cámaras fijas. Los mundos (estados) posibles en las estructuras geométricas de Kripke siguen siendo los mismos. Se basan en todos los conjuntos de otros agentes que se pueden ver de un vistazo después de cualquier rotación de cámara.

Bibliografía

- Gasquet, O., Goranko, V. & Schwarzentruber, F. Big Brother Logic: visual-epistemic reasoning in stationary multi-agent systems. Auto Agent Multi-Agent Syst 30, 793–825 (2016). https://doi.org/10.1007/s10458-015-9306-4

- Artículo. (s. f.). ResearchGate. https://www.researchgate.net/publication/282546747_Big_Brother_Logic_visual-epistemic_reasoning_in_stationary_multi-agent_systems