

Sistemas Multi-Agentes

AgentSpeak(L)

Dr. Alejandro Guerra-Hernández

Instituto de Investigaciones en Inteligencia Artificial
Universidad Veracruzana

*Campus Sur, Calle Paseo Lote II, Sección Segunda No 112,
Nuevo Xalapa, Xalapa, Ver., México 91097*

`mailto:aguerra@uv.mx`
`https://www.uv.mx/personal/aguerra/sma`

Maestría en Inteligencia Artificial 2024



Universidad Veracruzana

Problemas de las Lógicas BDI

- ▶ Las Lógicas BDI nos permiten razonar acerca de nuestros agentes racionales, permitiendo especificar y verificar el comportamiento de estos sistemas.
- ▶ Sus componentes modales Intencionales, temporales y de acción, son lo suficientemente **expresivos** como para abordar creencias, deseos, intenciones y la toma de decisión en esos términos.
- ▶ La **complejidad computacional** de su teoría de prueba es elevada, aunque no mayor que el de su componente temporal; y la evidencia clara de su completitud fue tardía [9].



Solución

- ▶ Analizar de dMARS, para formalizar su **semántica operacional**.
- ▶ *AgentSpeak(L)*, un lenguaje de programación basado en una lógica restringida de primer orden con eventos y acciones.
- ▶ El comportamiento de un agente está dado por su **programa** escrito en este lenguaje.
- ▶ Los operadores BDI **no son** expresiones modales, pero pueden verse como tales desde una postura Intencional:
 - ▶ El modelo del agente, ambiente y otros agentes, constituye las **creencias** del agente.
 - ▶ Los estados a los que un agente quiere llegar, con base en sus estímulos internos y externos, son sus **deseos**.
 - ▶ Y los planes que el agente adopta son sus **Intenciones**.



Alfabeto

- ▶ Se asumen conjuntos ilimitados de **símbolos** de:

Var Variables

Func Funciones

Pred Predicados

Actn Acciones

Const \subseteq *Func* Constantes

- ▶ Los **conectivos lógicos**: \neg , \wedge y \sim .
- ▶ Los **símbolos especiales**: $\{!, ?, +, -, :, ;, \leftarrow, :-, \top\}$.
- ▶ Se adoptará un estilo *à la Prolog* [2].



Término

- ▶ El conjunto de **términos** en $AgentSpeak(L)$ incluye las variables, las constantes y los términos compuestos:

$$t ::= X \mid c \mid f(t_1, \dots, t_n) \quad (n \geq 1) \quad (1)$$

donde $X \in Var$ es una variable; $c \in Const$, una función de aridad 0, es una constante; y $f \in Func$, de aridad $n \geq 1$, es una función aplicada a $t_{1 \leq i \leq n}$ términos.

- ▶ **Ejemplos:**
 - ▶ *Nombre*, X , Y , ...
 - ▶ *alejandro*, *guerra*, $1, 2, \dots$
 - ▶ *apellido(Nombre)*, *apellido(alejandro)*, *suma(1, 2)*, etc.



Átomo

- ▶ El conjunto de **átomos** en $AgentSpeak(L)$ incluye las proposiciones y los predicados, con anotaciones o sin ellas:

$$at ::= p(t_1, \dots, t_n) \quad (n \geq 0) \quad (2)$$

$$| p(t_1, \dots, t_n)[s_1, \dots, s_m] \quad (n \geq 0, m \geq 1) \quad (3)$$

$$s ::= \text{percept} \mid \text{self} \mid \text{id} \quad (4)$$

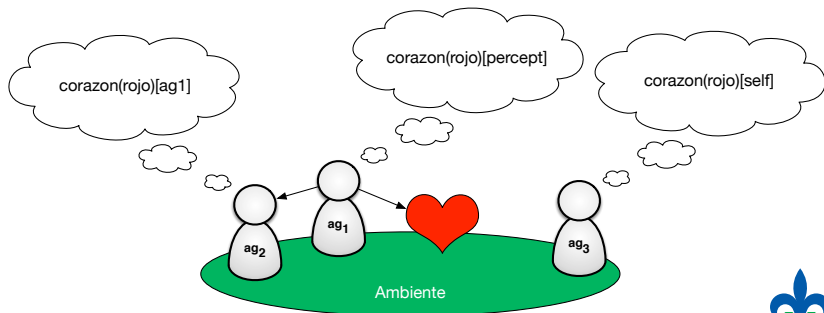
donde $p \in Pred$, tal que si $n = 0$, es una proposición atómica; si $n > 0$, es un predicado de aridad n , aplicado a $t_{1 \leq i \leq n}$ términos. Los átomos pueden adornarse con al menos una anotación s , que indica su fuente de origen.

- ▶ **Ejemplos:** $hoyEsLunes$, $mayorQue(\text{suma}(3, 4), 5)$



Fuentes

- Las **fuentes** de un átomo en *AgentSpeak(L)* pueden ser la percepción (ag_1), la comunicación (ag_2) o el razonamiento del propio agente (ag_3).



Términos compuestos vs Átomos

- ▶ Un término compuesto **denota un elemento** del Universo de Discurso (\mathcal{U}), ya sea porque:
 - ▶ Es una función del estilo $f : \mathcal{U}^n \mapsto \mathcal{U}$;
 - ▶ Es una estructura de datos del estilo $f : \mathcal{U}^n$.
- ▶ Un átomo **denota una relación** entre elementos del universo de discurso, son predicados del estilo $p : \mathcal{U}^n \mapsto Bool$.
- ▶ **Ejemplos:** Si asumimos que $\mathcal{U} = \mathbb{N}$
 - ▶ La función $suma(4, 3) \mapsto 7$ es un término;
 - ▶ La función $cons(1, cons(2, [])) \mapsto [1, 2]$ es un término.
 - ▶ El predicado $mayorQue(4, 3) \mapsto T$ es un átomo;
 - ▶ El predicado $vacía([1, 2]) \mapsto F$ es un átomo.

Literal

- ▶ Un átomo o su **negación fuerte** son una **literal**.
- ▶ La negación se conoce como **literal negativa** y el átomo no negado como **literal positiva**:

$$lit ::= at \mid \sim at \quad (5)$$

- ▶ **Ejemplos:**

- ▶ $mayorQue(3, 2), vacia([])$
- ▶ $\sim mayorQue(2, 3), \sim vacia([1, 2])$

Fórmula Lógica bien Formada

- ▶ Las **fórmulas lógicas bien formadas** (*fbf*) de *AgentSpeak(L)* incluyen los átomos, su negación y su conjunción:

$$fbf ::= lit \mid \neg fbf \mid fbf \wedge fbf \quad (6)$$

donde *lit* es una literal; \neg denota la negación débil y \wedge la conjunción.

- ▶ **Ejemplos:**

- ▶ $\sim negativo(X) \wedge mayorQue(X, 99)$
- ▶ $\neg negativo(X) \wedge mayorQue(X, 99)$



Programa de agente

- ▶ En *AgentSpeak(L)*, un **programa de agente** es un conjunto de creencias (*bs*), planes (*ps*) y metas (*gs*):

$$ag ::= bs \quad ps \quad gs \quad (7)$$

Creencia

- ▶ Una **creencia** en *AgentSpeak(L)* es una literal de base o una regla:

$$b ::= lit \mid rule \quad (8)$$

$$rule ::= lit_1 :- fbf \quad (9)$$

$$bs ::= b_1, \dots, b_n \quad (n \geq 0) \quad (10)$$

tal que *lit* no tiene variables sin instanciar como argumentos.

- ▶ **Ejemplos:**

- ▶ *factorial(1, 1)*

- ▶ *factorial(X, N) :- factorial(N - 1, Aux) ∧ N = X * Aux*



Planes

- ▶ Un **plan** en *AgentSpeak(L)* es una estructura compuesta por un evento disparador (*te*), un contexto (*ct*) y un cuerpo (*h*):

$$p ::= te : ct \leftarrow h. \quad (11)$$

$$ps ::= p_1, \dots, p_n \quad (n \geq 1) \quad (12)$$

- ▶ Ejemplo:

$$\begin{aligned}
 +!ir(paris) & : \text{ahorros}(X) \wedge X > 4000 \\
 \leftarrow & !reserva(hotel, paris), !reserva(vuelo, paris).
 \end{aligned}$$

Evento disparador de un plan

- ▶ El **evento disparador** de un plan en *AgentSpeak(L)* consiste en agregar (+), o eliminar (-), una creencia o una meta:

$$te ::= +b \mid -b \mid +g \mid -g \quad (13)$$

- ▶ Ejemplo:

$+!ir(paris) : ahorros(X) \wedge X > 4000$
 $\leftarrow !reserva(hotel, paris), !reserva(vuelo, paris).$

Contexto de un plan

- ▶ El **contexto** de un plan en $AgentSpeak(L)$ puede ser vacío o una fórmula lógica bien formada:

$$ct ::= \top \mid fbf \quad (14)$$

- ▶ Ejemplo:

$+!ir(paris) : ahorros(X) \wedge X > 4000$
 $\leftarrow !reserva(hotel, paris), !reserva(vuelo, paris).$

Cuerpo de un plan

- ▶ El **cuerpo** de un plan es una secuencia, posiblemente vacía, de acciones, actualizaciones de creencias y/o metas:

$$h ::= h1; \top \mid \top \quad (15)$$

$$h1 ::= a \mid u \mid g \mid h1; h1 \quad (16)$$

- ▶ Ejemplo:

+!ir(*paris*) : ahorros(*X*) \wedge *X* > 4000

← !reserva(*hotel, paris*), !reserva(*vuelo, paris*).



Acción

- ▶ Las **acciones** en *AgentSpeak(L)* son llamadas a procedimientos:

$$a ::= A(t_1, \dots, t_n) \quad (n \geq 0) \quad (17)$$

donde $A \in Act_n$ es una acción aplicada a $t_{1 \leq i \leq n}$ términos.

- ▶ **Ejemplos:**
 - ▶ `paga(hotel, 180)`
 - ▶ `agenda(paris,01/07/2017,13/07/2017)`
 - ▶ `.print("Reservación lista")`

Actualización de creencias

- ▶ Las **actualizaciones de creencias** en *AgentSpeak(L)* consisten en agregar una nueva creencia o eliminar aquellas que unifican con una literal dada:

$$u ::= +b \mid -lit \quad (18)$$

- ▶ Ejemplos:
 - ▶ +reservado(hotel,paris)
 - ▶ -curso(Curso,01/07/2017,13/07/2017)



Meta

- ▶ Las metas en *AgentSpeak(L)* pueden ser **alcanzables** (!) o **verificables** (?):

$$g ::= !lit \mid ?lit \quad (19)$$

$$gs ::= g_1, \dots, g_n \quad (n \geq 0) \quad (20)$$

- ▶ Ejemplos:

- ▶ !ir(paris)
- ▶ !reserva(hotel,paris)
- ▶ ?reservado(vuelo,paris)
- ▶ ?agenda(paris,Inicio,Fin)

Gramática BNF

$$\begin{array}{ll}
 ag & ::= bs \ ps \ gs \\
 bs & ::= b_1, \dots, b_n \quad (n \geq 0) \\
 b & ::= lit \mid rule \quad ground(lit) \\
 lit & ::= at \mid \sim at \\
 at & ::= p(t_1, \dots, t_n) \quad (p \in Pred, n \geq 0) \\
 & \quad \mid p(t_1, \dots, t_n)[s_1, \dots, s_m] \quad (p \in Pred, n \geq 0, m > 0) \\
 s & ::= percept \mid self \mid id \\
 rule & ::= lit :- fbf \\
 fbf & ::= lit \mid \neg fbf \mid fbf \wedge fbf \mid fbf \vee fbf \\
 t & ::= X \mid f \mid f(t_1, \dots, t_n) \quad (X \in Var, f \in Func, n \geq 1) \\
 ps & ::= p_1, \dots, p_n \quad (n \geq 1) \\
 p & ::= te : ct \leftarrow h \\
 te & ::= +lit \mid -lit \mid +g \mid -g \\
 ct & ::= fbf \mid \top \\
 h & ::= h_1; \top \mid \top \\
 h_1 & ::= a \mid u \mid g \mid h_1; h_1 \\
 a & ::= ac(t_1, \dots, t_n) \quad (ac \in Actn, n \geq 0) \\
 u & ::= +b \mid -lit \\
 g & ::= !lit \mid ?lit \\
 gs & ::= g_1, \dots, g_n \quad (n \geq 0)
 \end{array}$$

Programa *AgentSpeak(L)*

- ▶ El siguiente código *AgentSpeak(L)*:

```

1 // Agente cap05/src/asl/beto.asl
2
3 /* creencias iniciales y reglas */
4 factorial(1,1).
5 factorial(Num,Fact) :-
6     Fact = Num * FactAux & factorial(Num-1,FactAux).
7
8 /* metas iniciales */
9 !start(5).
10
11 /* planes */
12 +!start(X) : X>0 <-
13     ?factorial(X,F);
14     .print("El factorial de ", X, " es ",F,".").

```

es un programa válido, con salida:

```

1 [beto] El factorial de 5 es 120.

```



Pila

- ▶ La expresión $p = [e_1 \ddagger e_2 \ddagger \dots \ddagger e_n]$ denota una **pila** p de tamaño n . La pila vacía se denota como \emptyset .
- ▶ Se definen las siguientes operaciones sobre las pilas y sus elementos:
 - ▶ $top(p) = e_1$.
 - ▶ $pop(p) = e_1$ y $p = [e_2 \ddagger \dots \ddagger e_n]$.
 - ▶ $push(e, p) = [e \ddagger e_1 \ddagger e_2 \ddagger \dots \ddagger e_n]$.



Cola

- ▶ La expresión $c = \{e_1, e_2, \dots, e_n\}$ denota una **cola** c de tamaño n . La cola vacía se denota como \emptyset .
- ▶ Se definen las siguientes operaciones sobre las colas y sus elementos:
 - ▶ $first(c) = e_1$.
 - ▶ $pop(c) = e_1$ y $c = \{e_2, \dots, e_n\}$.
 - ▶ $push(e, c) = \{e_1, e_2, \dots, e_n, e\}$.

Semánticas operacionales

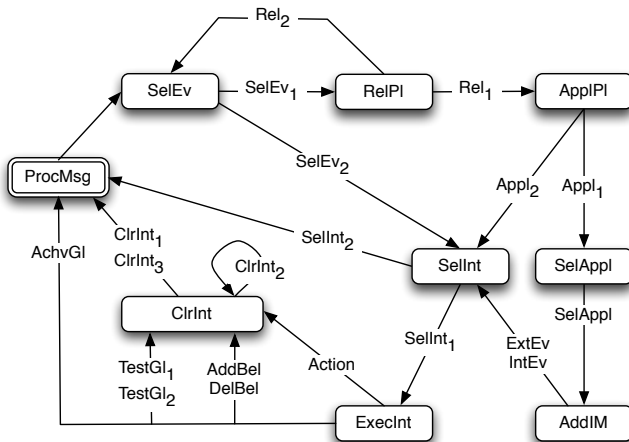
- ▶ La ejecución de un programa de agente *AgentSpeak(L)* está determinado por una **semántica operacional**, al estilo de las propuestas por Plotkin [8].
- ▶ Estas semánticas se basan en un sistema de transiciones entre configuraciones de un programa.

Sistema de transiciones

- ▶ Un **sistema de transiciones** es una estructura $\langle \Gamma, \rightarrow \rangle$, donde Γ es un conjunto de elementos γ , conocidos como **configuraciones**, y \rightarrow es una relación binaria sobre Γ , llamada **transición**.



El sistema de transiciones de *AgentSpeak(L)*



Configuraciones *AgentSpeak(L)* I

- ▶ Una **configuración** $\gamma = \langle ag, C, M, T, s \rangle$ está compuesta por:
 - ▶ Un **programa** del agente $ag = \langle bs, ps, gs \rangle$.
 - ▶ Una **circunstancia** del agente C es una tupla $\langle I, E, A \rangle$ donde:
 - ▶ I es el conjunto de **intenciones** $\{i_1, i_2, \dots, i_n\}$, t.q. cada $i \in I$ es una pila de planes parcialmente instanciados;
 - ▶ E es una cola de **eventos** $\{\langle te_1, i_1 \rangle, \langle te_2, i_2 \rangle, \dots, \langle te_n, i_n \rangle\}$, t.q. cada te es un evento disparador y cada i es una intención. Si $i = \top$ se dice que el evento es externo; en caso contrario el evento es interno, es decir, generado por una intención previa; y
 - ▶ A es el conjunto de **acciones** a ser ejecutadas por el agente en el ambiente.
 - ▶ El **buzón** M es una tupla $\langle In, Out, SI \rangle$ donde:
 - ▶ In es el buzón en entrada del agente;
 - ▶ Out es el buzón de salida; y
 - ▶ SI es un registro de intenciones suspendidas, i.e., aquellas que esperan un mensaje de respuesta para reanudarse.



Configuraciones *AgentSpeak(L)* II

- ▶ T es una tupla de registros $\langle R, Ap, \iota, \epsilon, \rho \rangle$ donde R es el conjunto de planes relevantes dado cierto evento; $Ap \subseteq R$ es el conjunto de planes aplicables, aquellos que el agente cree poder ejecutar; ι , ϵ y ρ son, respectivamente, la intención, el evento, y el plan actualmente considerados en el razonamiento del agente.
- ▶ $s \in \{SelEv, RelPI, AppPI, SelAppl, Sellnt, AddIM, Execlnt, Clrint, ProcMsg\}$ denota una etiqueta para cada estado de la configuración.



Transiciones en *AgentSpeak(L)*

- ▶ La relación de transición \rightarrow se definen en términos de un conjunto de **reglas semánticas** con la forma:

$$\text{(regla)} \quad \frac{cond}{\gamma \rightarrow \gamma'}$$

donde la configuración γ puede transformarse en una nueva configuración γ' , si las condiciones expresadas en *cond* se cumplen.



Notación

- ▶ Para hacer referencia al componente E de una circunstancia C , escribimos C_E . Ej. Las creencias de un agente se denotan como ag_{bs} .
- ▶ Para indicar que no hay ninguna intención siendo considerada en la ejecución del agente, se emplea $T_\perp = \top$. De forma similar para T_ϵ , T_ρ y demás registros de una configuración.
- ▶ Se usa $i[p]$ para denotar que p es el plan en el tope de la intención i ; también se puede usar $p = top(i)$.
- ▶ Si asumimos que p es un plan de la forma $te : ct \leftarrow h$, entonces:
 - ▶ $TrEv(p) = te$ denota el evento disparador de p .
 - ▶ $Ct(p) = ct$ denota el contexto de p .
 - ▶ $Head(p) = te : ct$ denota la cabeza de p .
 - ▶ $Body(p) = h$ denota el cuerpo de p .

Funciones de selección

- ▶ Se definen las siguientes funciones de **selección** de eventos $S_E = pop(C_E)$, planes aplicables $S_{Ap} = pop(T_{Ap})$ e intenciones $S_I = pop(C_I)$.
- ▶ Observen que estas funciones de selección son **destruktivas**, en el sentido que el elemento seleccionado es eliminado de la pila, cola o conjunto donde se encontraba.
- ▶ Estas funciones pueden **redefinirse** para una selección más informada.

Substituciones

- ▶ Una **substitución** es un conjunto finito de la forma:

$$\theta = \{t_1/v_1, \dots, t_n/v_n\}, \quad (n \geq 0)$$

donde las v_i son variables únicas y cada t_i es un término que no incluye a v_i .

- ▶ La forma t_i/v_i se conoce como **ligadura**.
- ▶ La sustitución vacía se denota por ϵ , y se conoce también como **substitución identidad**.



Ejemplos

- ▶ Los siguientes conjuntos son substituciones válidas:
 - ▶ $\theta = \{f(Y)/X, Z/Y\}$
 - ▶ $\theta' = \{a/X, b/Y, Y/Z\}$
- ▶ En cambio, las siguientes no lo son:
 - ▶ $\theta = \{f(X)/X, Z/Y\}$
 - ▶ $\theta' = \{a/X, b/Y, Y/X\}$

Aplicación de una substitución

- ▶ La **aplicación** de la substitución θ al término t se denota $t\theta$ y resulta en un nuevo término donde todas las ligaduras de la substitución se han llevado a cabo en t de forma paralela.
- ▶ **Ejemplos:** Sea la substitución $\theta = \{5/Num, F/Fact\}$ y el término $t = factorial(Num, Fact)$, $t\theta = factorial(5, F)$. En tanto que, $t\epsilon = factorial(Num, Fact)$, de ahí el nombre de substitución identidad.

Composición de sustituciones

- ▶ La **composición** de dos sustituciones, $\theta = \{t_1/v_1, \dots, t_n/v_n\}$ y $\theta' = \{t'_1/v'_1, \dots, t'_m/v'_m\}$, es a su vez una sustitución, definida de la siguiente manera:

$$\theta \circ \theta' = \{t_1\theta'/v_1, \dots, t_n\theta'/v_n\} \cup \{t'_i/v'_i \in \theta' \mid v'_i \text{ no ocurre en } \theta\}$$

Eliminando todas las ligaduras de la forma t/t .

- ▶ **Ejemplo:** Sean $\theta = \{g(X, Y)/W\}$ y $\theta' = \{a/X, b/Y, c/Z\}$ dos sustituciones, la composición $\theta \circ \theta' = \{g(a, b)/W, a/X, b/Y, c/Z\}$.

Unificador Más General (UMG)

- ▶ Sean t_1 y t_2 términos. Se dice que la sustitución θ es un **unificador** de esos términos si $t_1\theta = t_2\theta$. Una sustitución θ se dice más general que una sustitución σ , si y sólo si existe una sustitución τ tal que $\sigma = \theta \circ \tau$. Un unificador θ se dice el **unificador más general (umg)** de dos términos, si y sólo si es más general que cualquier otro unificador entre esos términos.



Ejemplo

- ▶ Consideren los términos $f(X)$ y $f(g(Y))$, el *umg* de ellos es $\theta = \{g(Y)/X\}$, pero existen otros muchos unificadores, por ejemplo $\{g(a)/X, a/Y\}$. Intuitivamente, el *umg* de dos términos es el más simple de todos sus unificadores. Observen que $f(X)\theta = f(g(Y))\theta = f(g(Y))$.

Unificación y anotaciones

- ▶ Los términos de $AgentSpeak(L)$ están **anotados** y esto debe tomarse en cuenta al computar la unificación.
- ▶ En general, dos términos $t_1[a_1, \dots, a_n]$ y $t_2[a'_1, \dots, a'_m]$ unifican, si y sólo si (i) existe un umg θ tal que $t_1\theta = t_2\theta$ y (ii) $\{a_1, \dots, a_n\} \subseteq \{a'_1, \dots, a'_m\}$. Esto es, si las fuentes de t_1 están incluidas en las de t_2 .

Ejemplos

- ▶ $umg(p(c)[s_1], p(c)[s_1, s_2]) = \top$ debido a que no hay variables ligadas y las anotaciones del primer término están incluidas en las del segundo;
- ▶ $umg(p(c)[a_1], p(c)[a_2])$ falla debido a que la anotaciones del primer término no están incluidas en las anotaciones del segundo.
- ▶ $umg(p(X)[a_2], p(c)[a_1, a_2, a_3]) = \{c/X\}$. Ahora el unificador incluye una ligadura.

Consecuencia lógica

- ▶ La expresión $ag_{bs} \models fbf$ denota que la fórmula bien formada fbf es **consecuencia lógica** de las creencias del agente ag . Siguiendo la definición sintáctica de una fbf (Ecuación 6), tenemos que:
 - ▶ $ag_{bs} \models lit[s_1, \dots, s_n]$, si y sólo si, existe una literal $lit'[s'_1, \dots, s'_m] \in ag_{bs}$, tal que $lit\theta = lit'$. θ se conoce como unificador de respuesta.
 - ▶ $ag_{bs} \models \neg fbf$, si y sólo si, $ag_{bs} \not\models fbf$, tal que $\theta = \epsilon$.
 - ▶ $ag_{bs} \models fbf_1 \wedge fbf_2$ θ , si y sólo si, $ag_{bs} \models fbf_1$ θ_1 y $ag_{bs} \models fbf_2$ θ_2 . El unificador de respuesta $\theta = \theta_1 \circ \theta_2$.

Reglas y consecuencia lógica

- ▶ Si consideramos a las reglas como parte de las creencias, debemos agregar el siguiente caso:
 - ▶ $ag_{bs} \models lit[s_1, \dots, s_n]$, si y sólo si, en las creencias del agente ag , existe una regla $lit'[s'_1, \dots, s'_n] :- fbf$ tal que: (i) $lit\theta' = lit'\theta'$ y (ii) $ag_{bs} \models fbf$ con sustitución de respuesta θ'' . $\theta = \theta' \circ \theta''$.

Ejemplos

- ▶ $p(X)[a_1]$ se sigue de $ag_{bs} = \{p(t)[a_1, a_2]\}$; pero
- ▶ $p(X)[a_1, a_2]$ no se sigue de $ag_{bs} = \{p(t)[a_1]\}$.



Substitución de respuesta

- ▶ La **substitución de respuesta** para una *fbf*, es un unificador θ tal que la aplicación de θ a la *fbf*, se sigue de las creencias del agente.
Formalmente:

$$\text{SubstResp}(fbf) = \theta \mid ag_{bs} \models fbf\theta \quad (21)$$

Planes relevantes

- ▶ El conjunto de los **planes relevantes** (R) dado un evento $\langle te, i \rangle \in C_E$, está compuesto por los planes cuyo evento disparador unifica con te :

$$PRels(te) = \{p\theta \mid p \in ag_{ps} \wedge \theta = umg(te, TrEv(p))\} \quad (22)$$

- ▶ **Ejemplo:** Dado el evento $\langle +!ir(mexico), \top \rangle$, los siguientes planes son relevantes:

```

1  +!ir(Dest) : autobus <-
2     !reservar(autobus, Dest);
3     !reservar(hotel, Dest).
4
5  +!ir(Dest) : avion <-
6     !reservar(avion, Dest);
7     !reservar(hotel, Dest).

```



Planes aplicables

- ▶ El conjunto de los **planes aplicables** ($Ap \subseteq R$) incluye aquellos cuyo contexto es consecuencia lógica de las creencias del agente.

$$PApls(PRels) = \{p\theta \mid p \in PRels \wedge ag_{bs} \models Ct(p)\theta\} \quad (23)$$

- ▶ **Ejemplo:** Dado el evento $\langle +!ir(mexico), \top \rangle$ y que $ag_{bs} \models avion$, solo el segundo plan es aplicable:

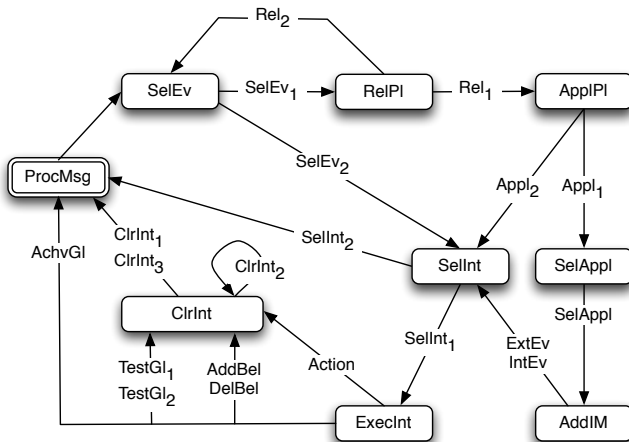
```

1  +!ir(Dest) : autobus <-
2    !reservar(autobus, Dest);
3    !reservar(hotel, Dest).
4
5  +!ir(Dest) : avion <-
6    !reservar(avion, Dest);
7    !reservar(hotel, Dest).

```



El sistema de transiciones de *AgentSpeak(L)*



Selección de eventos

$$(\text{SelEv}_1) \frac{C_E \neq \emptyset, S_E = \langle te, i \rangle}{\langle ag, C, M, T, \text{SelEv} \rangle \rightarrow \langle ag, C', M, T', \text{RelPl} \rangle} \quad (24)$$

$$\text{t.q. } T'_\epsilon = \langle te, i \rangle, C'_E = C_E \setminus T'_\epsilon,$$

$$(\text{SelEv}_2) \frac{C_E = \emptyset}{\langle ag, C, M, T, \text{SelEv} \rangle \rightarrow \langle ag, C, M, T, \text{Sellnt} \rangle} \quad (25)$$



Planes relevantes

$$(Rel_1) \frac{T_\epsilon = \langle te, i \rangle, PRels(te) \neq \emptyset}{\langle ag, C, M, T, RelPI \rangle \rightarrow \langle ag, C, M, T', AppPI \rangle} \quad (26)$$

$$\text{t.q. } T'_R = PRels(te)$$

$$(Rel_2) \frac{T_\epsilon = \langle te, i \rangle, PRels(te) = \emptyset}{\langle ag, C, M, T, RelPI \rangle \rightarrow \langle ag, C, M, T, SelEv \rangle} \quad (27)$$



Planes aplicables

$$(Appl_1) \quad \frac{PApls(T_R) \neq \emptyset}{\langle ag, C, M, T, ApplPI \rangle \rightarrow \langle ag, C, M, T', SelAppl \rangle} \quad (28)$$

$$\text{t.q. } T'_{Ap} = PApls(T_R)$$

$$(Appl_2) \quad \frac{PApls(T_R) = \emptyset}{\langle ag, C, M, T, ApplPI \rangle \rightarrow \langle ag, C, M, T, Sellnt \rangle} \quad (29)$$



Selección plan aplicable

$$(\text{SelAppl}) \frac{S_{Ap} = (p, \theta)}{\langle ag, C, M, T, \text{SelAppl} \rangle \rightarrow \langle ag, C, M, T', \text{AddIM} \rangle} \quad (30)$$

$$\text{t.q. } T'_\rho = (p, \theta)$$

Adopción de intenciones

$$(ExtEv) \frac{T_\epsilon = \langle te, \top \rangle, T_\rho = (p, \theta)}{\langle ag, C, M, T, AddIM \rangle \rightarrow \langle ag, C', M, T, SellInt \rangle} \quad (31)$$

$$\text{t.q. } C'_I = C_I \cup \{[p\theta]\}$$

$$(IntEv) \frac{T_\epsilon = \langle te, i \rangle, T_\rho = (p, \theta)}{\langle ag, C, M, T, AddIM \rangle \rightarrow \langle ag, C', M, T, SellInt \rangle} \quad (32)$$

$$\text{t.q. } C'_I = C_I \cup \{i[p\theta]\}$$



Selección de intenciones

$$(SellInt_1) \frac{C_I \neq \emptyset, S_I = i}{\langle ag, C, M, T, SellInt \rangle \rightarrow \langle ag, C, M, T', ExecInt \rangle} \quad (33)$$

t.q. $T'_l = i$

$$(SellInt_2) \frac{C_I = \emptyset}{\langle ag, C, M, T, SellInt \rangle \rightarrow \langle ag, C, M, T, ProcMsg \rangle} \quad (34)$$

Ejecución de una acción

$$\begin{aligned}
 \text{(Action)} \quad & \frac{T_l = i[\text{head} \leftarrow a; h]}{\langle ag, C, M, T, \text{ExecInt} \rangle \rightarrow \langle ag, C', M, T', \text{ClrInt} \rangle} \\
 & \text{t.q. } C'_A = C_A \cup \{a\}, T'_l = i[\text{head} \leftarrow h], \\
 & C'_I = (C_I \setminus \{T_l\}) \cup \{T'_l\}
 \end{aligned}
 \tag{35}$$

Ejecución de una meta alcanzable

$$\begin{array}{c}
 \text{(AchvGI)} \quad \frac{T_l = i[\text{head} \leftarrow !at; h]}{\langle ag, C, M, T, \text{ExecInt} \rangle \rightarrow \langle ag, C', M, T, \text{ProcMsg} \rangle} \\
 \text{t.q. } C'_E = C_E \cup \{ \langle +!at, T_l \rangle \}, \\
 C'_I = C_I \setminus \{ T_l \}
 \end{array} \tag{36}$$



Ejecución de una meta verificable

$$(\text{TestGl}_1) \frac{T_l = i[\text{head} \leftarrow ?at; h], \text{Test}(ag_{bs}, at) = at\theta}{\langle ag, C, M, T, \text{ExecInt} \rangle \rightarrow \langle ag, C', M, T, \text{Clrlnt} \rangle} \quad (37)$$

$$\text{t.q. } T'_l = i[(\text{head} \leftarrow h)\theta], \\ C'_l = (C_l \setminus \{T_l\}) \cup \{T'_l\}$$

$$(\text{TestGl}_2) \frac{T_l = i[\text{head} \leftarrow ?at; h], \text{Test}(ag_{bs}, at) = \perp}{\langle ag, C, M, T, \text{ExecInt} \rangle \rightarrow \langle ag, C', M, T, \text{Clrlnt} \rangle} \quad (38)$$

$$\text{t.q. } C'_E = C_E \cup \{\langle -?at, T_l \rangle\}, \\ C'_l = C_l \setminus \{T_l\}$$



Actualización de creencias: Agregar

$$(AddBel) \frac{T_l = i[head \leftarrow +b; h]}{\langle ag, C, M, T, ExecInt \rangle \rightarrow \langle ag', C', M, T', ClrInt \rangle}$$

$$\begin{aligned} \text{t.q. } ag'_{bs} &= ag_{bs} + b_{[self]}, & (39) \\ C'_E &= C_E \cup \{ \{ +b_{[self]}, T \} \}, \\ T'_l &= i[head \leftarrow h], \\ C'_i &= (C_i \setminus \{ T_l \}) \cup \{ T'_l \} \end{aligned}$$



Actualización de creencias: Borrar

$$\text{(DelBel)} \quad \frac{T_L = i[\text{head} \leftarrow -at; h]}{\langle ag, C, M, T, ExecInt \rangle \rightarrow \langle ag', C', M, T', ClrInt \rangle}$$

$$\begin{aligned} \text{t.q. } ag'_{bs} &= ag_{bs} - at_{[self]}, & (40) \\ C'_E &= C_E \cup \{ \langle -at_{[self]}, \top \rangle \}, \\ T'_L &= i[\text{head} \leftarrow h], \\ C'_I &= (C_I \setminus \{ T_L \}) \cup \{ T'_L \} \end{aligned}$$



Limpieza I

$$(Clrlnt_1) \frac{T_l = [head \leftarrow \top]}{\langle ag, C, M, T, Clrlnt \rangle \rightarrow \langle ag, C', M, T, ProcMsg \rangle} \quad (41)$$

$$\text{t.q. } C'_l = C_l \setminus \{T_l\}$$

$$(Clrlnt_2) \frac{T_l = i[head \leftarrow \top]}{\langle ag, C, M, T, Clrlnt \rangle \rightarrow \langle ag, C', M, T, Clrlnt \rangle} \quad (42)$$

$$\text{t.q. } C'_l = (C_l \setminus \{T_l\}) \cup \{k[(head' \leftarrow h)\theta]\}$$

$$\text{si } i = k[head' \leftarrow g; h]$$

$$\text{y } g\theta = TrEv(head)$$

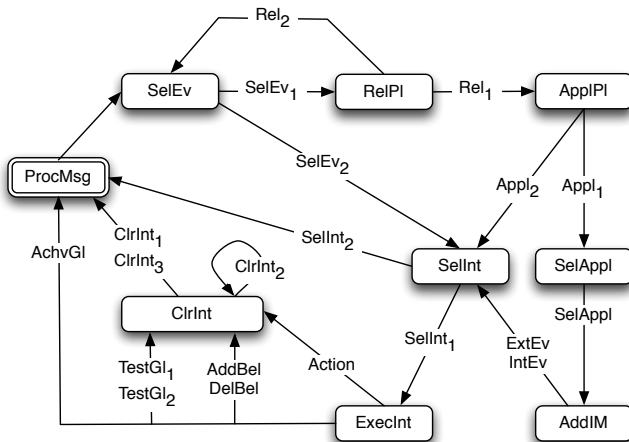


Limpieza II

$$(ClrInt_3) \quad \frac{T_i \neq [head \leftarrow \top] \wedge T_i \neq i[head \leftarrow \top]}{\langle ag, C, M, T, ClrInt \rangle \rightarrow \langle ag, C, M, T, ProcMsg \rangle} \quad (43)$$



Grafo de transiciones



Semántica operacional vs Lógicas BDI

- ▶ Ganamos una semántica clara, precisa y **computable**.
- ▶ Pero ¿Donde están mis operadores BDI_{CTL} ?
- ▶ ¿Qué **significa** que el agente crea, desee o intente algo?
- ▶ Recuperados en $CTL_{AgentSpeak(L)}$ [4, 5].



Sintaxis

- ▶ Si $at \in AgentSpeak(L)$, entonces at , $BEL(at)$, $DES(at)$, $INT(at)$ son fbf **intencionales** de $CTL_{AgentSpeak(L)}$.
- ▶ Las fbf de **estado** son:
 - s1 Toda fbf intencional es una fbf de estado.
 - s2 Si ϕ y ψ son fbf de estado, $\phi \wedge \psi$ y $\neg\phi$ lo son.
 - s3 Si ϕ es una fbf de camino, entonces $E\phi$ es una fbf de estado.
- ▶ Las fbf de **camino** son:
 - p1 Toda fbf de estado es una fbf de camino.
 - p2 Si ϕ y ψ son fbf de camino, $\neg\phi$, $\phi \wedge \psi$, $\bigcirc\phi$, $\diamond\phi$ y $\phi \cup \psi$ son fbf de camino.
- ▶ Ej. $INT(A\diamond go(paris)) \cup \neg BEL(go(paris, summer))$.



Semántica: Metas realizables

- ▶ Dada una intención, coleccionar las fbf atómicas que son sujeto de una **meta realizable** (*achieve goals*):

$$\begin{aligned}
 agls(\top) &= \{\}, \\
 agls(i[p]) &= \begin{cases} \{at\} \cup agls(i) & \text{if } p = +!at : ct \leftarrow h, \\ agls(i) & \text{otherwise} \end{cases}
 \end{aligned}$$



Semántica: BDI

$$\text{BEL}_{\langle ag, C \rangle}(\phi) \equiv ag_{bs} \models \phi$$

$$\text{INT}_{\langle ag, C \rangle}(\phi) \equiv \phi \in \bigcup_{i \in C_I} agIs(i) \vee \bigcup_{\langle te, i \rangle \in C_E} agIs(i)$$

$$\text{DES}_{\langle ag, C \rangle}(\phi) \equiv \langle +!\phi, i \rangle \in C_E \vee \text{INT}(\phi)$$



Semántica: Modelo

- ▶ $K = \langle S, R, V \rangle$ es una **estructura de Kripke** definida sobre el sistema de transición (Γ) de *AgentSpeak(L)*, donde:
 - ▶ S es un conjunto de configuraciones $\langle ag, C, M, T, s \rangle$.
 - ▶ $R \subseteq S^2$ es una relación serial t.q. para todo $(c_i, c_j) \in R$, $(c_i, c_j) \in \Gamma$.
 - ▶ V es una función de evaluación sobre los operadores intencionales y temporales, por ej., $V_{\text{BEL}}(c, \phi) \equiv \text{BEL}_{\langle ag, C \rangle}(\phi)$ en la configuración $c = \langle ag, C, M, T, s \rangle$, etc.
- ▶ $x = c_0, \dots, c_n$ denota un **camino**, una secuencia de configuraciones t.q. para todo $c_i \in S$, $(c_i, c_{i+1}) \in R$.
- ▶ x^i denota el sufijo del camino x a partir de la configuración c_i .



Semántica: fbf de estado

$$K, c_i \models \text{BEL}(\phi) \Leftrightarrow \phi \in V_{\text{BEL}}(c_i, \phi)$$

$$K, c_i \models \text{INT}(\phi) \Leftrightarrow \phi \in V_{\text{INT}}(c_i, \phi)$$

$$K, c_i \models \text{DES}(\phi) \Leftrightarrow \phi \in V_{\text{DES}}(c_i, \phi)$$

$$K, c_i \models \text{E}\phi \Leftrightarrow \exists x^i \exists c_{j \geq i} \in x^i \text{ t.q. } K, c_j \models \phi$$

$$K, c_i \models \text{A}\phi \Leftrightarrow \forall x^i \exists c_{j \geq i} \in x^i \text{ t.q. } K, c_j \models \phi$$

Semántica: fbf de camino

- ▶ La semántica del operador “hasta” es la del **weak until** (ψ puede no ocurrir nunca).
- ▶ Observe que $\models \bigcirc \alpha \equiv T_\iota = i[\text{head} \leftarrow \alpha; h] \wedge s = \text{ExecInt}$.

$K, c_i \models \phi \Leftrightarrow K, x^i \models \phi$, cuando ϕ es una fbf de estado

$K, c_i \models \bigcirc \phi \Leftrightarrow K, x^{i+1} \models \phi$

$K, c_i \models \diamond \phi \Leftrightarrow \exists c_{j \geq i} \in x^i \text{ t.q. } K, c_j \models \phi$

$K, c_i \models \phi \text{ U } \psi \Leftrightarrow (\exists c_{k \geq i} \text{ t.q. } K, x^k \models \psi \wedge$
 $\forall c_{i \leq j < k} K, x^j \models \phi) \vee (\exists c_{j \geq i} K, x^j \models \phi).$



Semántica: Corridas, satisfacción y validez

- ▶ Dada una configuración inicial c_0 , la **corrida** K_{Γ}^0 denota una secuencia de configuraciones c_0, c_1, \dots t.q. $\forall i \geq 1, c_i = \Gamma(c_{i-1})$.
- ▶ Dada una corrida K_{Γ}^0 , la propiedad $\phi \in CTL_{AgentSpeak(L)}$ se **satisface** si $\forall i \geq 0, K_{\Gamma}^0, c_i \models \phi$.
- ▶ Una propiedad $\phi \in CTL_{AgentSpeak(L)}$ es **válida**, si $K_{\Gamma}^0, c_0 \models \phi$ para toda configuración inicial c_0 .

Asimetría en sistemas $B^{KD45}D^{KD}I^{KD}$ vs Jason

- ▶ Jason no es equivalente a las Lógicas BDI [1]:

	AT1	AT2	AT3	AT4	AT5	AT6	AT7	AT8	AT9
Realismo fuerte	✓		✓	✓		✓	✓		✓
Realismo	✓	✓		✓	✓		✓	✓	
Realismo débil	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>AgentSpeak(L)</i>		✓	✓	✓		✓		✓	✓



Posposición finita

- ▶ ¿Estos agente mantienen sus intenciones indefinidamente o no?
- ▶ ¿Es válido el axioma de posposición finita?

$$\text{INT}(\phi) \rightarrow A\Diamond(\neg\text{INT}(\phi))$$

- ▶ Prueba: Asumamos $K, c_0 \models \text{INT}(\phi)$, entonces dada la definición de INT, existe un plan $p \in C_I \cup C_E$ con la cabeza $+!\phi$ en c_0 . p es finito por definición. Siguiendo $Clrlnt_X$ el plan es abandonado por éxito o fracaso.
- ▶ Observación: **Mecanismos de fallo.**



Compromiso ciego

- ▶ Los agentes *AgentSpeak(L)* **no satisfacen** la propiedad de compromiso ciego:

$$\text{INT}(A \diamond \phi) \rightarrow A \diamond \text{BEL}(\phi)$$

- ▶ Prueba: Asumamos c_0 donde
 $ag = \langle \{b(t_1)\}, \{+b(t_1) : \top \leftarrow p(t_2). \quad +!p(t_2) : \top \leftarrow +b(t_3).\} \rangle$. Se obtendrá una configuración c' donde $C_I = \{[+!p(t_2) : \top \leftarrow +b(t_3).]\}$ y $C_E = \{\}$ t.q. $K, c' \models \text{INT}(p(t_2))$ y $K, c' \not\models \text{BEL}(p(t_2))$. En la siguiente configuración c'' , $K, c'' \models \neg \text{INT}(p(t_2))$ y $K, c'' \not\models \text{BEL}(p(t_2))$.

Compromiso racional

- ▶ Un agente $AgentSpeak(L)$ **satisface** una forma limitada del compromiso racional:

$$INT(A\Diamond\phi) \rightarrow A(INT(A\Diamond\phi) \cup \neg BEL(E\Diamond\phi))$$

- ▶ Prueba: Siguiendo la demostración de posposición finita, en los casos de fallo el agente satisface eventualmente $\bigcirc\neg INT(\phi)$ debido a Rel_2
 - Para un evento $\langle te, i[+!\phi : c \leftarrow h.] \rangle$ no hubo planes relevantes y la intención asociada es abandonada.
- ▶ Se trata de una forma limitada de compromiso racional porque no hay representación explícita de la razón de abandono.



Antecedentes

- ▶ Vieira et al. [10] han propuesto una semántica operacional extendida de *AgentSpeak(L)* para implementar un lenguaje de comunicación basado en actos de habla para Jason.
- ▶ Debido a una literatura más abundante, pero sobre todo por razones históricas, se optó por adoptar **KQML** en esta tarea.
- ▶ La semántica de KQML fue propuesta por Labrou y Finin [6], con base en los trabajos de Perrault y Allen [7] y Cohen y Levesque [3].



Condiciones para Tell

- ▶ Condiciones preparatorias:
 - ▶ $Pre(S) : bel(S, X) \wedge know(S, want(R, know(R, bel(S, X))))$
 - ▶ $Pre(R) : intend(R, know(R, bel(S, X)))$
- ▶ Condiciones posteriores:
 - ▶ $Pos(S) : know(S, know(R, bel(S, X)))$
 - ▶ $Pos(R) : know(R, bel(S, X))$
- ▶ Satisfacción:
 - ▶ $know(R, bel(S, X))$

Sintaxis

- ▶ Asumiremos la existencia de una acción especial predefinida para enviar mensajes: $.send(Id, If, Cnt)$, donde:
 - ▶ Id es la identidad del destinatario,
 - ▶ If es la fuerza ilocutoria del mensaje y
 - ▶ Cnt su contenido.

Mensajes

- ▶ Los **mensajes** están soportados por una estructura $\langle mid, dest, fil, cnt \rangle$, donde:
 - ▶ *mid* es un **identificador** único del mensaje, generado automáticamente.
 - ▶ *dest* es el nombre del agente **destinatario**.
 - ▶ $fil \in \{Tell, Untell, Achieve, Unachieve, TellHow, UntellHow, AskOne, AskAll\}$ es la **fuerza ilocutoria** del mensaje.
 - ▶ *cnt* es el **contenido** del mismo. Dependiendo de la fuerza performativa del mensaje, su contenido puede ser: una fórmula atómica (*at*), o un conjunto de fórmulas atómicas (*ATs*), o una creencia (*b*), es decir una literal de base, o un conjunto de creencias (*Bs*), o un conjunto de planes (*PLs*).

Enviar una solicitud de información

$$\begin{array}{c}
 T_i = i[\text{head} \leftarrow \text{.send}(id, \text{ilf}, \text{cnt}); h], \\
 \text{ilf} \in \{\text{AskOne}, \text{AskAll}, \text{AskHow}\} \\
 \text{(ExecActSndAsk)} \frac{}{\langle ag, C, M, T, \text{ExecInt} \rangle \rightarrow \langle ag, C', M', T, \text{ProcMsg} \rangle}
 \end{array}
 \tag{44}$$

Donde: $M'_{Out} = M_{Out} \cup \{\langle mid, id, \text{ilf}, \text{cnt} \rangle\}$

$M'_{SI} = M_{SI} \cup \{(mid, i[\text{head} \leftarrow h])\}$

$C'_I = C_I \setminus \{T_i\}$

Enviar otro tipo de mensaje

$$(\text{ExecActSnd}) \frac{T_i = i[\text{head} \leftarrow .\text{send}(id, \text{ilf}, \text{cnt}); h], \quad \text{ilf} \notin \{\text{AskOne}, \text{AskAll}, \text{AskHow}\}}{\langle ag, C, M, T, \text{ExecInt} \rangle \rightarrow \langle ag, C', M', T, \text{ClrInt} \rangle} \quad (45)$$

Donde: $M'_{Out} = M_{Out} \cup \{\langle mid, id, \text{ilf}, \text{cnt} \rangle\}$
 $C'_i = (C_i \setminus \{T_i\}) \cup \{i[\text{head} \leftarrow h]\}$

Funciones auxiliares

- ▶ Una nueva función de **Función de selección** \mathcal{S}_M opera sobre los buzones: Regresa el primer mensaje en el buzón que recibe como parámetro.
- ▶ Una función booleana $SocAcc(id, ilf, at)$ donde ilf es la fuerza ilocutoria del mensaje del agente id con contenido proposicional at , que determina si el mensaje es **Socialmente aceptable** en un contexto dado.

Recibir información no solicitada

$$\begin{aligned}
 & S_M(M_{In}) = \langle mid, id, Tell, Bs \rangle, \\
 & \forall i (mid, i) \notin M_{SI}, \\
 & SocAcc(id, Tell, Bs) \\
 \text{(Tell)} \quad & \frac{}{\langle ag, C, M, T, ProcMsg \rangle \rightarrow \langle ag', C', M', T, SelEv \rangle} \quad (46)
 \end{aligned}$$

Donde:

$$\begin{aligned}
 M'_{In} &= M_{In} \setminus \{ \langle mid, id, Tell, Bs \rangle \} \\
 \forall b. b \in Bs : ag'_{bs} &= ag_{bs} + b[id] \\
 C_{E'} &= C_E \cup \{ \langle +b[id], T \rangle \}
 \end{aligned}$$

Recibir información solicitada

$$\begin{array}{c}
 S_M(M_{In}) = \langle mid, id, Tell, Bs \rangle, \\
 \exists i (mid, i) \in M_{SI}, \\
 SocAcc(id, Tell, Bs) \\
 \hline
 \langle ag, C, M, T, ProcMsg \rangle \rightarrow \langle ag', C', M', T, SelEv \rangle
 \end{array}$$

(TellRepl)

Donde: $M'_{In} = M_{In} \setminus \{ \langle mid, id, Tell, Bs \rangle \}$

$$M'_{SI} = M_{SI} \setminus \{ (mid, i) \}$$

$$C'_I = C_I \cup \{ i \}$$

$$\forall b. b \in Bs : ag'_{bs} = ag_{bs} + b[id]$$

$$C_{E'} = C_E \cup \{ \langle +b[id], T \rangle \}$$

(47)



Recibir una retractación no solicitada

$$\begin{array}{c}
 S_M(M_{In}) = \langle mid, id, Untell, ATs \rangle, \\
 \forall i (mid, i) \notin M_{SI}, \\
 SocAcc(id, Tell, ATs) \\
 \hline
 \langle ag, C, M, T, ProcMsg \rangle \rightarrow \langle ag', C', M', T, SelEv \rangle
 \end{array}
 \tag{48}$$

Donde: $M'_{In} = M_{In} \setminus \{ \langle mid, id, Tell, ATs \rangle \}$
 $\forall b. b \in \{ at\theta \mid \theta \in Test(ag_{bs}, at) \wedge at \in ATs \} :$
 $ag'_{bs} = ag_{bs} - b[id]$
 $C_{E'} = C_E \cup \{ \langle -b[id], T \rangle \}$

Recibir una retractación solicitada

$$\begin{array}{c}
 S_M(M_{In}) = \langle mid, id, Untell, ATs \rangle \\
 \exists i (mid, i) \in M_{SI} \\
 \text{SocAcc}(id, Untell, ATs) \\
 \hline
 \text{(UntellRepl)} \quad \langle ag, C, M, T, ProcMsg \rangle \rightarrow \langle ag', C', M', T, SelEv \rangle
 \end{array}$$

$$\text{Donde: } M'_{In} = M_{In} \setminus \{ \langle mid, id, Tell, ATs \rangle \} \quad (49)$$

$$M'_{SI} = M_{SI} \setminus \{ (mid, i) \}$$

$$C'_I = C_I \cup \{ i \}$$

$$\forall b. b \in \{ at\theta \mid \theta \in Test(ag_{bs}, at) \wedge at \in ATs \} :$$

$$ag'_{bs} = ag_{bs} - b[id]$$

$$C_{E'} = C_E \cup \{ \langle -b[id], T \rangle \}$$

Recibir una meta realizable

$$\begin{array}{c}
 S_M(M_{In}) = \langle mid, id, Achieve, at \rangle \\
 SocAcc(id, Achieve, at) \\
 \hline
 \text{(Achieve)} \quad \langle ag, C, M, T, ProcMsg \rangle \rightarrow \langle ag, C', M', T, SelEv \rangle
 \end{array} \quad (50)$$

Donde: $M'_{In} = M_{In} \setminus \{ \langle mid, id, Achieve, at \rangle \}$
 $C'_E = C_E \cup \{ +!at, T \}$

Recibir un plan solicitado (*know-how*)

$$\begin{array}{c}
 \mathcal{S}_M(M_{In}) = \langle mid, id, TellHow, PLs \rangle \\
 \exists i (mid, i) \in M_{SI} \\
 SocAcc(id, TellHow, PLs) \\
 \hline
 \langle ag, C, M, T, ProcMsg \rangle \rightarrow \langle ag', C', M', T, SelEv \rangle
 \end{array}
 \tag{51}$$

Donde: $M'_{In} = M_{In} \setminus \{ \langle mid, id, TellHow, at \rangle \}$

$M'_{SI} = M_{SI} \setminus \{ (mid, i) \}$

$C'_I = C_I \cup \{ i \}$

$ag'_{ps} = ag_{ps} \cup PLs$

Recibir una retractación de plan

$$\begin{array}{c}
 S_M(M_{In}) = \langle mid, id, UntellHow, PLs \rangle \\
 SocAcc(id, UntellHow, PLs) \\
 \hline
 \langle ag, C, M, T, ProcMsg \rangle \rightarrow \langle ag', C', M', T, SelEv \rangle
 \end{array}
 \tag{52}$$

Donde:

$$\begin{aligned}
 M'_{In} &= M_{In} \setminus \{ \langle mid, id, UntellHow, at \rangle \} \\
 M'_{SI} &= M_{SI} \setminus \{ (mid, i) \} \\
 ag'_{ps} &= ag_{ps} - PLs
 \end{aligned}$$

Recibir una pregunta

$$(AskOne) \frac{S_M(M_{In}) = \langle mid, id, AskOne, \{at\} \rangle}{SocAcc(id, AskOne, \{at\})}{\langle ag, C, M, T, ProcMsg \rangle \rightarrow \langle ag, C, M', T, SelEv \rangle}$$

Donde: $M'_{In} = M_{In} \setminus \{ \langle mid, id, AskOne, at \rangle \}$

$$M'_{Out} = \begin{cases} M_{Out} \cup \{ \langle mid, id, Tell, AT \rangle \}, \\ AT = \{ at\theta \}, \theta \in Test(ag_{bs}, at) & \text{Si } Test(ag_{bs}, at) \neq \{ \} \\ M_{Out} \cup \{ \langle mid, id, Untell, \{at\} \rangle \} & \text{En otro caso} \end{cases}$$

(53)



Universidad Veracruzana

Referencias I

- [1] RH Bordini y AF Moreira. "Proving BDI properties of agent-oriented programming languages". En: *Annals of Mathematics and Artificial Intelligence* 42 (2004), págs. 197-226.
- [2] WF Clocksin y CS Melish. *Programming in Prolog, using the ISO standard*. Berlin-Germany: Springer-Verlag, 2003.
- [3] PR Cohen y HJ Levesque. "Speech Acts and Rationality". En: *ACL*. 1985, págs. 49-60.
- [4] A Guerra-Hernández, JM Castro-Manzano y A El-Fallah-Seghrouchni. "Toward an AgentSpeak(L) Theory of Commitment and Intentional Learning". En: *MICAI 2008*. Ed. por A Gelbuch y EF Morales. Vol. 5317. Lecture Notes in Artificial Intelligence. Berlin, Germany: Springer-Verlag, 2008, págs. 848-858.
- [5] A Guerra-Hernández, JM Castro-Manzano y A El-Fallah-Seghrouchni. "CTL AgentSpeak(L): a Specification Language for Agent Programs". En: *Journal of Algorithms* 64 (2009), págs. 31-40.
- [6] Y Labrou y TW Finin. "A Semantics Approach for KQML - A General Purpose Communication Language for Software Agents". En: *CIKM '94 Proceedings of the third international conference on Information and knowledge management*. New York, NY, USA: ACM, 1994, págs. 447-455.

Referencias II

- [7] CR Perrault y JF Allen. "A Plan-Based Analysis of Indirect Speech Acts". En: *American Journal of Computational Linguistics* 6.3-4 (1980), págs. 167-182.
- [8] GD Plotkin. *A Structural Approach to Operational Semantics*. Inf. téc. DAIMI FN-19. University of Aarhus, 1981. URL: citeseer.ist.psu.edu/article/plotkin81structural.html.
- [9] AS Rao y MP Georgeff. "Decision Procedures for BDI Logics". En: *Journal of Logic and Computation* 8.3 (1998), págs. 293-342.
- [10] R Vieira et al. "On the Formal Semantics of Speech-Act Based Communication in an Agent-Oriented Programming Language". En: *Journal of Artificial Intelligence Research* 29 (2007), págs. 221-267.