



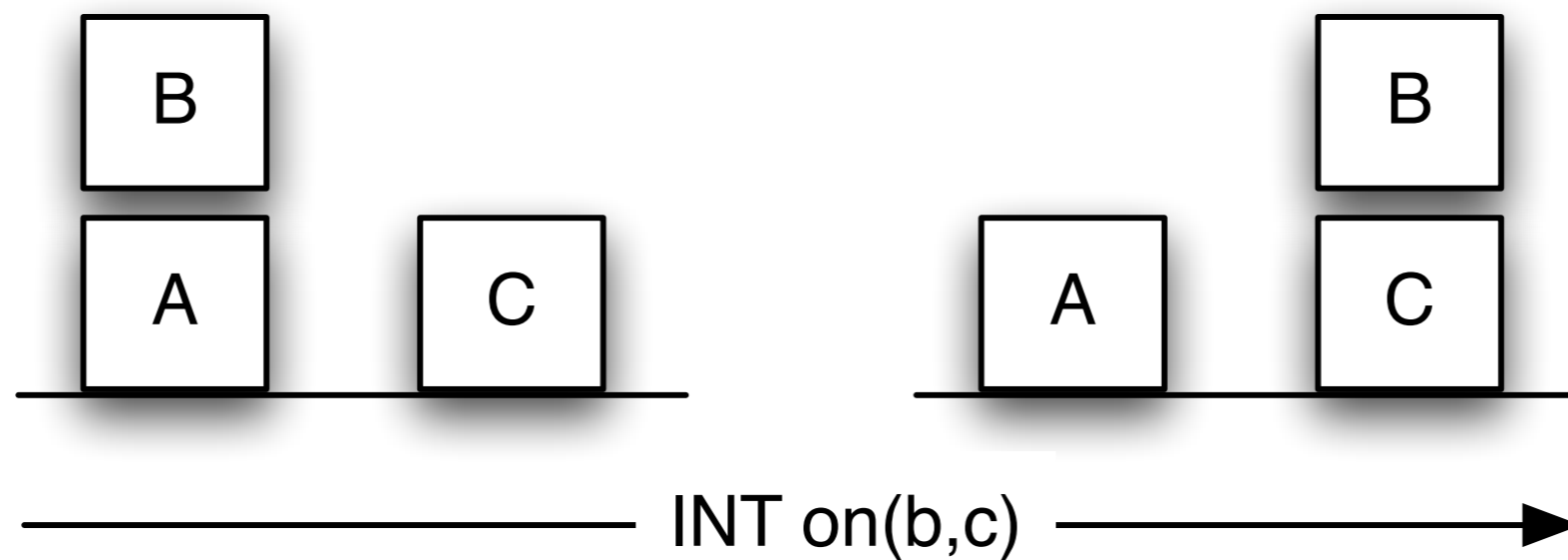
Universidad de Valencia

**Master en Computación
Avanzada y Sistemas
Inteligentes**

Aprendizaje Intencional y
Reconsideración

Dr. Alejandro Guerra Hernández
Universidad Veracruzana (México)
Departamento de Inteligencia Artificial

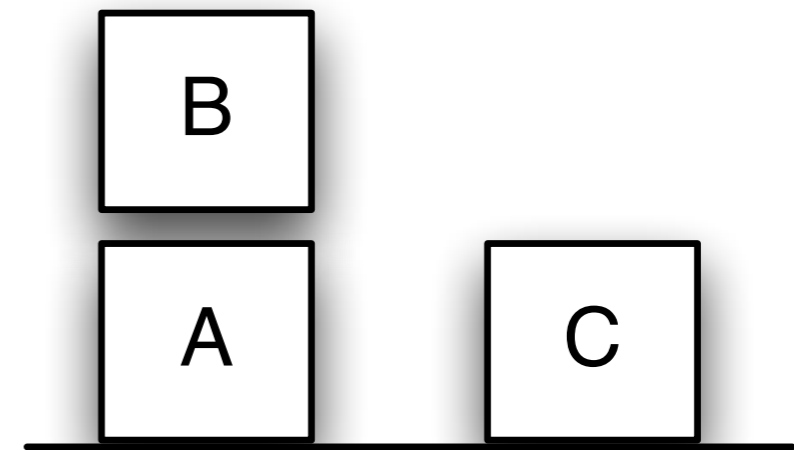
Intenciones, Planes y Razonamiento Práctico



Mundo de los bloques

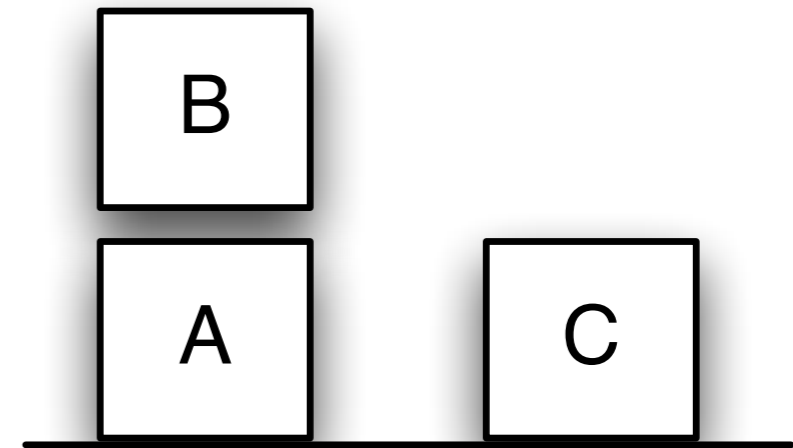
Creencias

```
6 on(b, a) .  
7 on(a, table) .  
8 on(c, table) .
```



Creencias y Reglas

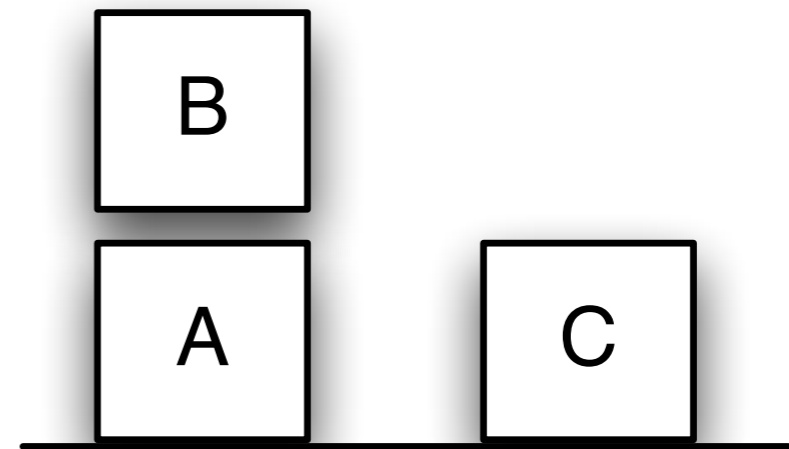
```
6 on(b, a) .  
7 on(a, table) .  
8 on(c, table) .
```



```
4 clear(table) .  
5 clear(X) :- not(on(_, X)) .
```

Creencias, Reglas y Preguntas (*test goals*)

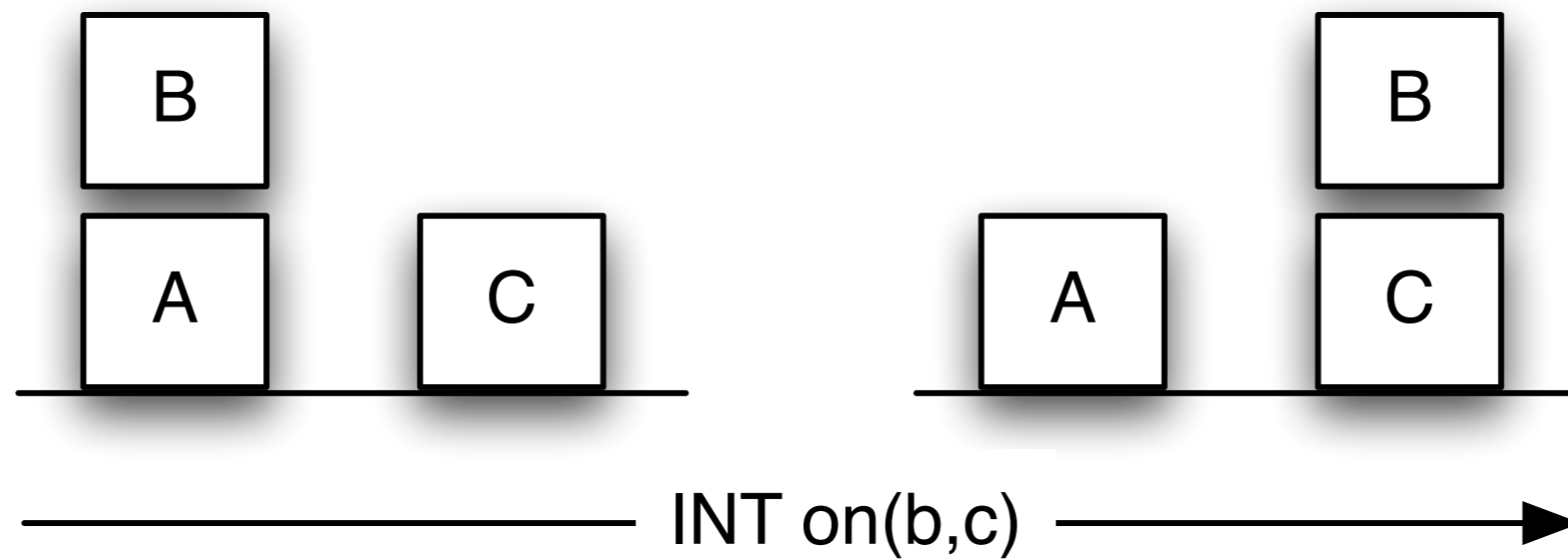
```
6 on(b, a) .  
7 on(a, table) .  
8 on(c, table) .
```



```
4 clear(table) .  
5 clear(X) :- not(on(_, X)) .
```

?clear(b) => true ?on(a, X) => {X\b}

Deseos (*achieve goals*)



! on (b , c)

Planes (*trigger_event* : *ctxt* <- *body*)

Evento Disparador
(*trigger_event*)

Agregar ó Eliminar una meta (test o achive) ó una creencia.

```
26 +!on(X, Y) : on(X, Y). // already achieved
27 +!on(X, Y) <- !clear(X); !clear(Y); move(X, Y).
```

Planes (*trigger_event : ctxt <- body*)

Contexto (*ctxt*)

Conjunción de literales
de primer orden

```
26 +!on(X, Y) : on(X, Y). // already achieved
27 +!on(X, Y) <- !clear(X); !clear(Y); move(X, Y).
```

Planes (*trigger_event : ctxt <- body*)

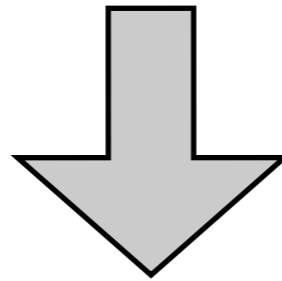
```
26 +!on(X,Y) : on(X,Y). // already achieved
27 +!on(X,Y) <- !clear(X); !clear(Y); move(X,Y).
```

Secuencia de metas, acciones, y actualizaciones de creencias

Cuerpo (*body*)

Intenciones

```
+!on(a,c) <- !clear(a) ; !clear(c) ; move(a,c) .
```



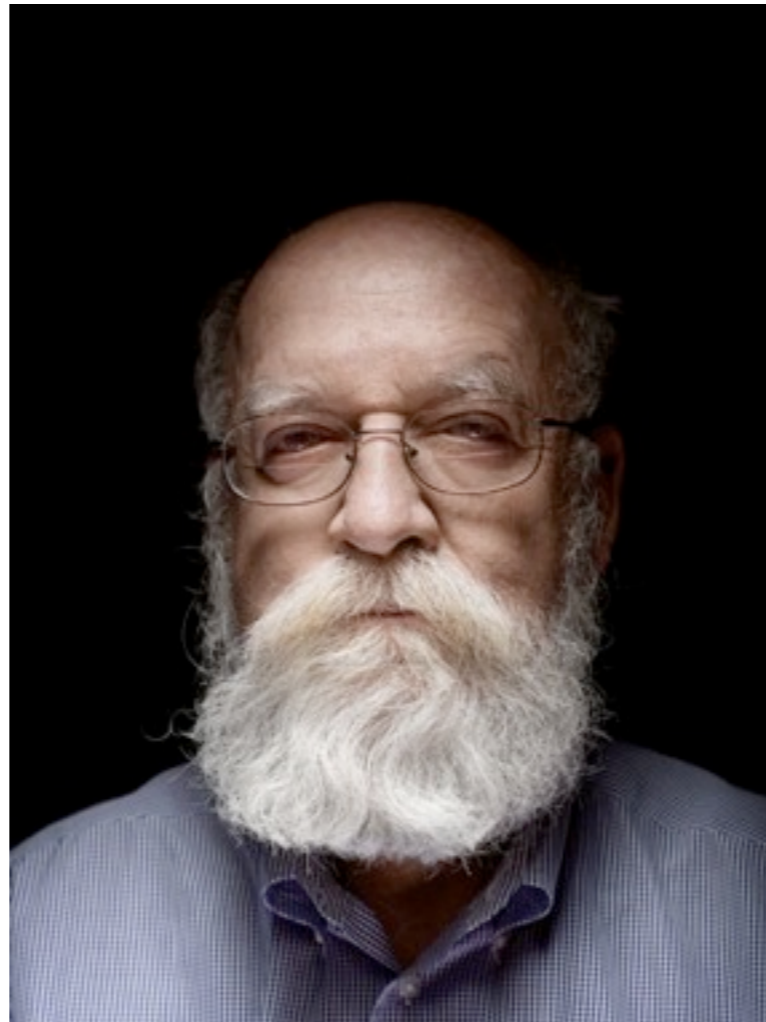
```
+!clear(a) : ?on(b,a) <- ?clear(W) ; move(b,W) .
```

```
+!on(a,c) <- !clear(c) ; move(a,c) .
```

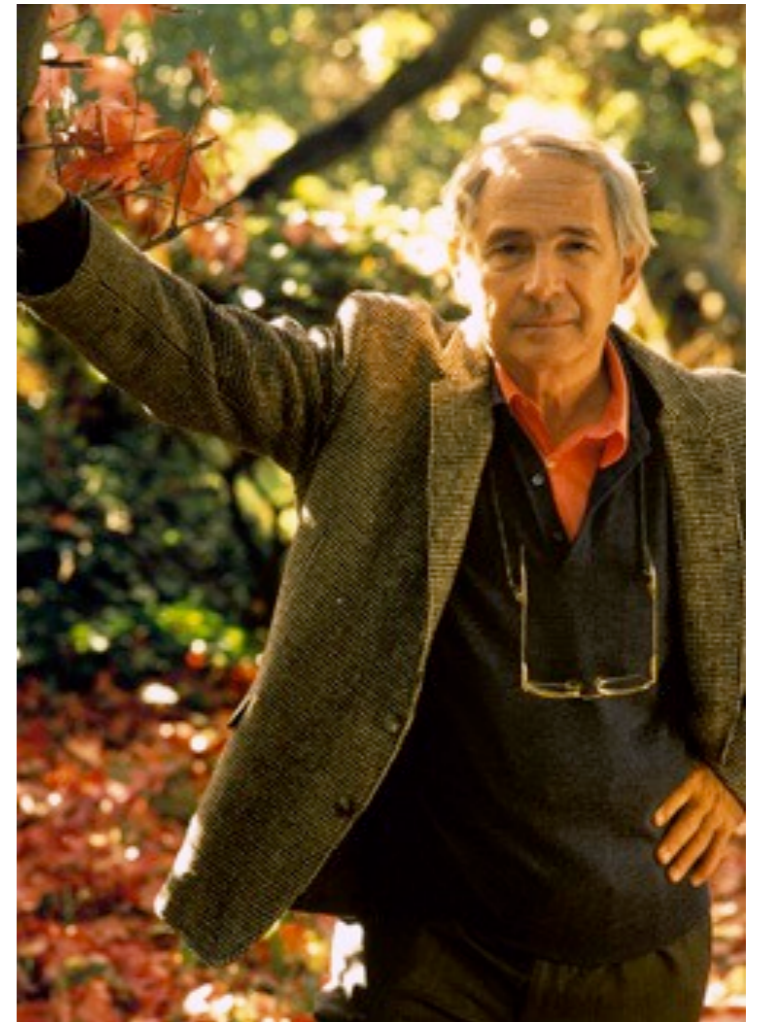
Modelo racional BDI (agentes)



Bratman

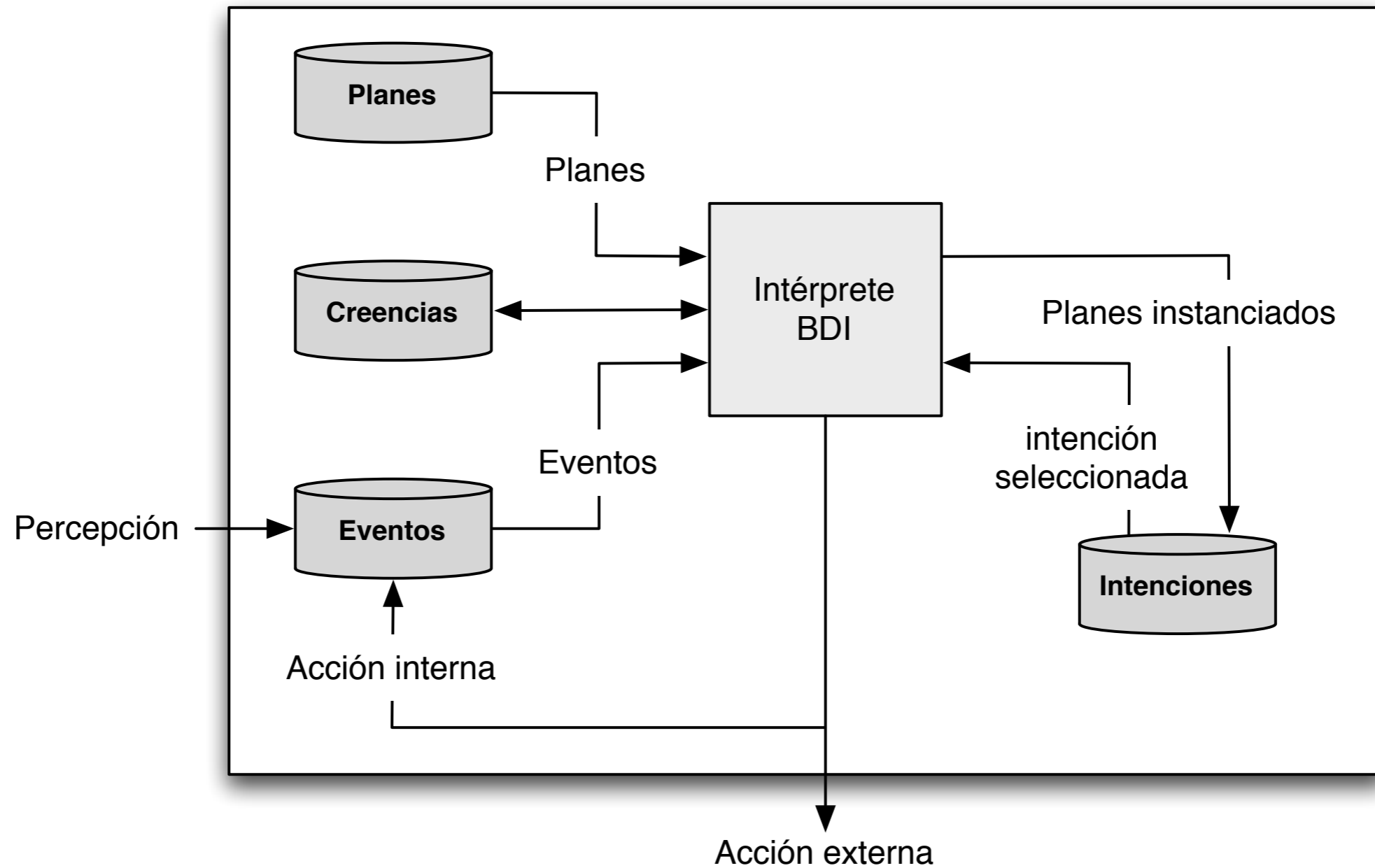


Dennett



Searle

Agente BDI



Agente AgentSpeak(L): configuraciones

$$c = \langle ag, C, M, T, s \rangle$$

Donde:

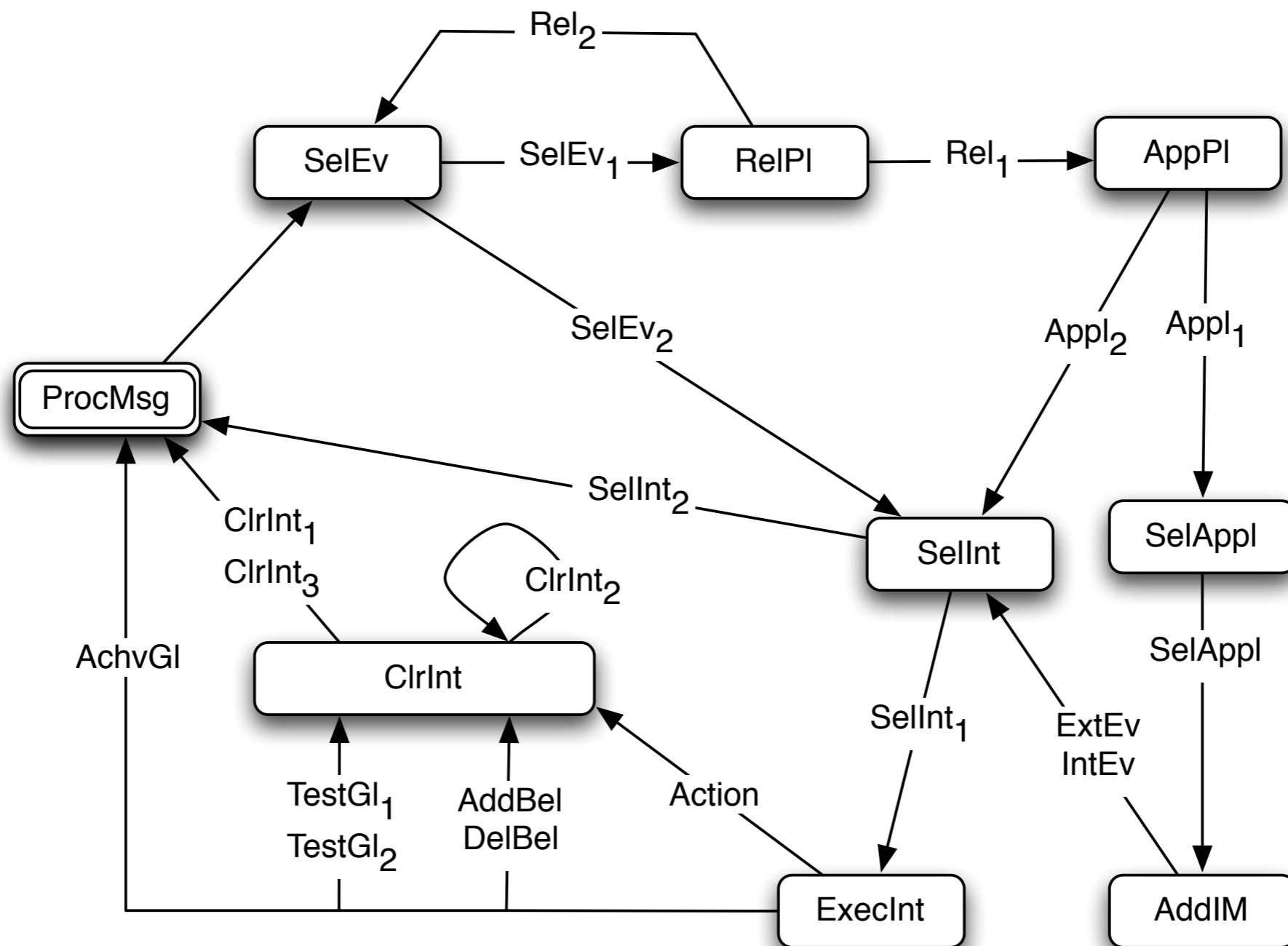
$$ag = \langle bs, ps \rangle$$

$$C = \langle I, E, A \rangle$$

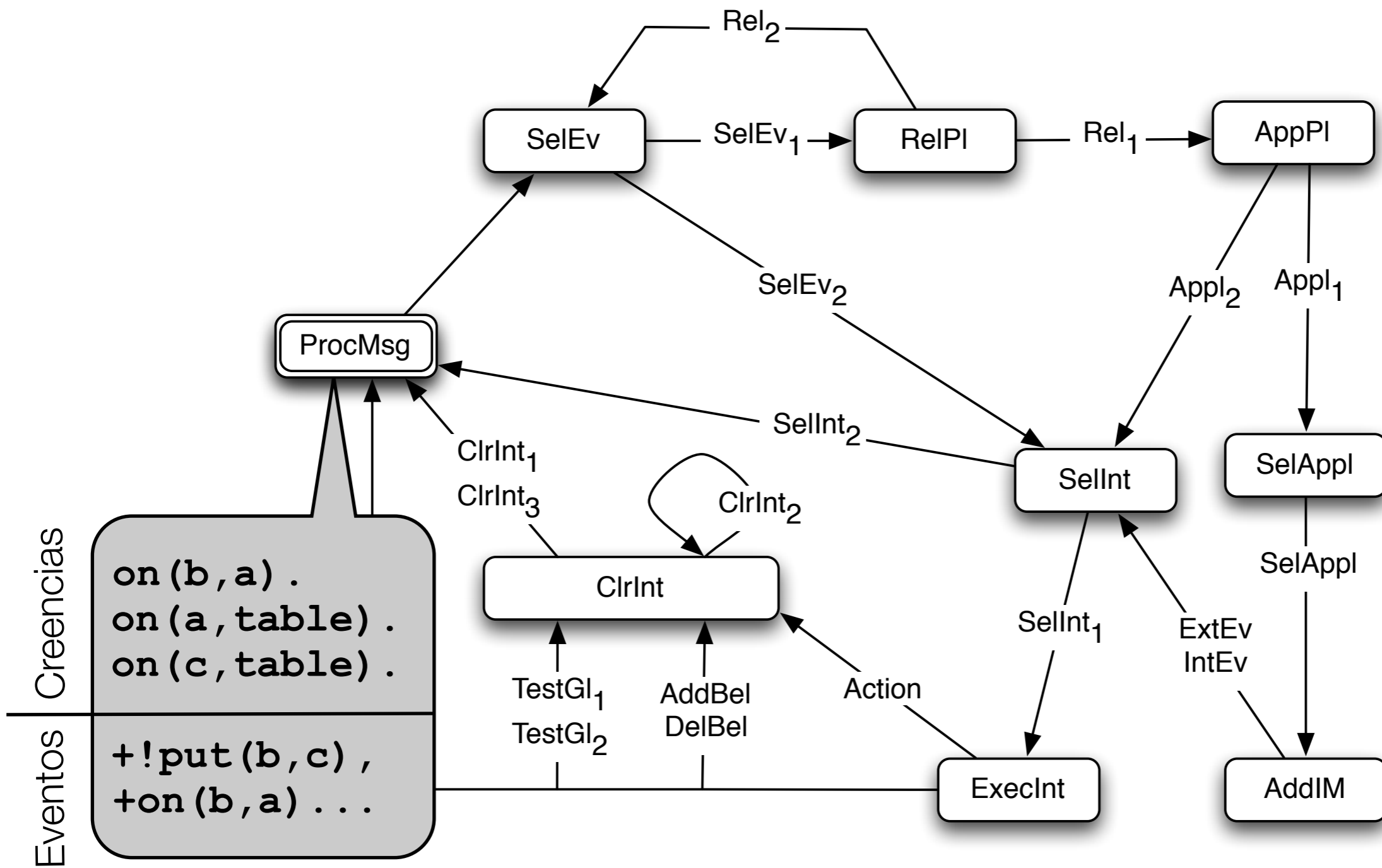
$$M = \langle In, Out, SI \rangle$$

$$s \in Labels$$

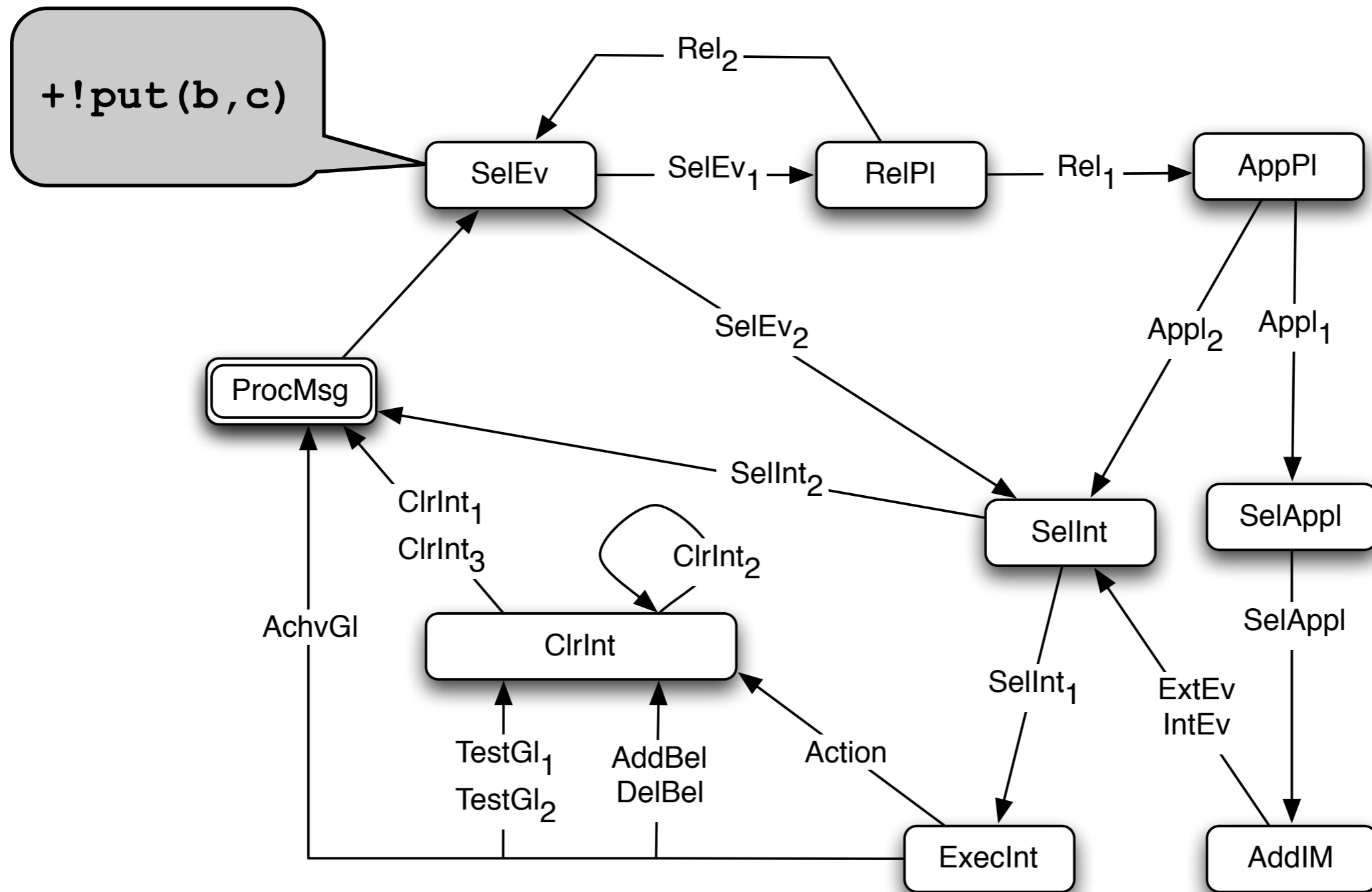
AgentSpeak(L): Semántica



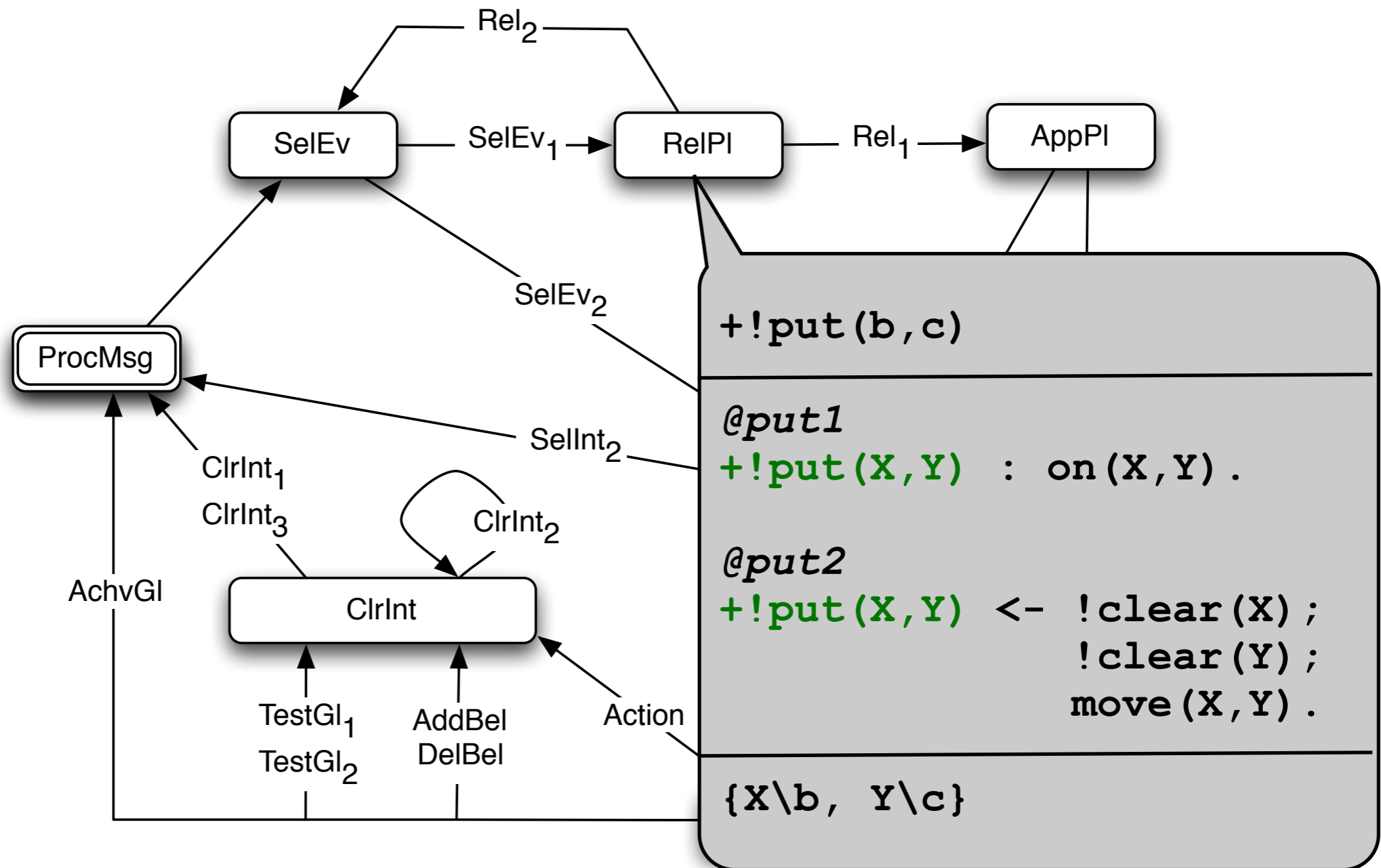
AgentSpeak(L): Procesar mensajes y percepción



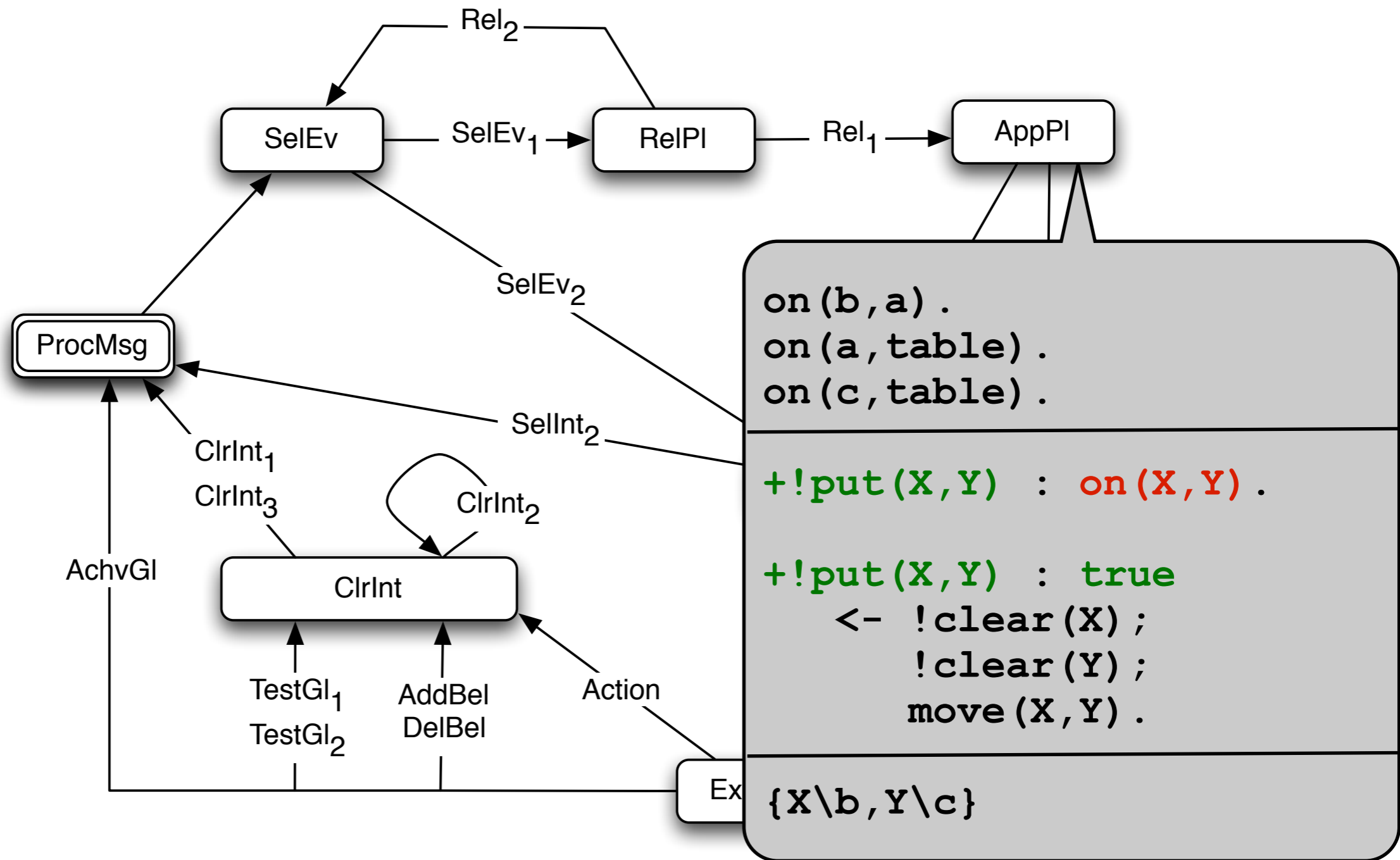
AgentSpeak(L): Seleccionar un evento



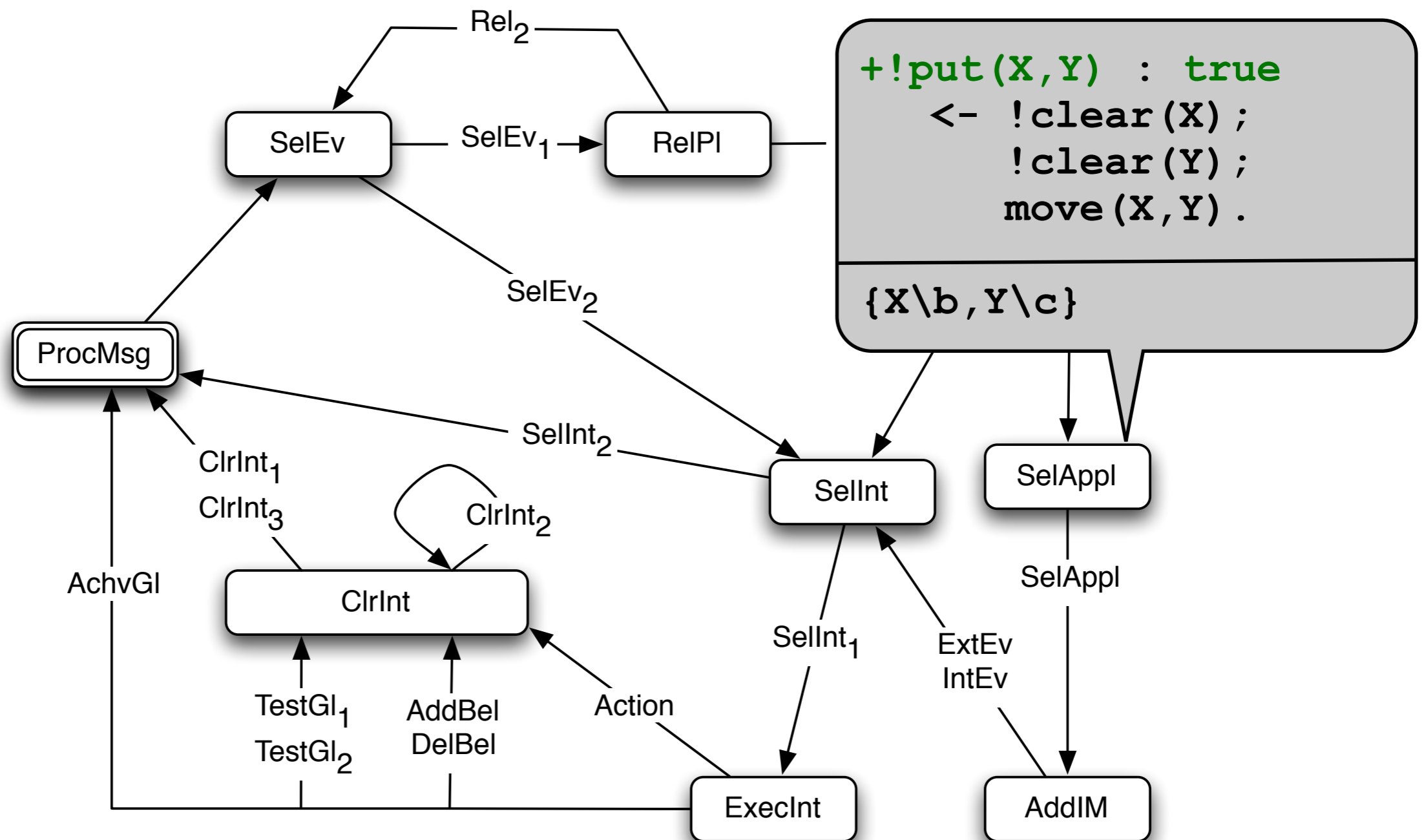
AgentSpeak(L): Seleccionar planes relevantes



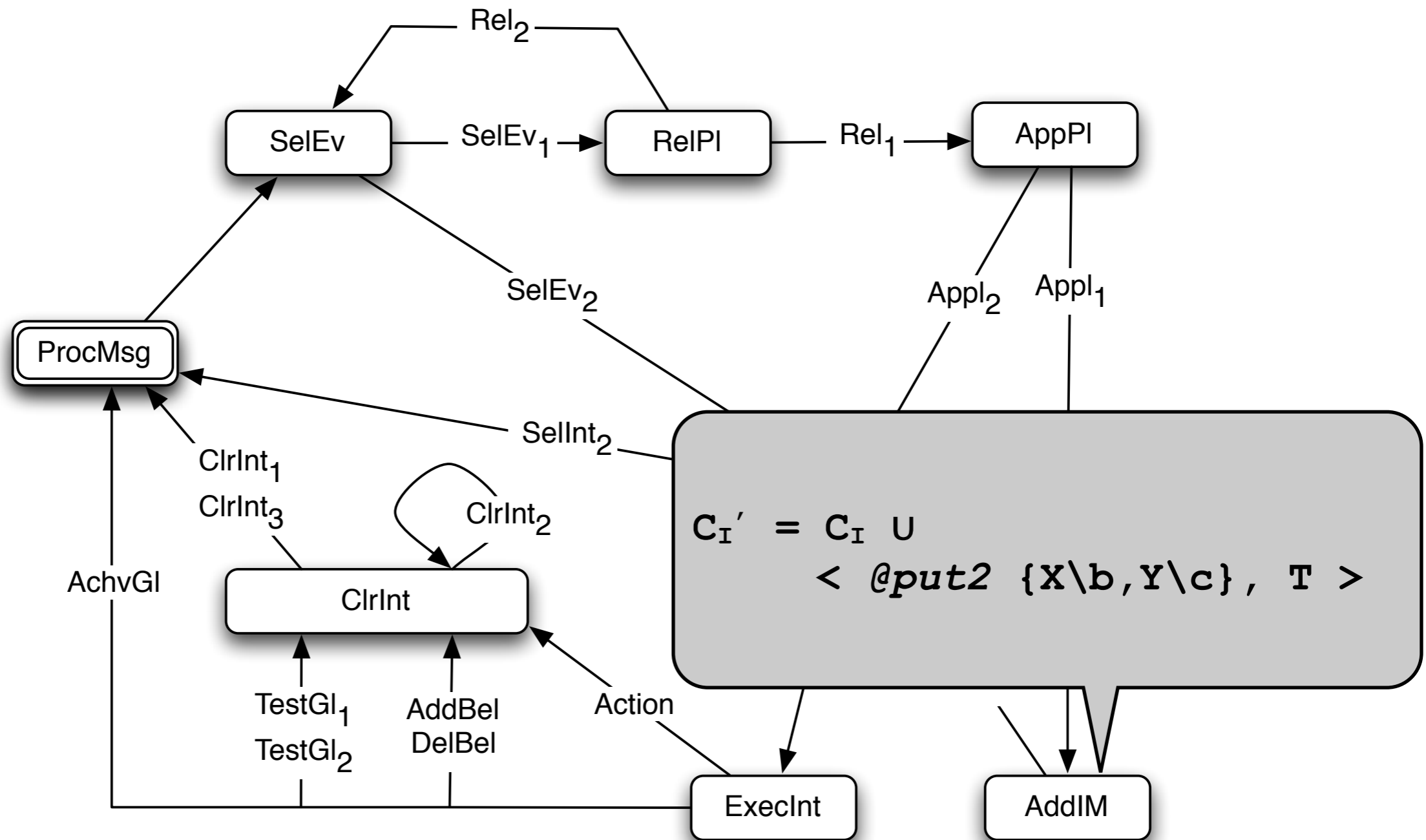
AgentSpeak(L): Seleccionar planes aplicables



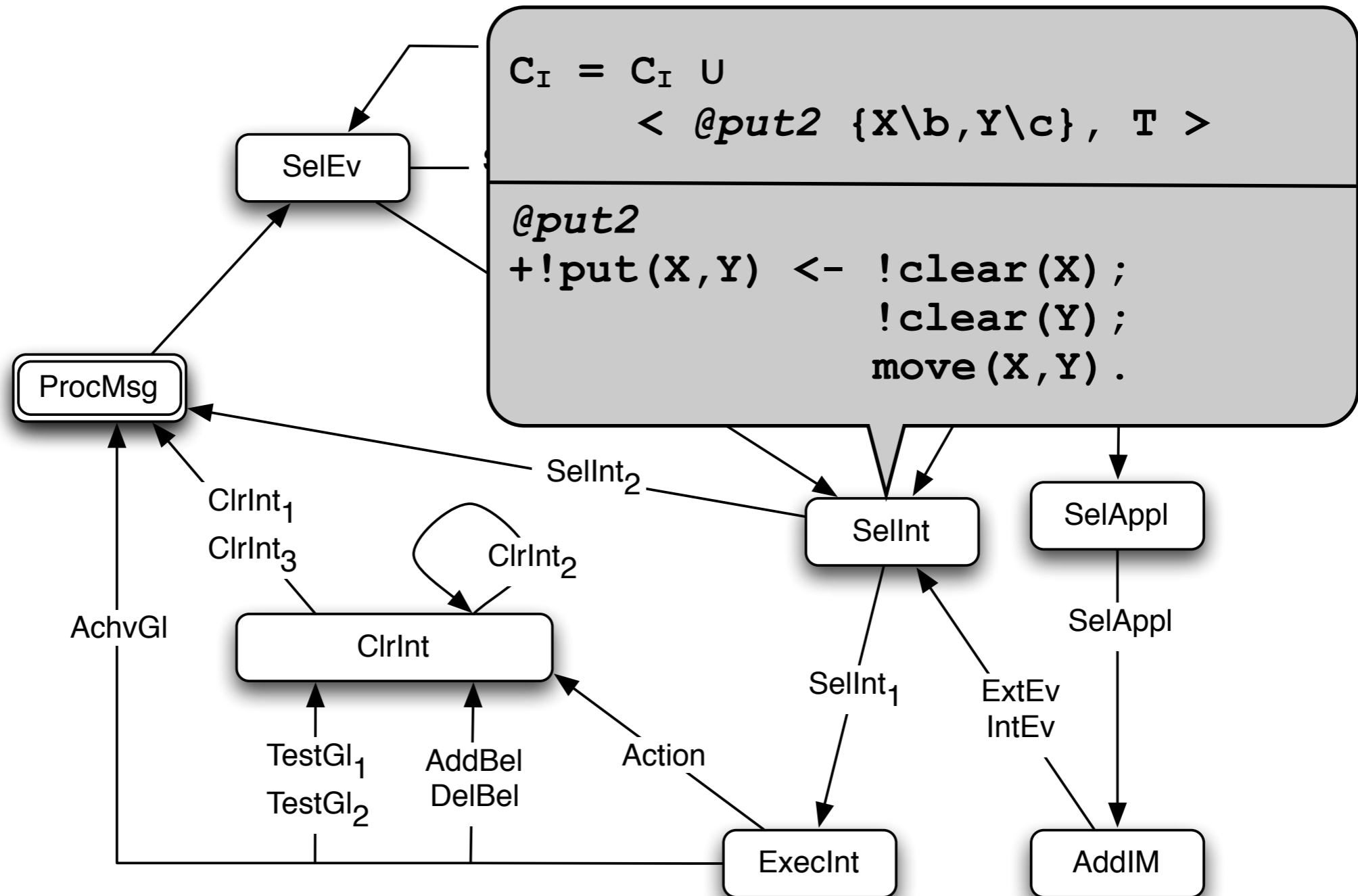
AgentSpeak(L): Seleccionar un plan aplicable



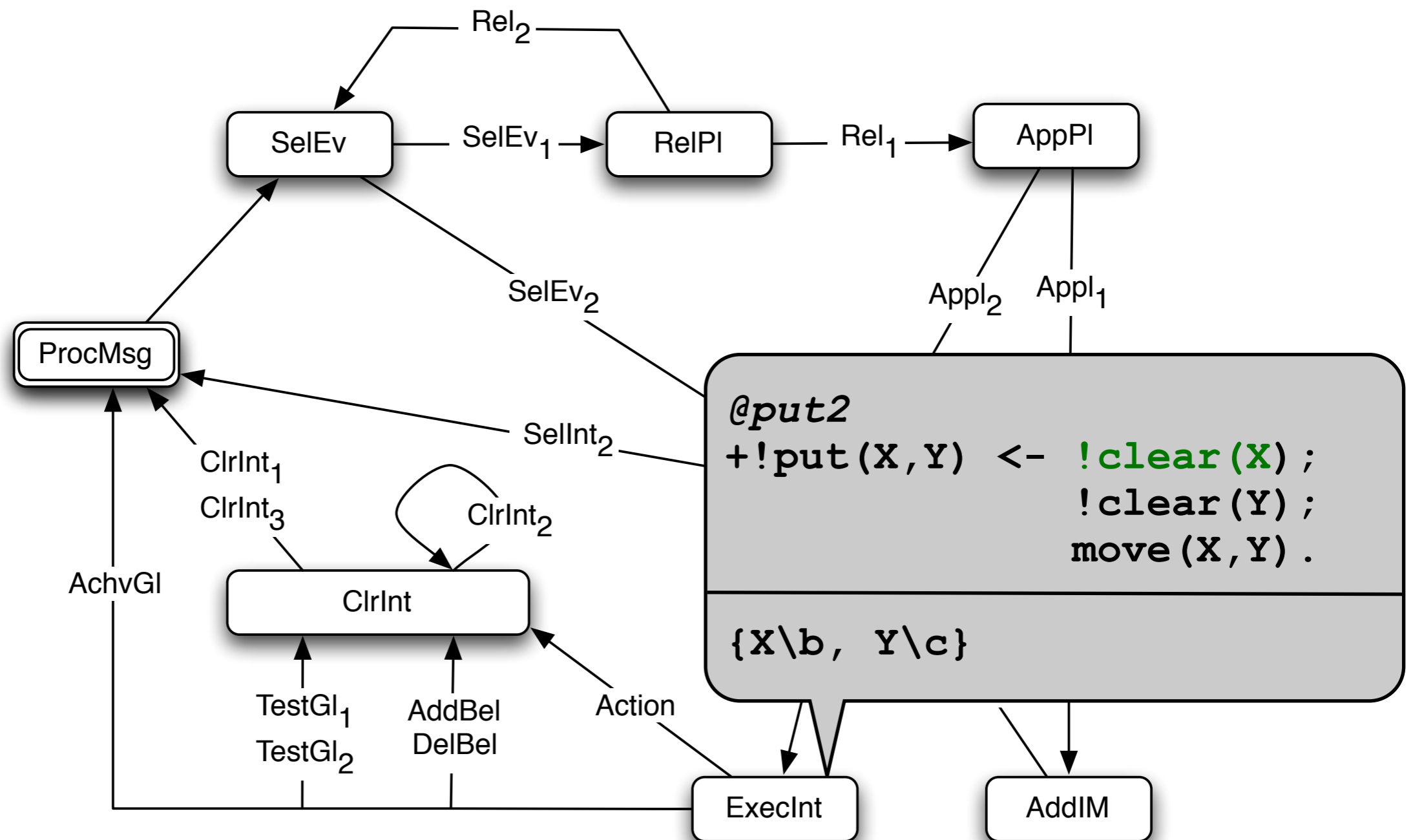
AgentSpeak(L): Formar una Intención



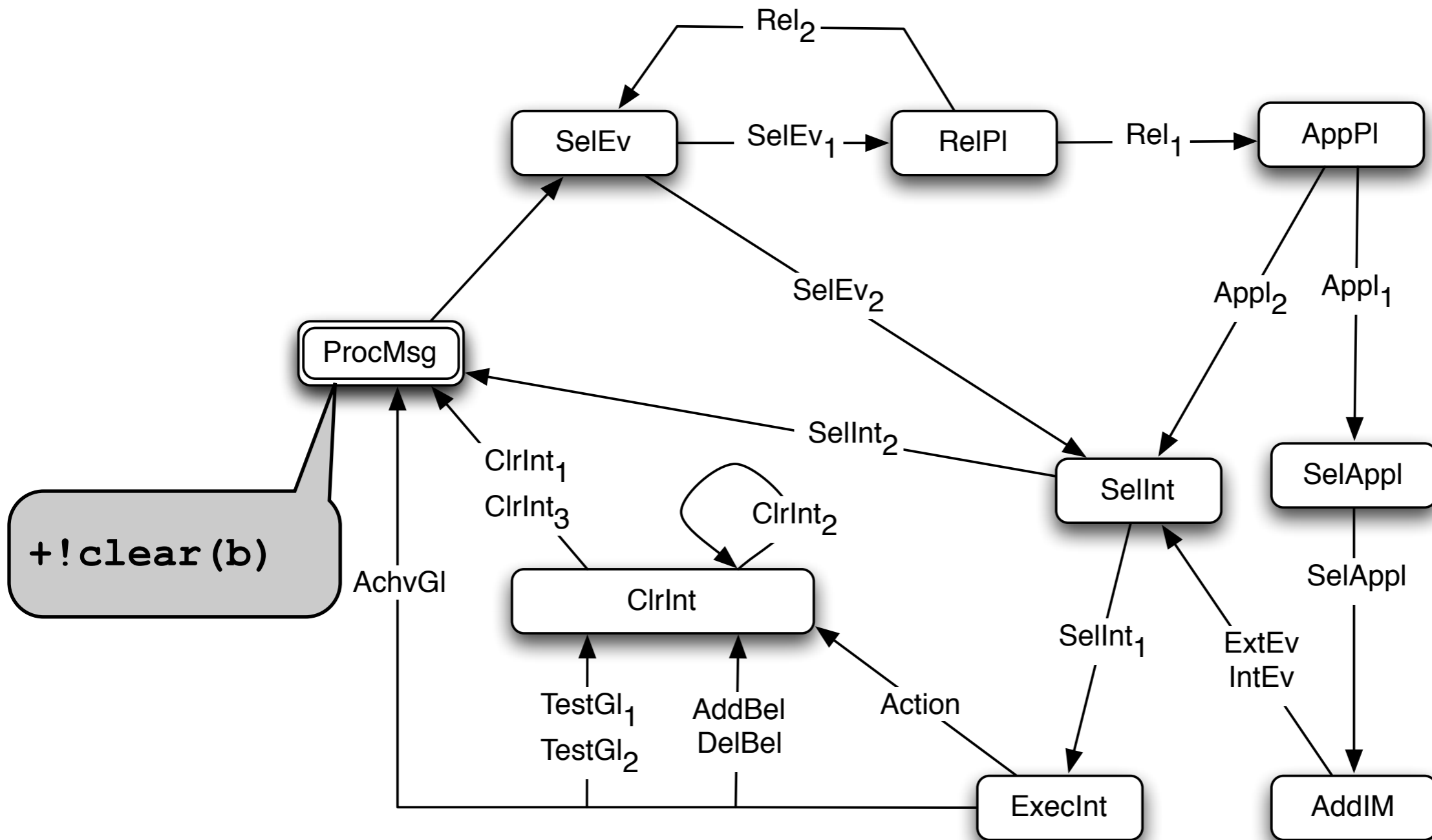
AgentSpeak(L): Seleccionar una Intención



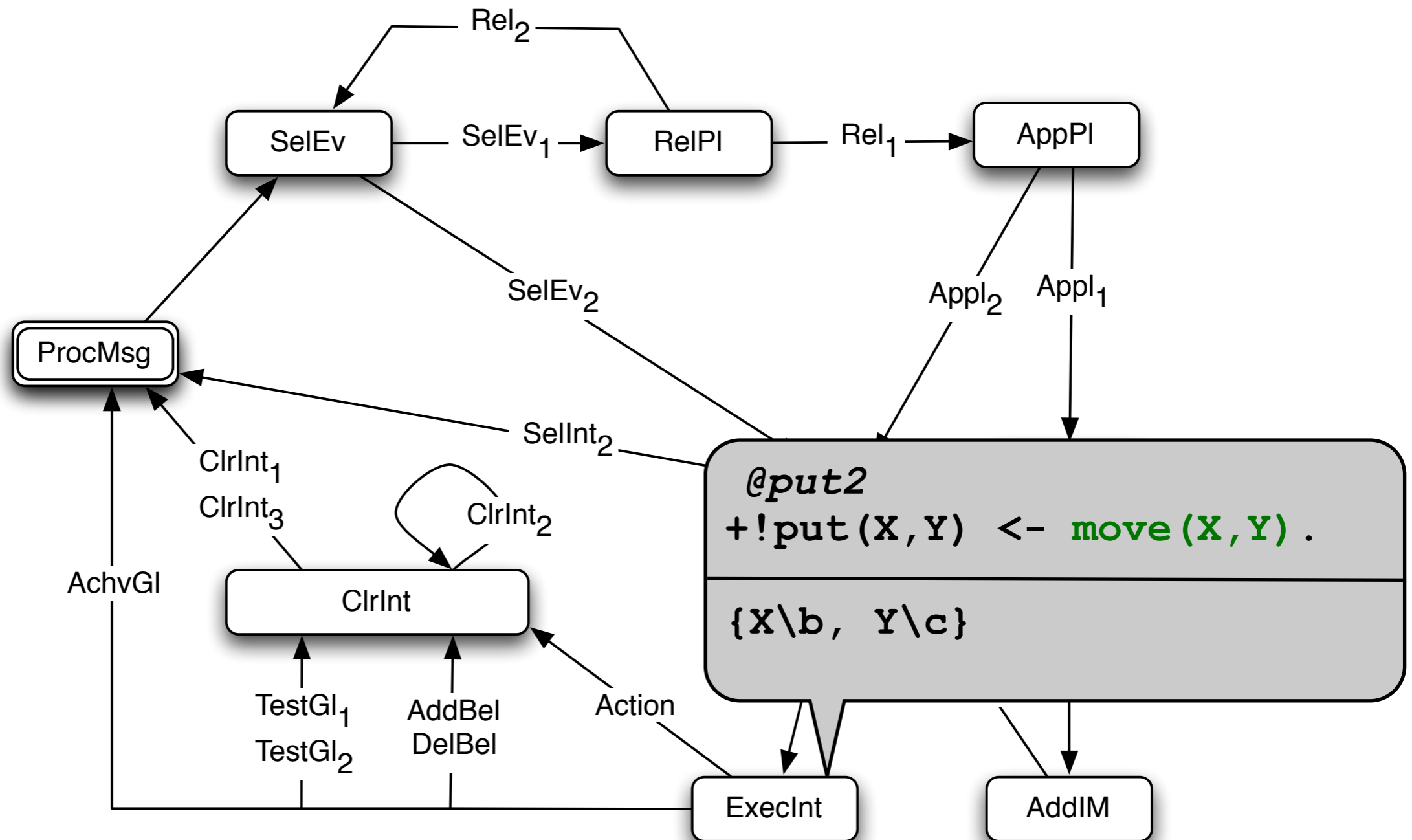
AgentSpeak(L): Ejecutar la Intención seleccionada



AgentSpeak(L): Agregar nuevo evento



AgentSpeak(L): Si todo va bien...



AgentSpeak(L): ¿Y si move falla?

- La intención completa falla!
- Es necesario corregir estos fallos, aprendiendo un nuevo contexto para el plan fallido, que prevenga la adopción incorrecta de intenciones en el futuro (Aprendizaje Intencional):

+!put (X,Y) : clear (X) & clear (Y) <- move (X,Y) .

- Es necesario formar políticas de abandono de intenciones que nos permitan abandonar las intenciones antes de llegar a su inminente fallo (Reconsideración):

drop (put (X,Y)) :- .intend (put (X,Y)) & not clear (X) .

drop (put (X,Y)) :- .intend (put (X,Y)) & not clear (Y) .

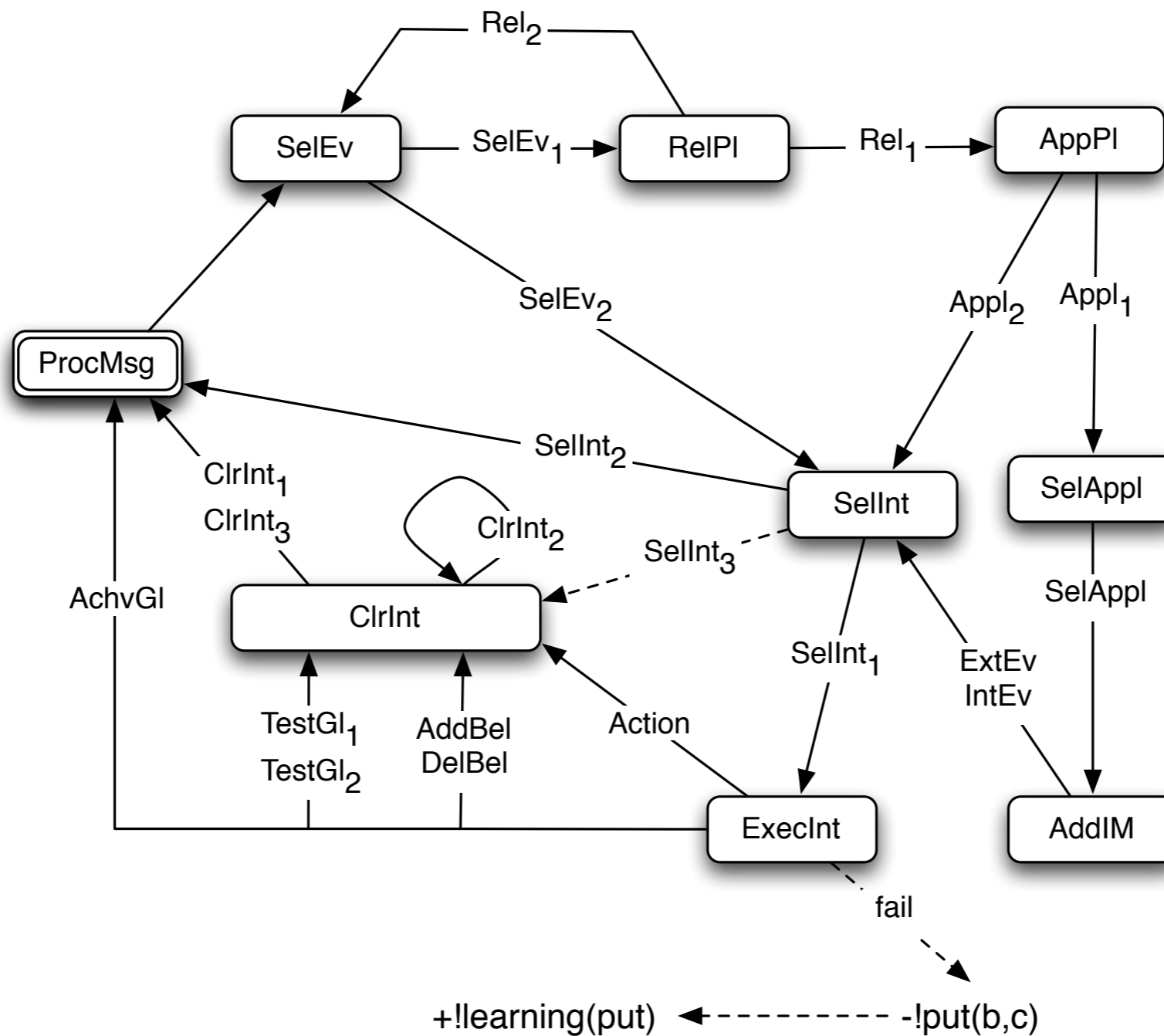
Reconsideración

- Formalmente, deseamos definir agentes que sigan un compromiso racional (*single-minded commitment*):

$$\text{INTEND}(A\Diamond\phi) \implies A(\text{INTEND}(A\Diamond\phi) \text{ U } (\text{BEL}(\phi) \vee \neg\text{BEL}(\text{E}\Diamond\phi)))$$

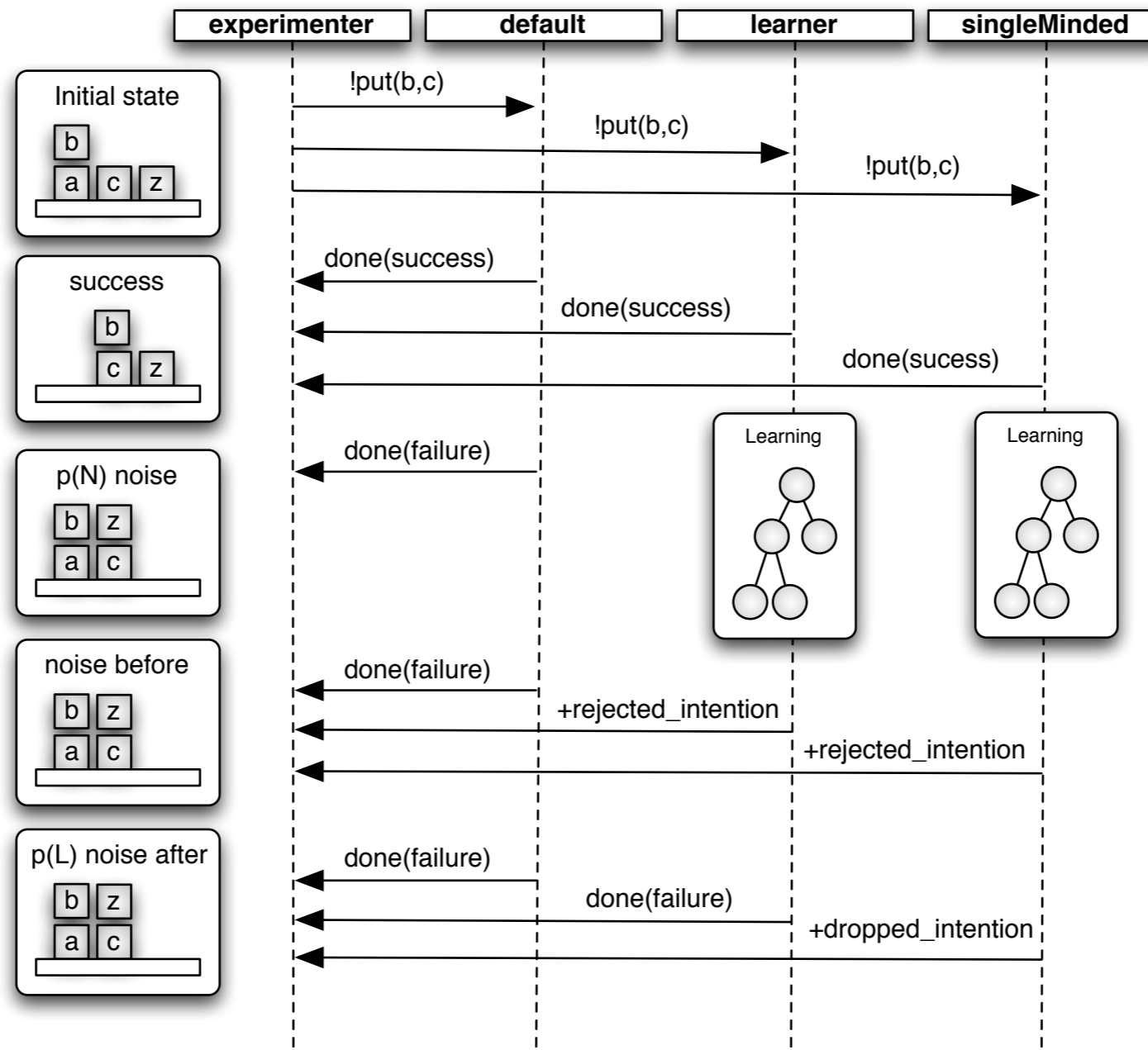
- Donde, según la definición de BDI_{CTL} , $A =$ para todo futuro posible, $\Diamond =$ eventualmente y $\text{U} =$ hasta que.
- Las razones para no creer que en algún futuro eventualmente Φ sea el caso, deben ser aprendidas por el agente de forma que...

AgentSpeak(L): Abandono de intenciones



Aprendizaje Intencional: Un experimento

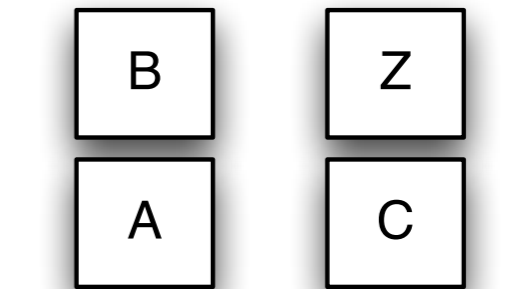
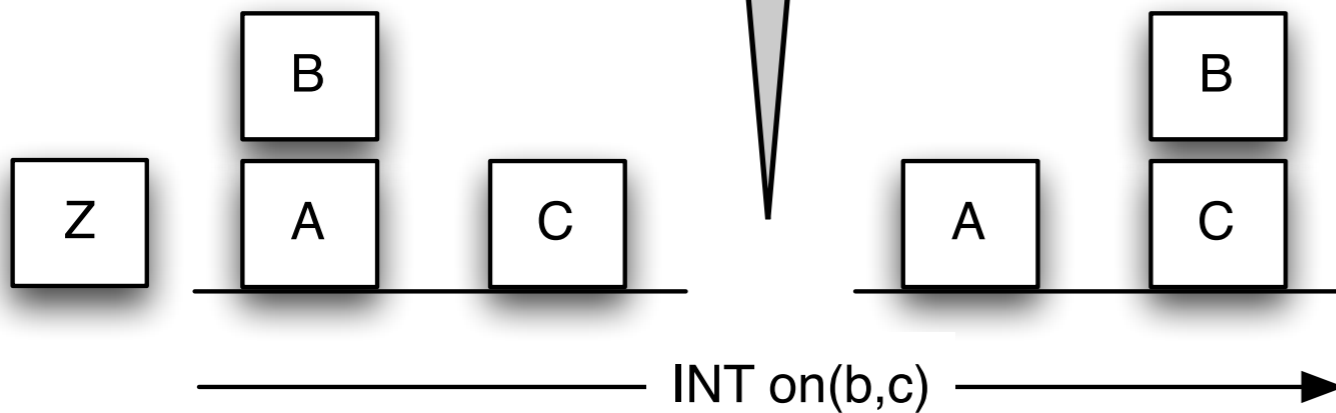
`+!put(X,Y) <- move(X,Y).`



Aprendizaje Intencional: Ejemplos

```
begin(model(1))  
succ.  
intend(put,b,c).  
on(b,a).  
on(a,table).  
on(c,table).  
on(z,table).  
end(model(1))
```

```
begin(model(2))  
fail.  
intend(put,b,c).  
on(b,a).  
on(a,table).  
on(c,table).  
on(z,c).  
end(model(2))
```



Aprendizaje intencional: Sesgo del lenguaje

```
rmode (clear (+V1)) .  
rmode (on (+V1, +V2)) .  
rmode (on (+V2, +V1)) .
```

```
<-Q = intend (put, A, B) .
```

```
clear (A) .  
clear (B) .  
on (A, B) .  
on (B, A) .
```

Aprendizaje Intencional: Conocimiento previo

```
10 clear(X) :- not(on(_,X)).  
11 clear(table).
```

Aprendizaje Intencional: Algoritmo

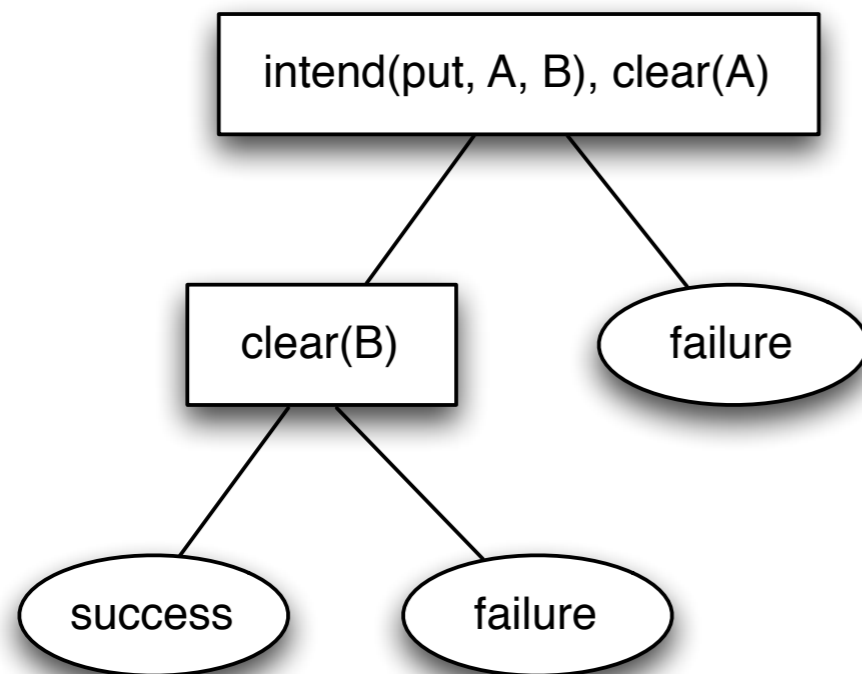
Algorithm 1 Top-down Induction of Logical Decision Trees.

```
1: procedure BUILDTREE( $E, Q$ )
2:    $\leftarrow Q_b := best(\rho(\leftarrow Q))$ 
3:   if  $stopCriteria(\leftarrow Q_b)$  then
4:      $T := leaf(majority\_class(E))$ 
5:   else
6:      $Conj \leftarrow Q_b \setminus Q$ 
7:      $E_1 \leftarrow \{e \in E \mid e \wedge B \models Q_b\}$ 
8:      $E_2 \leftarrow \{e \in E \mid e \wedge B \not\models Q_b\}$ 
9:      $buildTree(Left, E_1, Q_b)$ ;
10:     $buildTree(Right, E_2, Q)$ 
11:     $T \leftarrow nodei(Conj, Left, Right)$ 
12:   end if
13:   return  $T$ 
14: end procedure
```

▷ E is a set of examples, Q a query
▷ $best$ max information gain
▷ E.g., No information gain obtained

▷ The built tree

Aprendizaje Intencional: Lo Aprendido

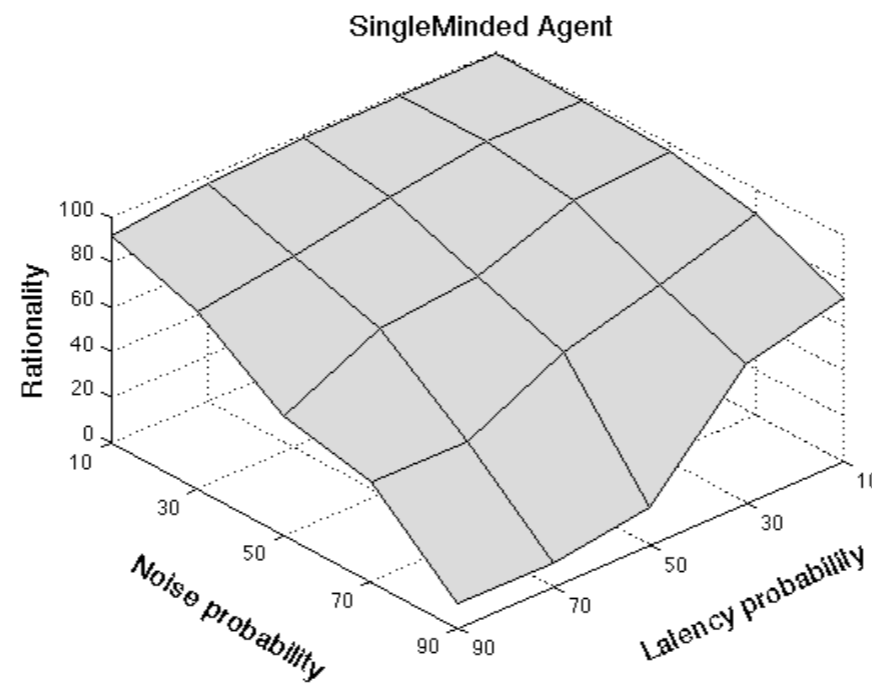
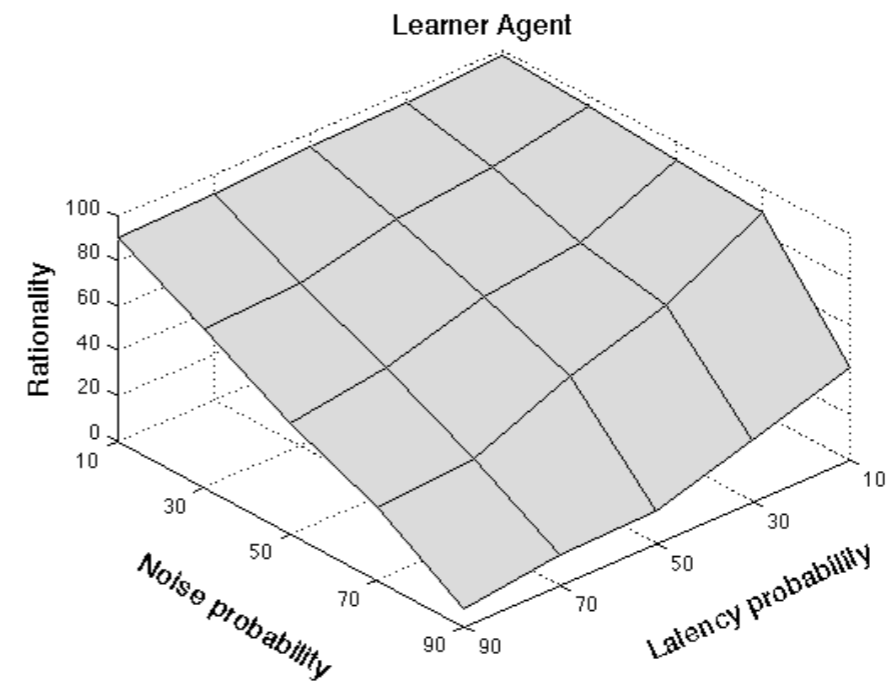
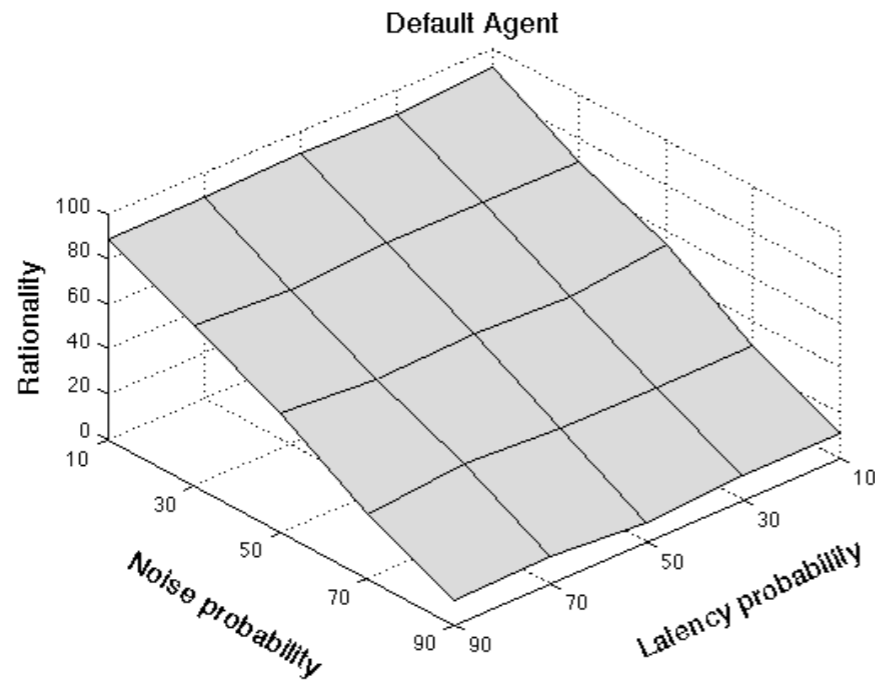


```
+!put(X,Y) : clear(X) & clear(Y)  
            <- move(X,Y) .
```

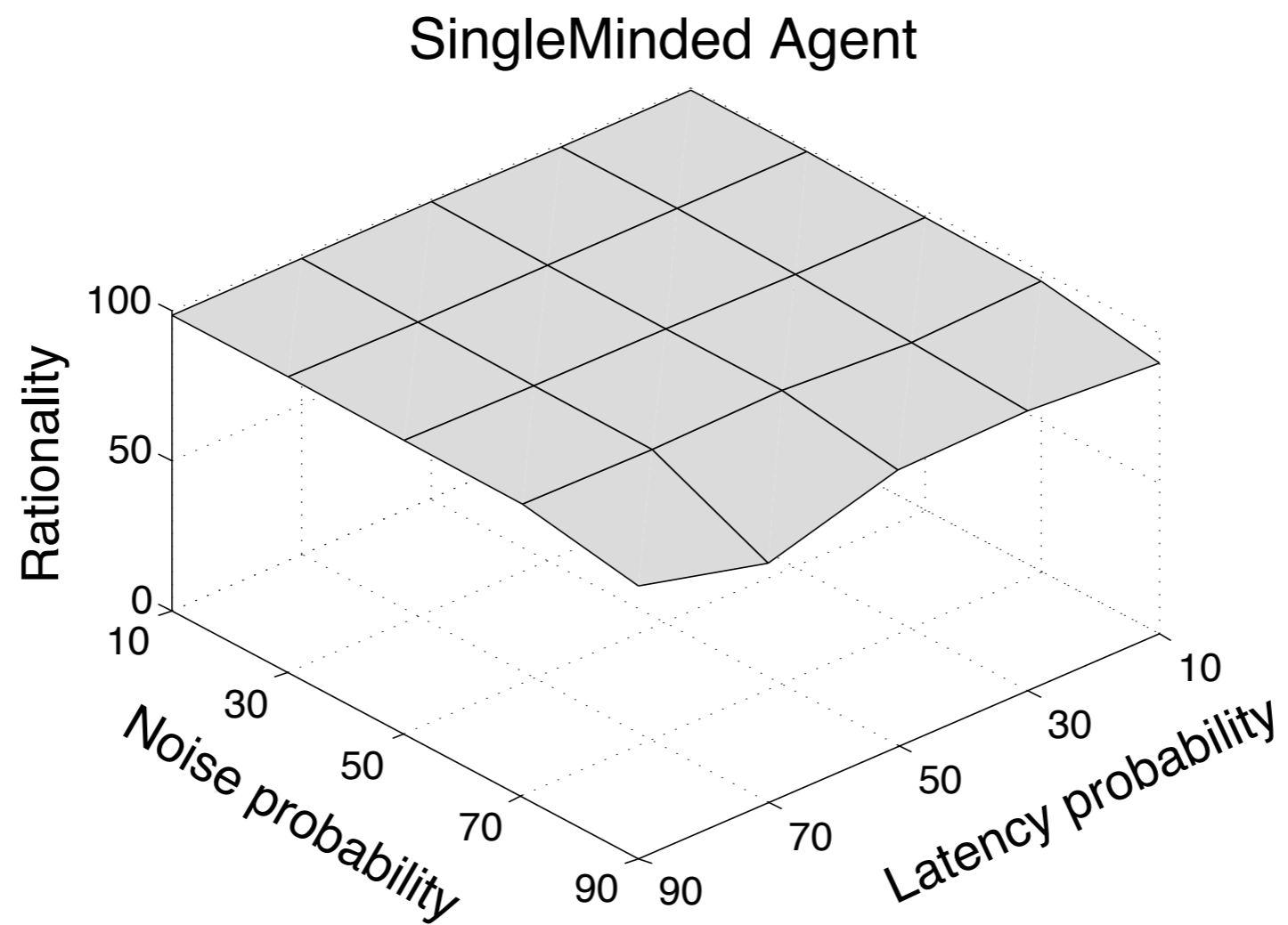
```
drop(put(X,Y)) :- .intend(put(X,Y)) & not clear(X) .
```

```
drop(put(X,Y)) :- .intend(put(X,Y)) & not clear(Y) .
```

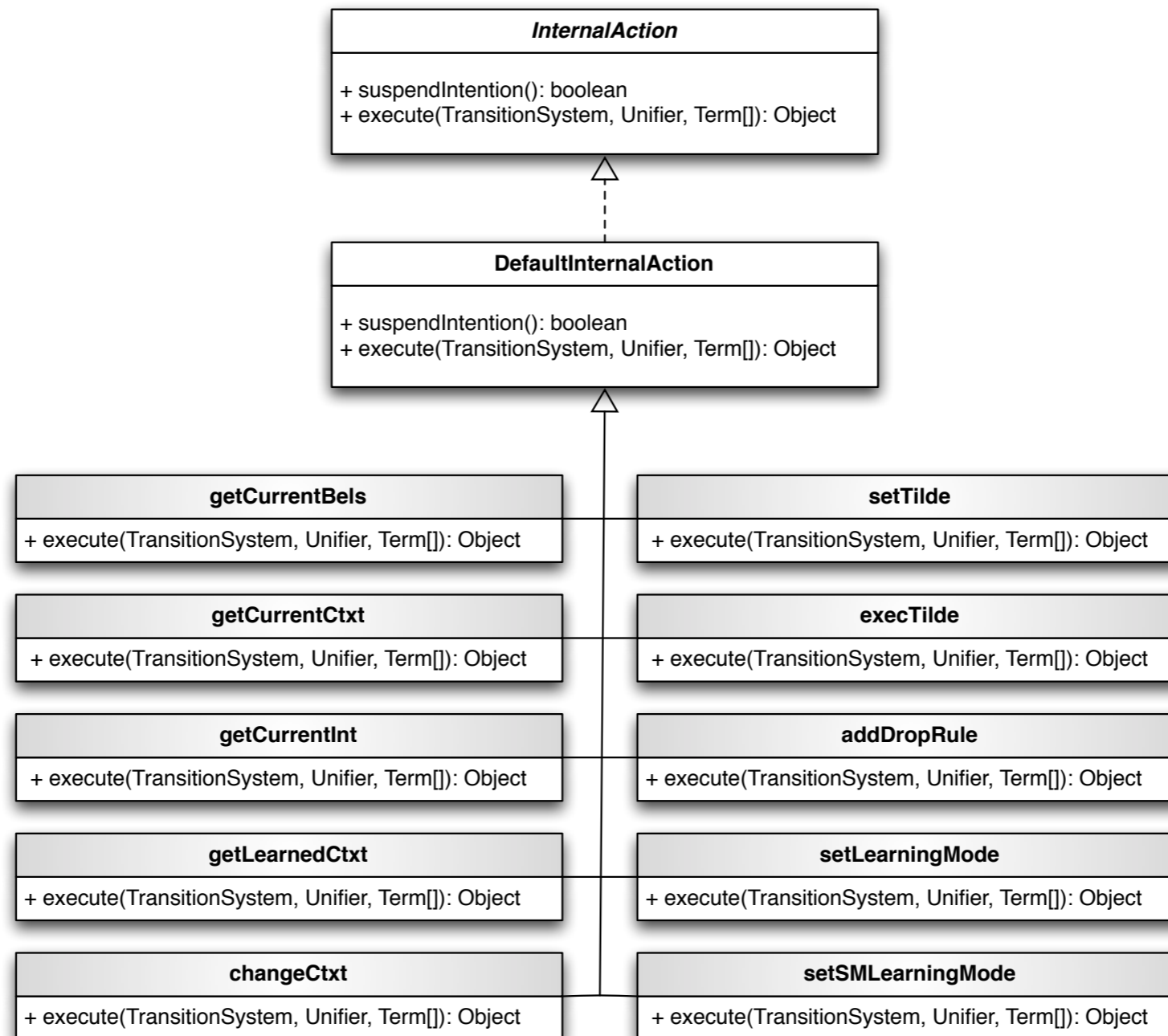
Resultados: Comparación entre agentes



Resultados: Con creencias al final.



Implementación (JILDT): Acciones Internas



Implementación (JILD T): Extensión de planes

```
@put  
+!put(X,Y) : true <-  
  jildt.getCurrentInt(I);  
  jildt.getCurrentBels(Bs);  
  +intending(I,Bs);  
  move(X,Y);  
  -intending(I,Bs);  
  +example(I,Bs,succ).
```

Implementación (JILDT): Cuando aprender?

```
@put_failCase  
-!put(X,Y) : intending(put(X,Y), Bs) <-  
  -intending(I,Bs);  
  +example(I,Bs,fail);  
  !learning(put);  
  +example_processed.
```

Implementación (JILDt): Aprendiendo

```
@learning  
+!learning(P): true <-  
  .print(\"Trying to learn a better context... \");  
  jildt.setTilde(P);  
  jildt.execTilde(false, false);  
  jildt.getLearnedCtxt(P, LC, F);  
  !learningTest(P, LC, F) .
```

Referencias

- H. Blockeel, L. D. Raedt, N. Jacobs, and B. Demoen. Scaling up inductive logic programming by learning from interpretations. *Data Mining and Knowledge Discovery*, 3(1):59–93, 1999.
- R. H. Bordini, J. F. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in Agent-Speak using Jason*. John Wiley & Sons Ltd, 2007.
- M. E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA., USA, and London, England, 1987.

Referencias

- A. Guerra-Hernández, A. El-Fallah-Seghrouchni, and H. Soldano. Learning in BDI Multi-agent Systems. CLIMA IV, Revised and Selected Papers, volume 3259 of LNCS, 218–233, Berlin Heidelberg, 2004. Springer-Verlag.
- A. Guerra-Hernández and G. Ortíz-Hernández. Toward BDI sapient agents: Learning intentionally. In *Toward Artificial Sapience: Principles and Methods for Wise Systems*, pages 77–91. Springer, London, 2008.
- A. Guerra-Hernández, J. M. Castro-Manzano, and A. El-Fallah-Seghrouchni. CTL AgentSpeak(L): a Specification Language for Agent Programs. *Journal of Algorithms*, (64):31–40, 2009.
- A. Guerra-Hernández, C. A. González-Alarcón, and A. E. F. Seghrouchni. Jason induction of logical decision trees. In MICAI 2010, Part I, volume 6437 of LNAI, 374–385, Berlin Heidelberg, 2010. Springer-Verlag.

Contacto

Dr. Alejandro Guerra Hernández

Profesor - SNI Nivel I - Perfil Deseable

Universidad Veracruzana
Facultad de Física e Inteligencia Artificial
Departamento de Inteligencia Artificial
Sebastián Camacho No. 5
Xalapa, Ver., México 91000
T (+52) 228 817-29-57
F (+52) 228 817-28-55

www.uv.mx/aguerra
aguerra@uv.mx • aguerrahdz@gmail.com