Verifying Resource Requirements for Distributed Rule-Based Systems

Natasha Alechina, Brian Logan
Nguyen Hoang Nga, and Abdur Rakib
The University of Nottingham

RuleML-2008

Orlando, Florida: October 30-31, 2008

Outline

- Rule based system
- Challenges to application developers
- Distributed rule based system
- Resources and Actions
- Measuring resources
- The model of distributed rule-based systems
- Example
- Verifying resource bounds
- Experimental results

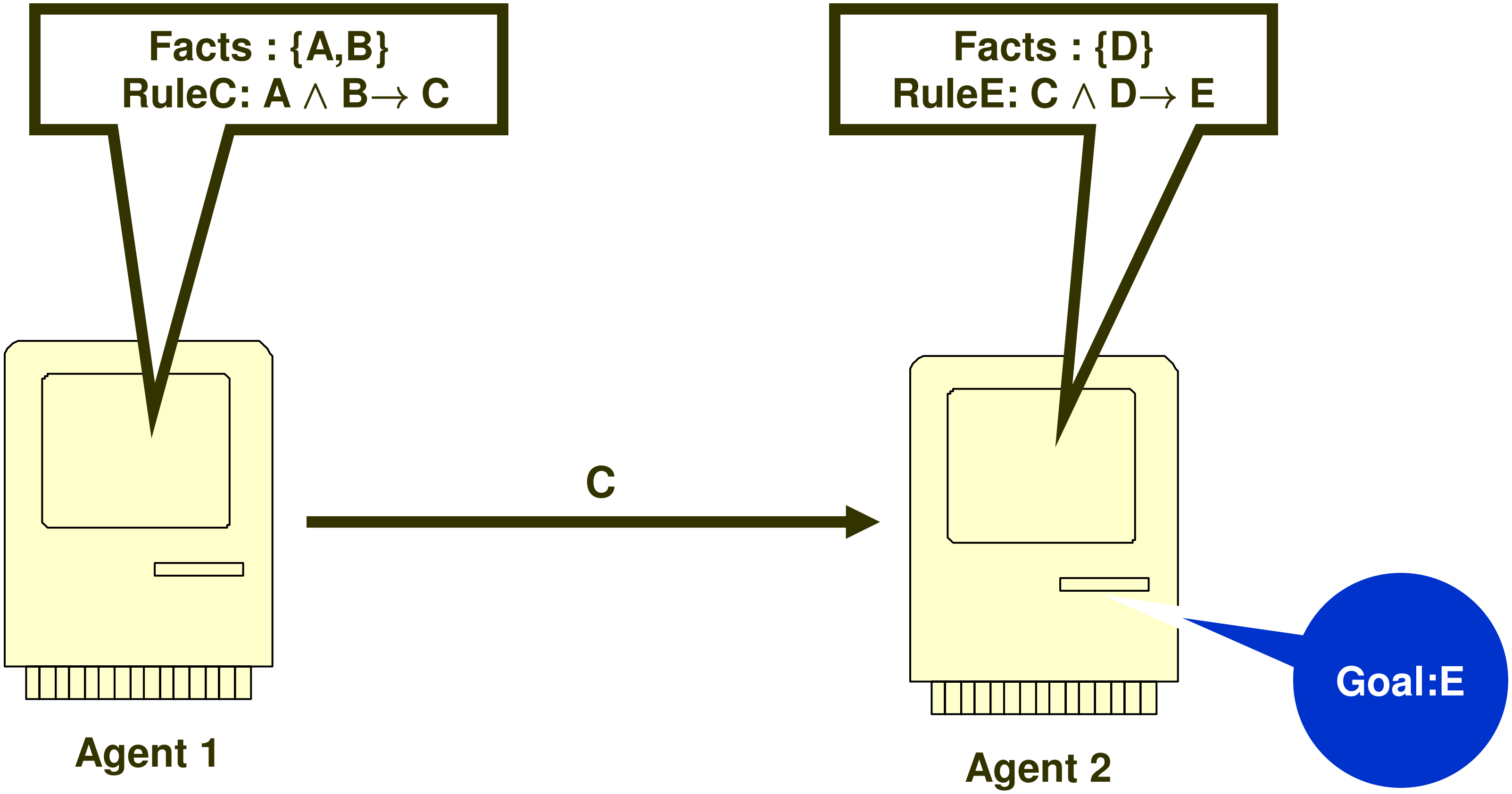
Rule based system

- A rule based reasoning system is a particular type of reasoning system.
- The system mainly consist of three parts :
  - A set of condition-action rules specifying which action(s) to perform when a given condition is true;
  - A set of facts which constitute the current state of the system;
  - A rule engine which matches the rule conditions against the facts and fires those rules which match.

Challenges to application developers

- How to ensure the following properties of rule-based system designs
  - *correctness* : will a rule-based system produce the correct output for all legal inputs;
  - *termination :* will a rule-based system produce an output at all;
  - *response time :* how much computation will a rule-based system have to do before it generates an output.
- These problems become even more challenging in the case of *distributed rule-based systems*.

Distributed rule based system



Facts : {A,B}
RuleC: $A \wedge B \rightarrow C$

Facts : {D}
RuleE: $C \wedge D \rightarrow E$

C

Goal:E

Agent 1

Agent 2

Resources and Actions

Time : How many inference steps does the system need to perform, in parallel ?

Communication : How many messages do the agents need to exchange ?

Actions

Rule : If *antecedents* of a rule are present in agent's working memory but c*onsequent* is not in a state *s* then consequent will be added to the agent's working memory in the successor state upon firing that rule;

Copy : Agent can copy facts from other agents memory, if it is not present in its working memory;

Idle : Agent leaves its configuration unchanged.

Resources and Actions contd.

Time	Agent1	Agent2	#Messages
t_0	{A,B}	{D}	0
Operation:	RuleC	Idle	0
t_1	{A,B,C}	{D}	0
Operation:	Idle	Copy	1
t_2	{A,B,C}	{C,D}	1
Operation:	Idle	RuleE	1
t_3	{A,B,C}	{C,D,E}	1

Measuring resources

- A problem is considered to be solved if one of the agents has derived the goal
- We take the time complexity of a derivation to be the total number of steps by the system
- Our model of communication complexity is based on the number of facts exchanged by the agents
- The communication complexity of a joint derivation is then the (total) number of Copy operations in the derivation.

The model of Communicating rule-based systems

The framework is based on L_{CRB}

- **Let $A = \{1,\ldots,n_{Ag}\}$ be the set of all agents, and P a finite common alphabet of facts.**
- **Let P_i be a finite set of rules of the form $p_1 \wedge \ldots \wedge p_n \rightarrow p$, where $n \geq 0$, $p_i, p \in P$ for all $i \in \{1,\ldots,n\}$ and $p_i \neq p_j$ for all $i \neq j$.**
- **Let $cp_i^{=n}$ denotes that the value of agent i's communication counter is n for all $n \in \{0,\ldots,n_C(i)\}$ and $i \in A$, where $n_C(i)$ is the upper bound of Copy action that agent i can perform.**

The model of Communicating rule-based systems contd.

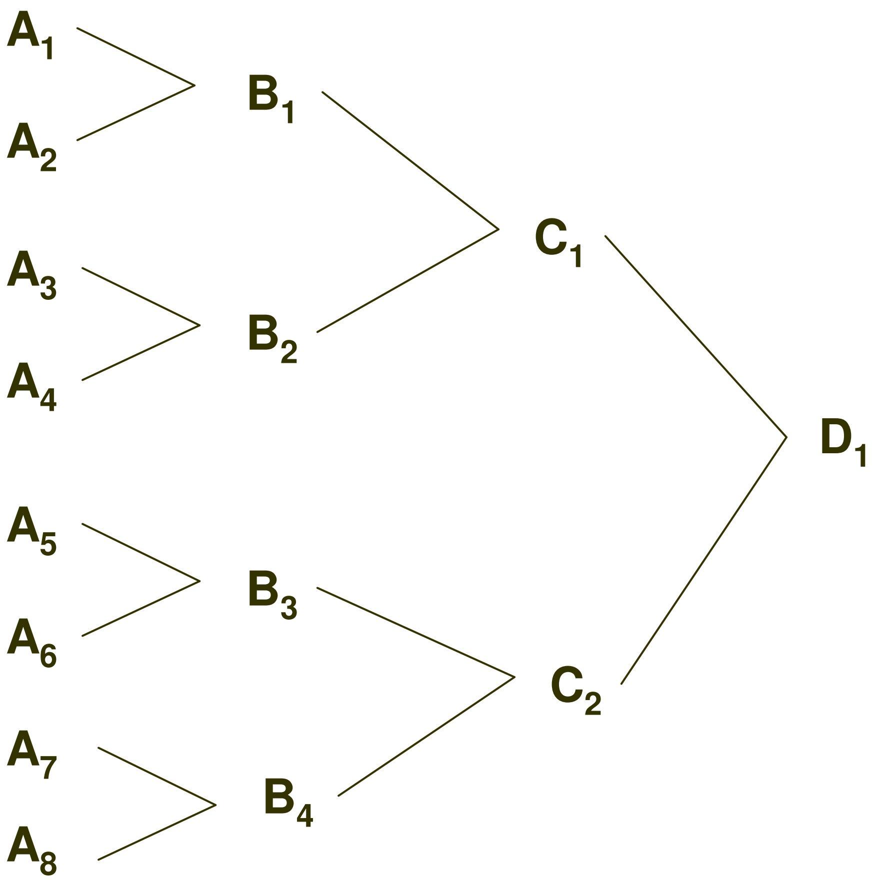
- The syntax of L_{CRB} includes the temporal operators of CTL with belief operators and communication counters

$$\phi \; ::= \top \mid cp_i^{=n} \mid B_i p \mid B_i \rho \mid \neg \phi \mid \phi \wedge \psi \mid X \phi \mid \phi \, U \, \psi \mid A \phi$$

Other classical abbreviations for ⊥, ∨ , → and ↔, and temporal operations are as usual.

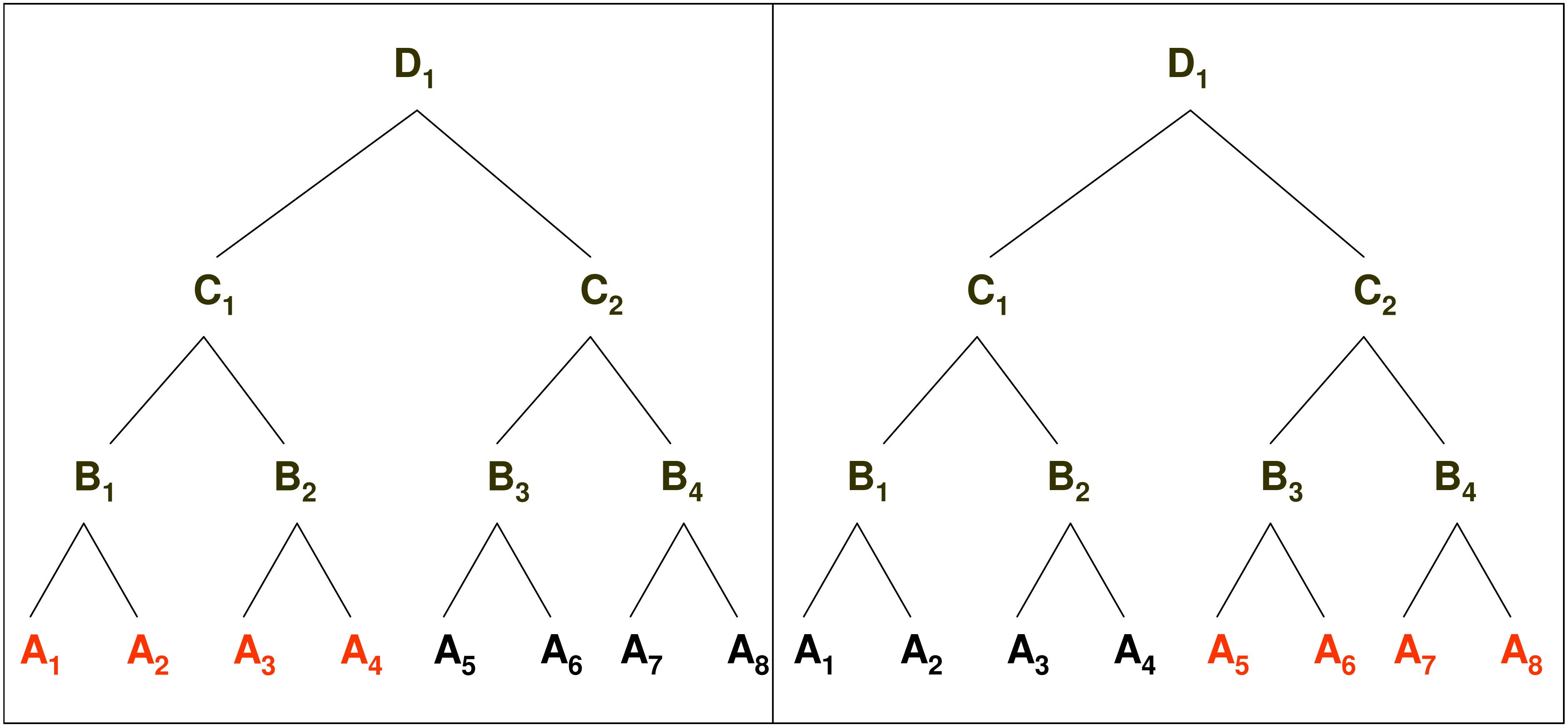
- The semantics of L_{CRB} is defined by L_{CRB} transition systems which are based on ω-tree structures.

Example

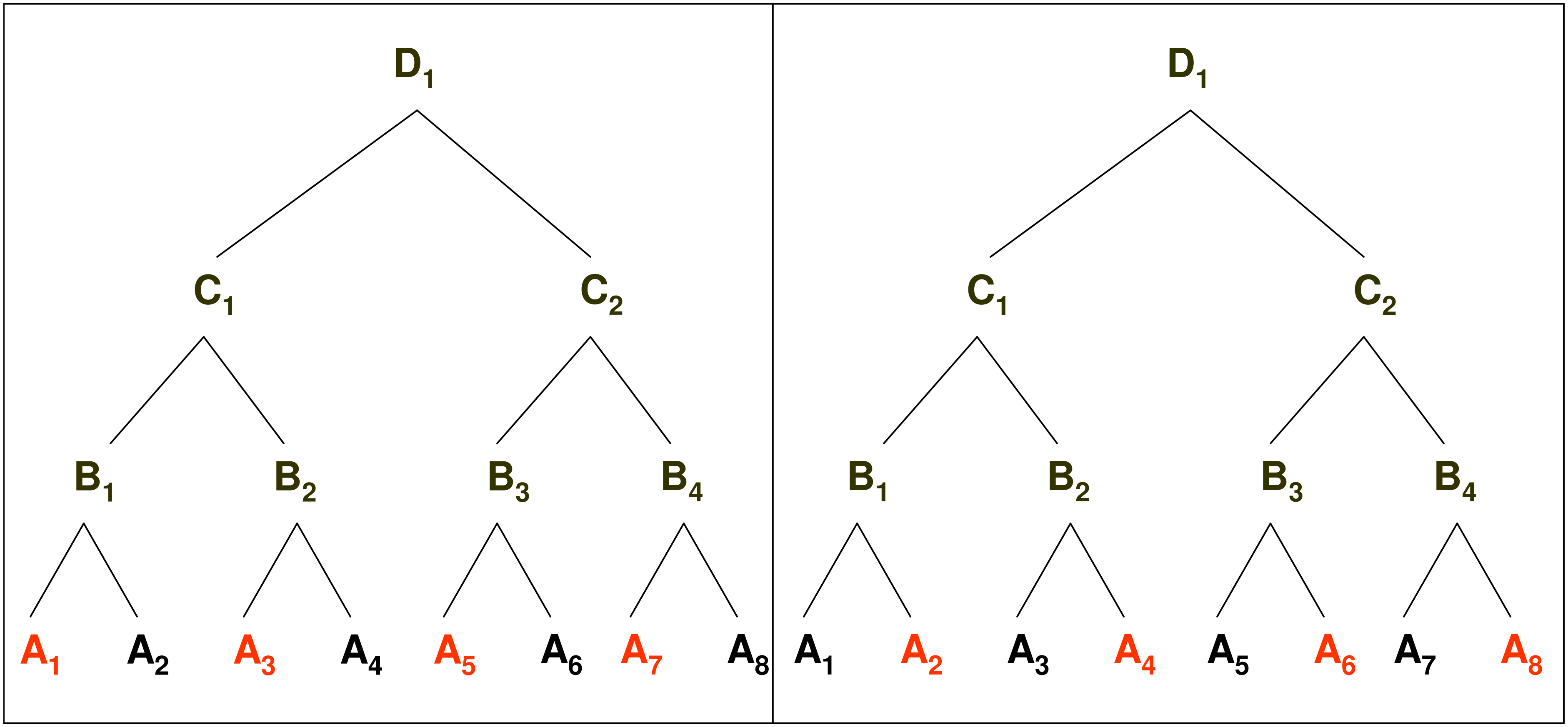


$\text{Rule}B_1$	$A_1 \wedge A_2 \rightarrow B_1$
$\text{Rule}B_2$	$A_3 \wedge A_4 \rightarrow B_2$
$\text{Rule}B_3$	$A_5 \wedge A_6 \rightarrow B_3$
$\text{Rule}B_4$	$A_7 \wedge A_8 \rightarrow B_4$
$\text{Rule}C_1$	$B_1 \wedge B_2 \rightarrow C_1$
$\text{Rule}C_2$	$B_3 \wedge B_4 \rightarrow C_2$
$\text{Rule}D_1$	$C_1 \wedge C_2 \rightarrow D_1$

Example contd.



Example contd.



Example contd.

#TimeStep	Agent 1	Agent 2	#Messages
1	{A1,A2,A3,A4}	{A5,A6,A7,A8}	0,0
operation:	Rule B2	Rule B4	
2	{A1,A2,A3,A4, B2}	{A5,A6,A7,A8,B4}	0,0
operation:	Rule B1	Rule B3	
3	{A1,A2,A3,A4, B2,B1}	{A5,A6,A7,A8,B4, B3}	0,0
operation:	Rule C1	Rule C2	
4	{A1,A2,A3,A4, B2,B3, C1}	{A5,A6,A7,A8,B4, B3, C2}	0,0
operation:	Idle	Copy (C1 from Agent 1)	
5	{A1,A2,A3,A4, B2,B3, C1}	{A5,A6,A7,A8,B4, B3, C2, C1}	0,1
operation:	Idle	Rule D1	
6	{A1,A2,A3,A4, B2,B3, C1}	{A5,A6,A7,A8,B4, B3, C2, C1, D1}	0,1

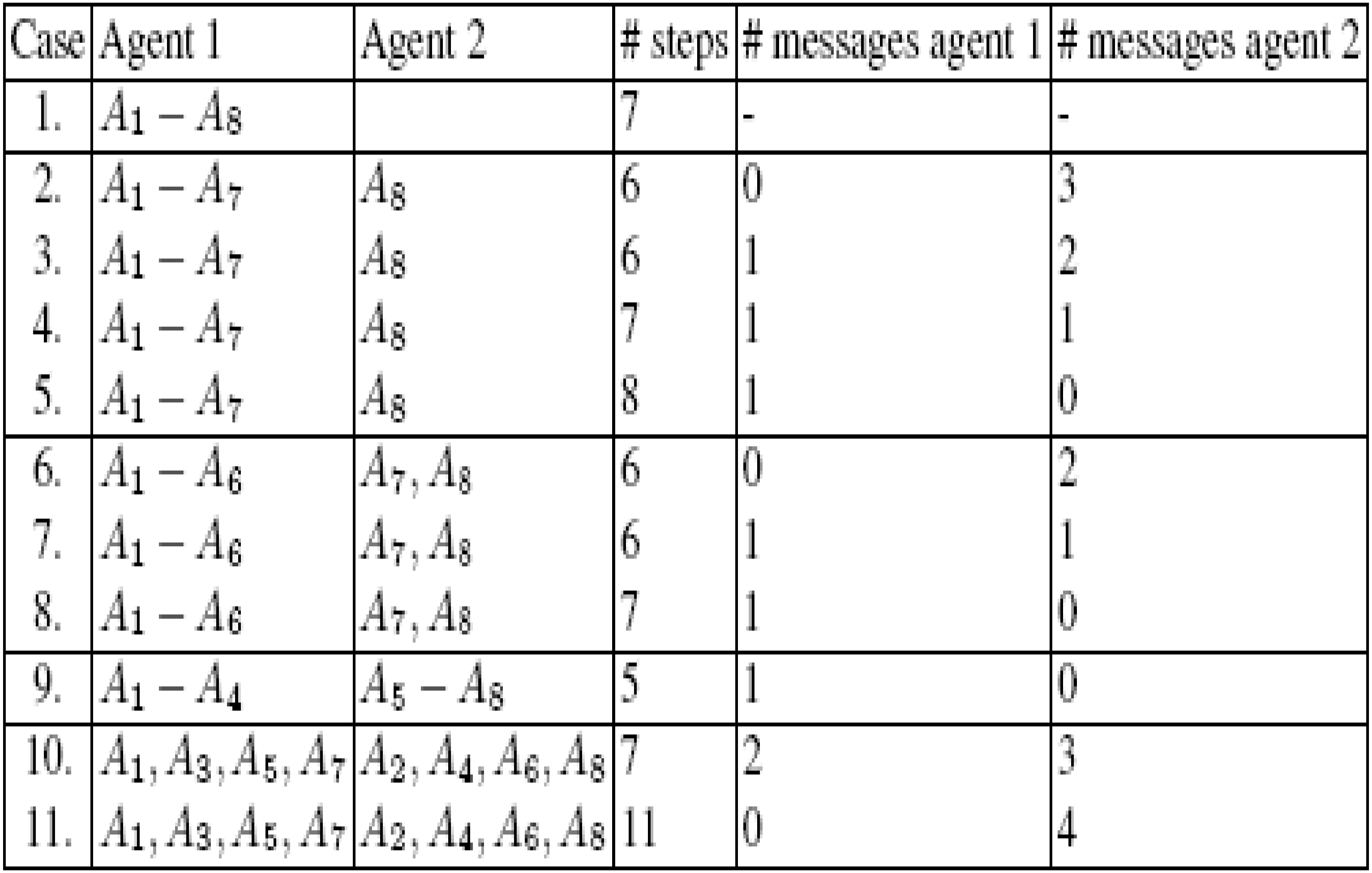
Verifying resource bounds : Model checking (MOCHA)

- **The specification language of Mocha is *ATL*, which includes *CTL*.**
- **We can express properties such as**

  'agent *i* may derive belief α in *n* steps' as $EX^n\, tr(B_i\alpha)$

  where $tr(B_i\alpha)$ is a state variable encoding of the fact that is present in the agent's working memory.
- **To obtain the actual derivation, we can verify an invariant which states that $tr(B_i\alpha)$ is never true, and use the counterexample trace to show how the system reaches the state where α is proved.**
- **To bound the number of messages used, we can include a bound on the value of the message counter of one or more agents in the property to be verified.**

Experimental results



Case	Agent 1	Agent 2	# steps	# messages agent 1	# messages agent 2
1.	$A_1 - A_8$		7	-	-
2.	$A_1 - A_7$	A_8	6	0	3
3.	$A_1 - A_7$	A_8	6	1	2
4.	$A_1 - A_7$	A_8	7	1	1
5.	$A_1 - A_7$	A_8	8	1	0
6.	$A_1 - A_6$	A_7, A_8	6	0	2
7.	$A_1 - A_6$	A_7, A_8	6	1	1
8.	$A_1 - A_6$	A_7, A_8	7	1	0
9.	$A_1 - A_4$	$A_5 - A_8$	5	1	0
10.	A_1, A_3, A_5, A_7	A_2, A_4, A_6, A_8	7	2	3
11.	A_1, A_3, A_5, A_7	A_2, A_4, A_6, A_8	11	0	4

Resource requirements for optimal derivation in 8 leaves cases

Case	Agent 1	Agent 2	# steps	# copy 1	# copy 2
1.	$A_1 - A_{16}$		15	-	-
2.	$A_1 - A_{15}$	A_{16}	12	0	6
3.	$A_1 - A_{15}$	A_{16}	12	1	4
4.	$A_1 - A_{15}$	A_{16}	13	1	3
5.	$A_1 - A_{15}$	A_{16}	14	1	2
6.	$A_1 - A_{15}$	A_{16}	15	1	1
7.	$A_1 - A_{15}$	A_{16}	16	1	0
8.	$A_1 - A_{14}$	A_{15}, A_{16}	11	0	5
9.	$A_1 - A_{14}$	A_{15}, A_{16}	11	1	4
10.	$A_1 - A_{14}$	A_{15}, A_{16}	12	1	3
11.	$A_1 - A_{14}$	A_{15}, A_{16}	13	1	2
12.	$A_1 - A_{14}$	A_{15}, A_{16}	14	1	1
13.	$A_1 - A_{14}$	A_{15}, A_{16}	15	1	0
14.	$A_1 - A_{12}$	$A_{13}, A_{14}, A_{15}, A_{16}$	11	0	4
15.	$A_1 - A_{12}$	$A_{13}, A_{14}, A_{15}, A_{16}$	11	1	2
16.	$A_1 - A_{12}$	$A_{13}, A_{14}, A_{15}, A_{16}$	12	1	1
17.	$A_1 - A_{12}$	$A_{13}, A_{14}, A_{15}, A_{16}$	13	1	0
18.	$A_1 - A_3, A_5 - A_7, A_9 - A_{11}, A_{13} - A_{15}$	A_4, A_8, A_{12}, A_{16}	13	2	6
19.	$A_1 - A_3, A_5 - A_7, A_9 - A_{11}, A_{13} - A_{15}$	A_4, A_8, A_{12}, A_{16}	19	4	0
20.	$A_1, A_3, A_5, A_7, A_9, A_{11}, A_{13}, A_{15}$	$A_2, A_4, A_6, A_8, A_{12}, A_{14}, A_{16}$	13	4	5
21.	$A_1, A_3, A_5, A_7, A_9, A_{11}, A_{13}, A_{15}$	$A_2, A_4, A_6, A_8, A_{12}, A_{14}, A_{16}$	23	0	8

Conclusions

- We analyze the time and communication resources required by a system of rule-based reasoning agents to achieve a goal
- We show how L_{CRB} transition systems can be encoded as input to the Mocha model-checker and how properties can be verified automatically
- We described results of some experiments on a synthetic example which show interesting trade-offs between time required by the agents to solve the problem and the number of messages they need to exchange.

Thanks!