Abstract Machines for Argumentation

Logic and Interactions 2012, Week 2

Kurt Ranalter

SIAG Bolzano/Bozen

CIRM, 10/02/2012

Overview

Introduction

Summary of content

- Related work and motivations
- Aim of talk and contributions

Introduction

Summary of content

- Related work and motivations
- Aim of talk and contributions

Lecomte and Quatrini

- *Ludics and its applications to natural language semantics* (in LNAI 5514, 2009)
- A theory of meaning that is based on ludics
  - convergence via daimon
  - meaning via orthogonality
- Match between rules of ludics and moves in dialogue
  - rules of ludics: positive vs negative
  - roles in dialogue: speaker vs hearer
  - actions in dialogue: sender vs receiver
- Put these aspects together by means of normalisation

Introduction

Curien and Herbelin

- *Abstract machines for dialogue games* (in Panoramas et Synthèses 27, 2009)
- Proofs in ludics regarded as abstract Böhm trees
- Various abstract machines for computing with ABTs

Introduction

Curien and Herbelin

- *Abstract machines for dialogue games* (in Panoramas et Synthèses 27, 2009)
- Proofs in ludics regarded as abstract Böhm trees
- Various abstract machines for computing with ABTs

Combining these strands

- Want to extend duality to abstract Böhm trees
  - rules of ludics: positive vs negative
  - roles in dialogue: speaker vs hearer
  - actions in dialogue: sender vs receiver
  - abstract Böhm trees: replies vs queries
- Towards computational account for modelling dialogue
  - normalisation by means of geometric abstract machine
- ABTs more expressive than MLL-based variant of ludics

Introduction

Basaldella and Faggian

- *Ludics with repetitions: exponentials, interactive types and completeness* (in LMCS 7, 2011)
- An extension of ludics that deals with exponentials
- Add pointers to trace occurrences of subformulae

Introduction

Basaldella and Faggian

- *Ludics with repetitions: exponentials, interactive types and completeness* (in LMCS 7, 2011)
- An extension of ludics that deals with exponentials
- Add pointers to trace occurrences of subformulae

Relation to our framework

- Relevant differences mostly of technical nature
  - normalisation via view abstract machine
  - pointer interaction not a primary concern
  - main focus on repetition of actions
- Should be possible to translate all of our examples
- Pointer interaction one of the central topics of this talk

Abstract machines

Summary of content

- Sketch of formal definitions
- How does GAM actually work?

Abstract machines

Summary of content

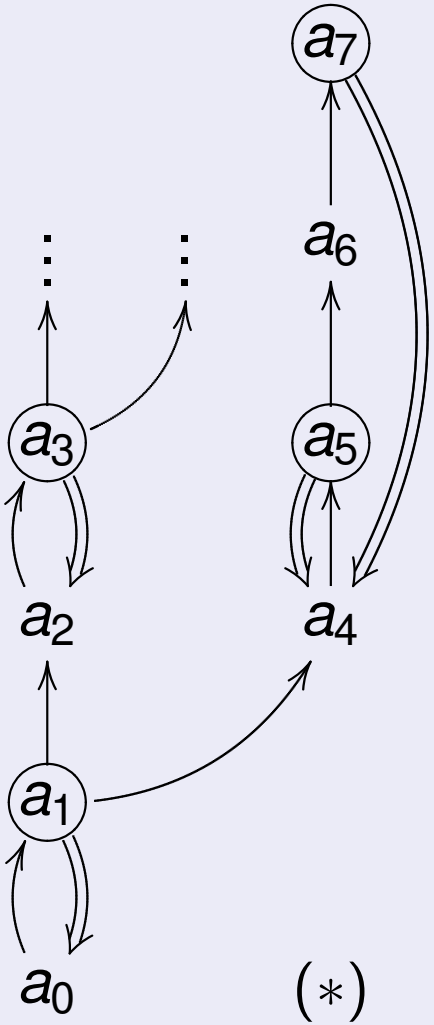
- Sketch of formal definitions
- How does GAM actually work?

General considerations

- Operational account of concepts from game semantics
- Crisp graphical representation for abstract Böhm trees
  - interaction may be seen as interleaved tree traversal
  - graphical representation vs concrete implementation
- Small number of rules leads to compact implementation
- Rapid prototyping as main benefit of implementation
  - a potential framework for developing applications
  - why not abstract Böhm trees as data structures?

Abstract machines

Abstract Böhm trees



- Two types of moves
  - queries: a_0, a_2, a_4, a_6
  - replies: a_1, a_3, a_5, a_7
- Pointer conditions
  - from reply to query
  - only within branch
- Branching condition
  - only after replies
- (Counter-)strategies
  - $(*)$ is counterstrategy
  - strategy when 1) $a_0 = \star$ and 2) no pointers to $\star$

Abstract machines

Geometric abstract machine

$$\frac{}{\mapsto \{1 \leftarrow \star\}}\,(1) \qquad \frac{hd(\Gamma) = \{\overline{2n} \leftarrow \mathbf{q}[a, -]\}}{\Gamma \mapsto \Gamma\{2n \leftarrow a\}}\,(2n)_f$$

$$\frac{hd(\Gamma) = \{2n-1 \leftarrow \mathbf{q}\},\ \phi(\mathbf{q}) = [a, \kappa]}{\Gamma \mapsto \Gamma\{\overline{2n} \leftarrow \mathbf{q}[a, \kappa]\}}\,(\overline{2n})$$

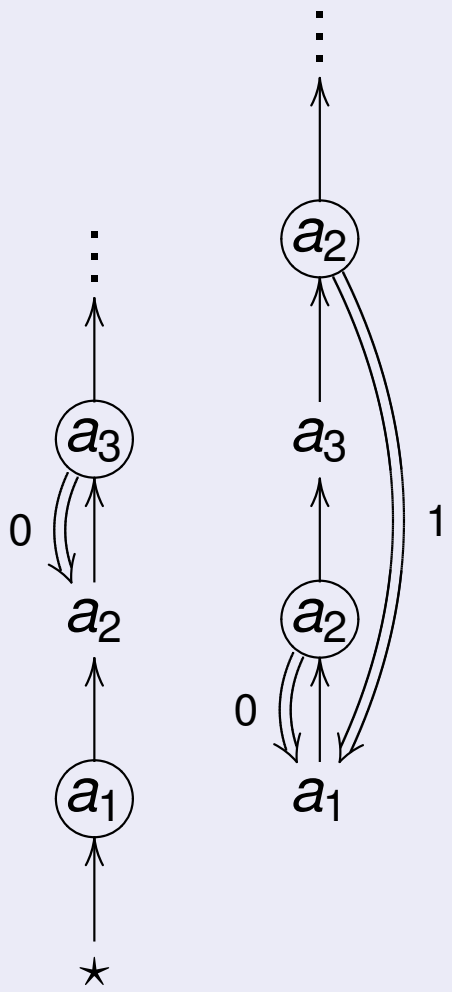
$$\frac{hd(\Gamma) = \{\overline{2n} \leftarrow \mathbf{q}[a, \iota]\},\ \pi(pop^{\iota}(\mathbf{q})) = 2k-1,\ \Gamma \bullet \overline{2k-1} = \mathbf{r}}{\Gamma \mapsto \Gamma\{2n \leftarrow \mathbf{r}a\}}\,(2n)_b$$

$$\frac{hd(\Gamma) = \{2n \leftarrow \mathbf{q}\},\ \psi(\mathbf{q}) = [a, \kappa]}{\Gamma \mapsto \Gamma\{\overline{2n+1} \leftarrow \mathbf{q}[a, \kappa]\}}\,(\overline{2n+1})$$

$$\frac{hd(\Gamma) = \{\overline{2n+1} \leftarrow \mathbf{q}[a, \iota]\},\ \pi(pop^{\iota}(\mathbf{q})) = 2k,\ \Gamma \bullet \overline{2k} = \mathbf{r}}{\Gamma \mapsto \Gamma\{2n \leftarrow \mathbf{r}a\}}\,(2n+1)$$

Abstract machines

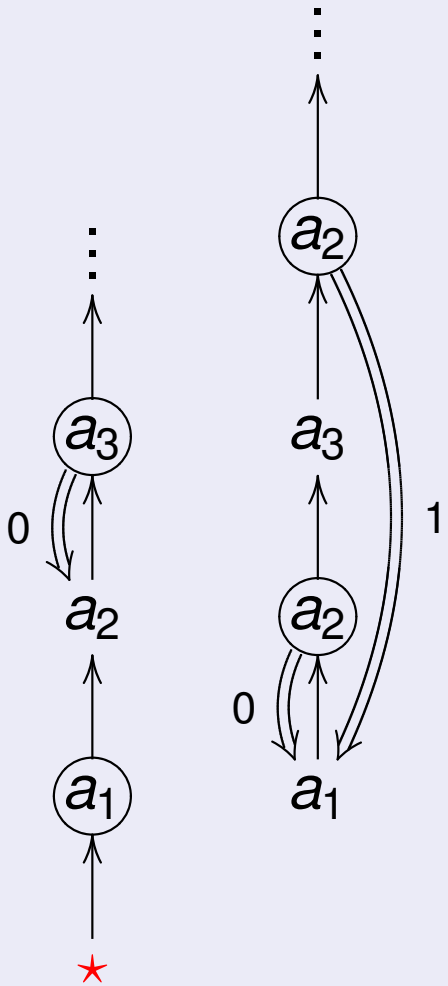
GAM at work: outline




```
1 *
2̄ *[a1,-]
2 a1
3̄ a1[a2,0]
3 *[a1,-]a2
4̄ *[a1,-]a2[a3,0]
4 a1[a2,0]a3
5̄ a1[a2,0]a3[a2,1]
5 *[a1,-]a2
6̄ *[a1,-]a2[a3,0]
6 a1[a2,0]a3[a2,1]···
```

Abstract machines

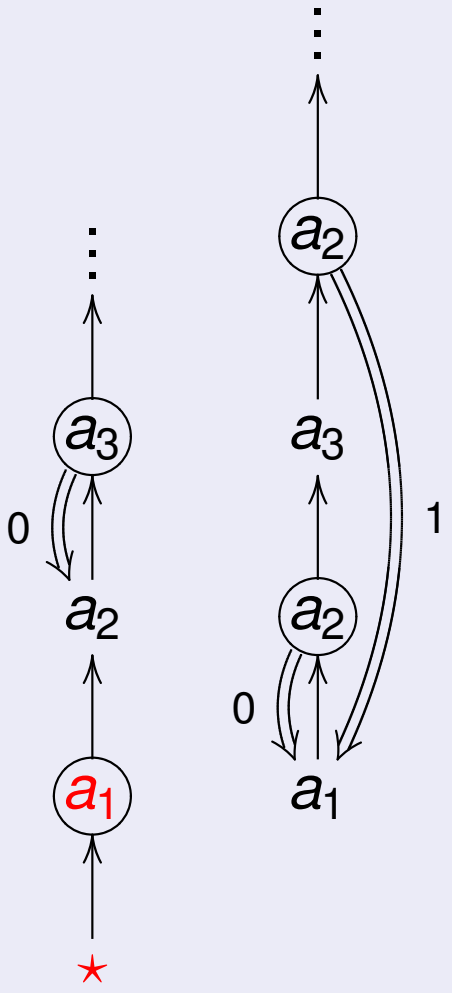
GAM at work: step 1



1	$\star$
$\overline{2}$	`*[a1,-]`
2	`a1`
$\overline{3}$	`a1[a2,0]`
3	`*[a1,-]a2`
$\overline{4}$	`*[a1,-]a2[a3,0]`
4	`a1[a2,0]a3`
$\overline{5}$	`a1[a2,0]a3[a2,1]`
5	`*[a1,-]a2`
$\overline{6}$	`*[a1,-]a2[a3,0]`
6	`a1[a2,0]a3[a2,1]`$\cdots$

Abstract machines

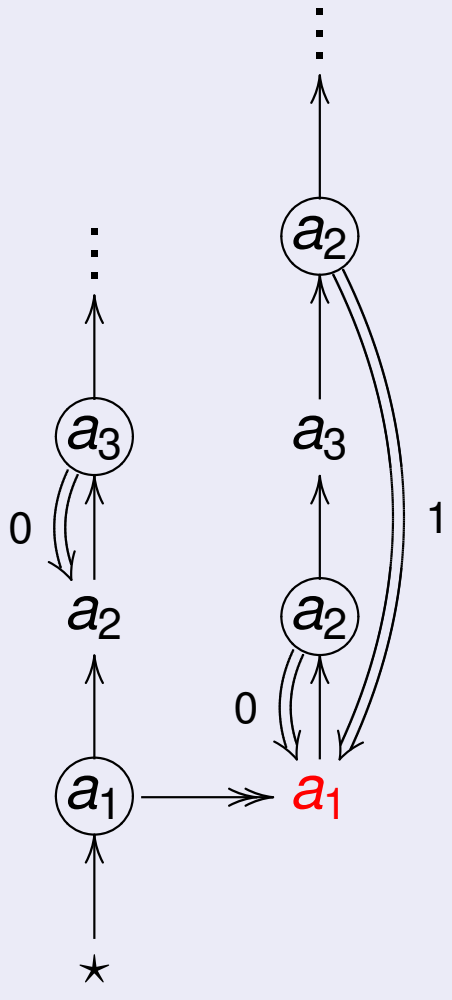
GAM at work: step $\overline{2}$



$$
\begin{array}{ll}
1 & \star \\
\overline{2} & \star[a1,-] \\
2 & a1 \\
\overline{3} & a1[a2,0] \\
3 & \star[a1,-]a2 \\
\overline{4} & \star[a1,-]a2[a3,0] \\
4 & a1[a2,0]a3 \\
\overline{5} & a1[a2,0]a3[a2,1] \\
5 & \star[a1,-]a2 \\
\overline{6} & \star[a1,-]a2[a3,0] \\
6 & a1[a2,0]a3[a2,1]\cdots
\end{array}
$$

Abstract machines

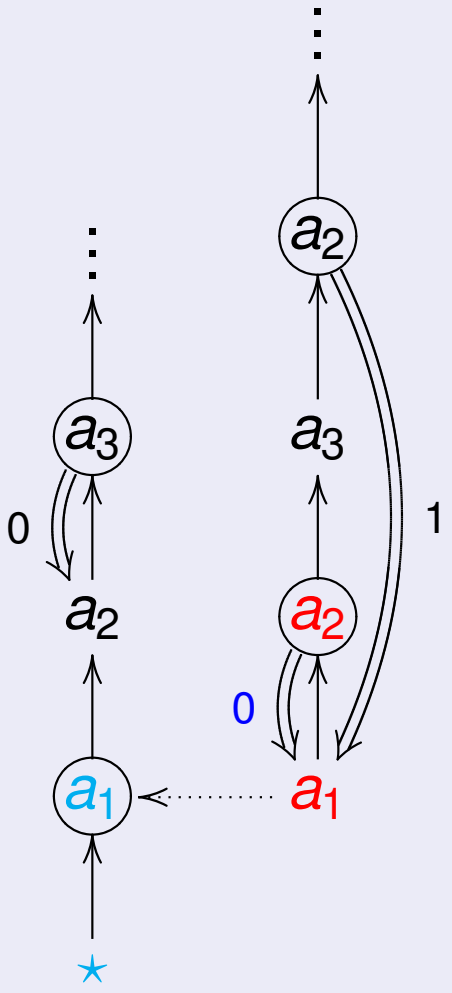
GAM at work: step 2



1 ⋆
2 ⋆[a1,-]
2 a1
3 a1[a2,0]
3 ⋆[a1,-]a2
4 ⋆[a1,-]a2[a3,0]
4 a1[a2,0]a3
5 a1[a2,0]a3[a2,1]
5 ⋆[a1,-]a2
6 ⋆[a1,-]a2[a3,0]
6 a1[a2,0]a3[a2,1]⋯

Abstract machines

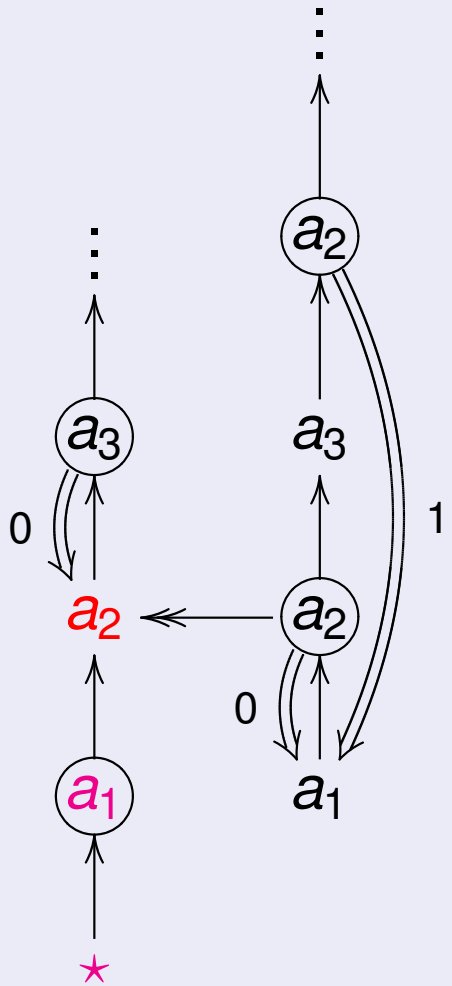
GAM at work: step $\overline{3}$



$$
\begin{array}{ll}
1 & \star \\
\overline{2} & \star[a1,-] \\
2 & a1 \\
\overline{3} & a1[a2,0] \\
3 & \star[a1,-]a2 \\
\overline{4} & \star[a1,-]a2[a3,0] \\
4 & a1[a2,0]a3 \\
\overline{5} & a1[a2,0]a3[a2,1] \\
5 & \star[a1,-]a2 \\
\overline{6} & \star[a1,-]a2[a3,0] \\
6 & a1[a2,0]a3[a2,1]\cdots
\end{array}
$$

Abstract machines

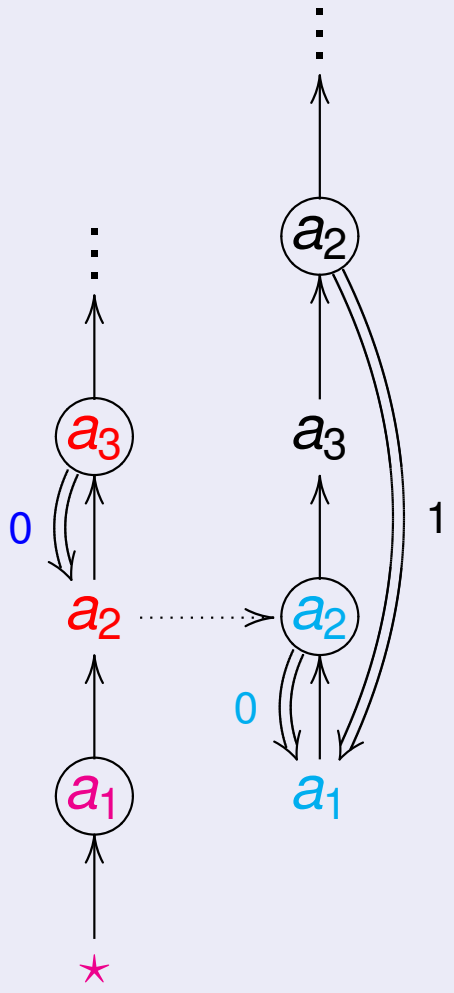
GAM at work: step 3



1	$\star$
$\overline{2}$	$\star$[a1,−]
2	a1
$\overline{3}$	a1[a2,0]
3	$\star$[a1,−]a2
$\overline{4}$	$\star$[a1,−]a2[a3,0]
4	a1[a2,0]a3
$\overline{5}$	a1[a2,0]a3[a2,1]
5	$\star$[a1,−]a2
$\overline{6}$	$\star$[a1,−]a2[a3,0]
6	a1[a2,0]a3[a2,1]$\cdots$

Abstract machines

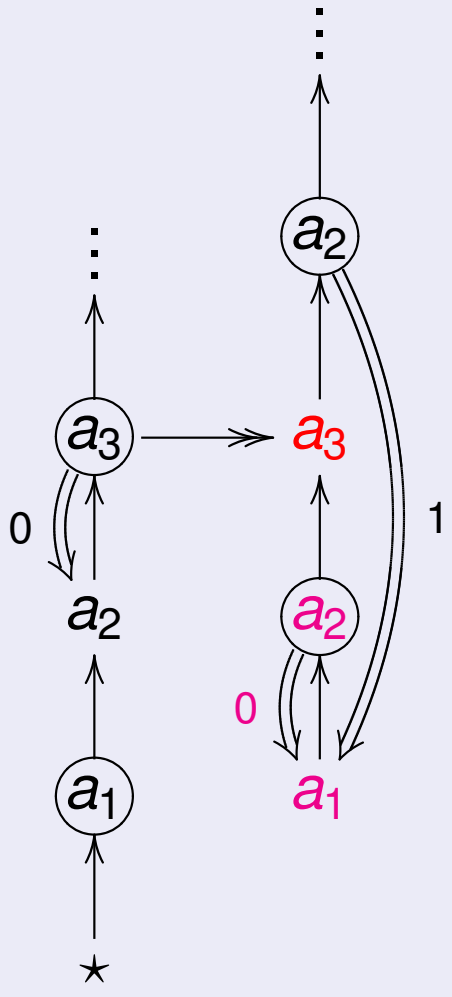
GAM at work: step $\overline{4}$



$$
\begin{array}{ll}
1 & \star \\
\overline{2} & \star[a1,-] \\
2 & a1 \\
\overline{3} & a1[a2,0] \\
3 & \star[a1,-]a2 \\
\overline{4} & \star[a1,-]a2[a3,0] \\
4 & a1[a2,0]a3 \\
\overline{5} & a1[a2,0]a3[a2,1] \\
5 & \star[a1,-]a2 \\
\overline{6} & \star[a1,-]a2[a3,0] \\
6 & a1[a2,0]a3[a2,1]\cdots
\end{array}
$$

Abstract machines

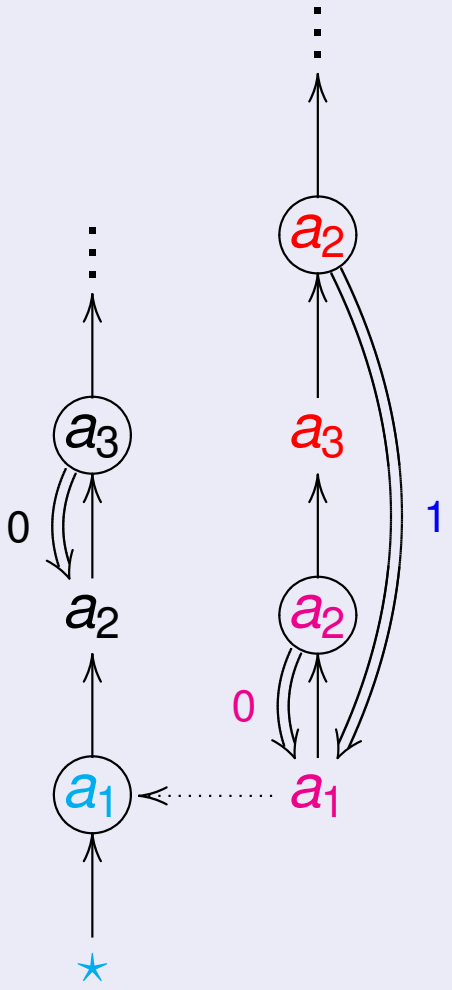
GAM at work: step 4



1 $\star$
$\overline{2}$ $\star$[a1,-]
2 a1
$\overline{3}$ a1[a2,0]
3 $\star$[a1,-]a2
$\overline{4}$ $\star$[a1,-]a2[a3,0]
4 a1[a2,0]a3
$\overline{5}$ a1[a2,0]a3[a2,1]
5 $\star$[a1,-]a2
$\overline{6}$ $\star$[a1,-]a2[a3,0]
6 a1[a2,0]a3[a2,1]$\cdots$

Abstract machines

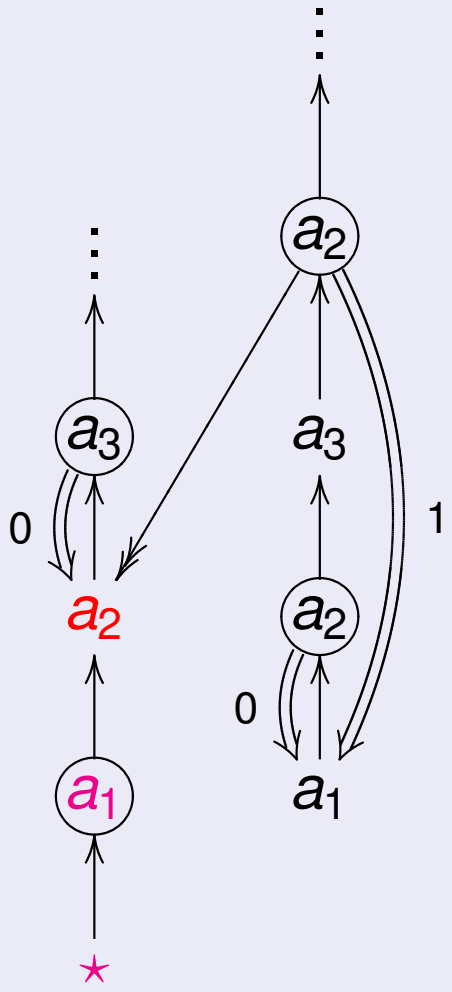
GAM at work: step $\overline{5}$



$$
\begin{array}{ll}
1 & \star \\
\overline{2} & \star[a1,-] \\
2 & a1 \\
\overline{3} & a1[a2,0] \\
3 & \star[a1,-]a2 \\
\overline{4} & \star[a1,-]a2[a3,0] \\
4 & a1[a2,0]a3 \\
\overline{5} & a1[a2,0]a3[a2,1] \\
5 & \star[a1,-]a2 \\
\overline{6} & \star[a1,-]a2[a3,0] \\
6 & a1[a2,0]a3[a2,1]\cdots
\end{array}
$$

Abstract machines

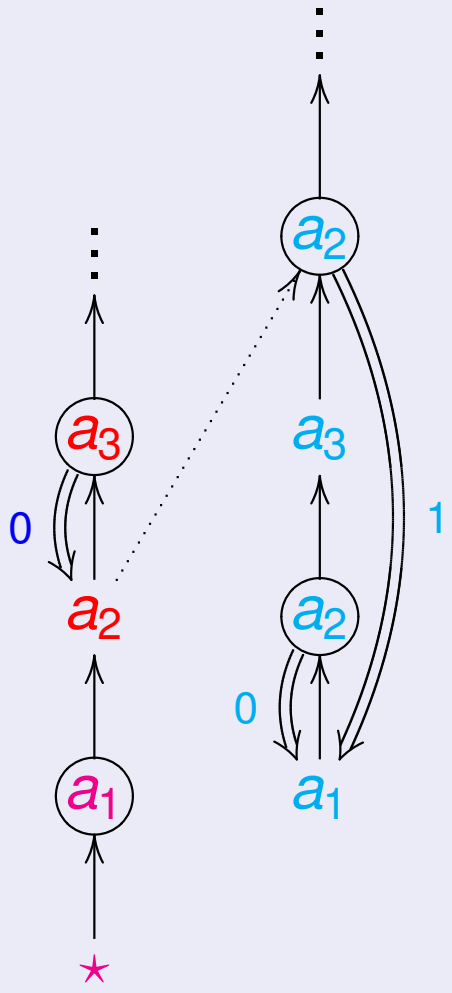
GAM at work: step 5



1 $\star$
$\overline{2}$ $\star$[a1,−]
2 a1
$\overline{3}$ a1[a2,0]
3 $\star$[a1,−]a2
$\overline{4}$ $\star$[a1,−]a2[a3,0]
4 a1[a2,0]a3
$\overline{5}$ a1[a2,0]a3[a2,1]
5 $\star$[a1,−]a2
$\overline{6}$ $\star$[a1,−]a2[a3,0]
6 a1[a2,0]a3[a2,1]$\cdots$

Abstract machines

GAM at work: step $\overline{6}$



$\overline{1}$ $\star$

$\overline{2}$ $\star$[a1,-]

2 a1

$\overline{3}$ a1[a2,0]

3 $\star$[a1,-]a2

$\overline{4}$ $\star$[a1,-]a2[a3,0]

4 a1[a2,0]a3

$\overline{5}$ a1[a2,0]a3[a2,1]

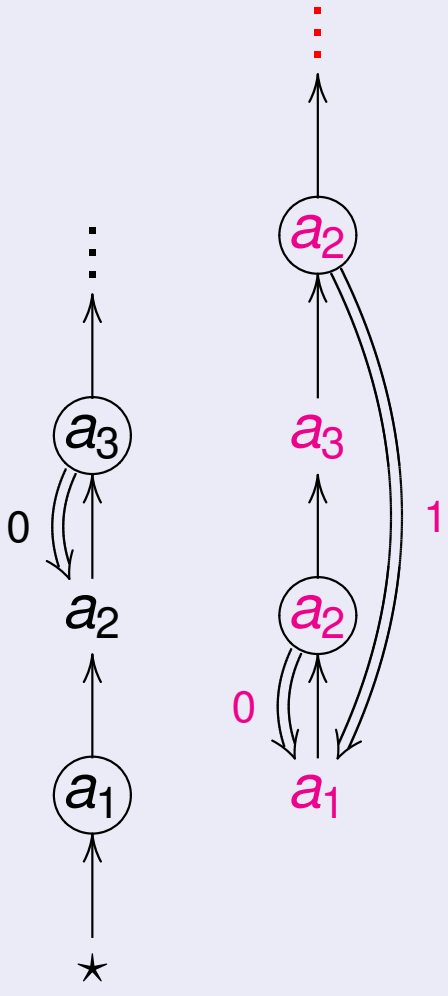
5 $\star$[a1,-]a2

$\overline{6}$ $\star$[a1,-]a2[a3,0]

6 a1[a2,0]a3[a2,1]$\cdots$

Abstract machines

GAM at work: step 6



$\overline{1}$ $\star$
$\overline{2}$ $\star$[a1,-]
2 a1
$\overline{3}$ a1[a2,0]
3 $\star$[a1,-]a2
$\overline{4}$ $\star$[a1,-]a2[a3,0]
4 a1[a2,0]a3
$\overline{5}$ a1[a2,0]a3[a2,1]
5 $\star$[a1,-]a2
$\overline{6}$ $\star$[a1,-]a2[a3,0]
6 a1[a2,0]a3[a2,1]$\cdots$

Argumentation

Summary of content

- Example dialogue about burden of proof
- Synthesised dialogue and formalisation

Argumentation

Summary of content

- Example dialogue about burden of proof
- Synthesised dialogue and formalisation

Prakken, Reed and Walton

- *Dialogues about the burden of proof* (in ICAIL, 2005)
- Combine persuasion dialogue with burden of proof
  - argumentation schemes, critical questions
  - technical solution based on dialogue levels

Argumentation

Summary of content

- Example dialogue about burden of proof
- Synthesised dialogue and formalisation

Prakken, Reed and Walton

- *Dialogues about the burden of proof* (in ICAIL, 2005)
- Combine persuasion dialogue with burden of proof
  - argumentation schemes, critical questions
  - technical solution based on dialogue levels

Use of pointer interaction

- Embedded dialogues and concept of backtracking
  - backtracking: returning to earlier point in dialogue
- Dialogue as product of normalisation via GAM

Argumentation

Example of legal dispute

u_1:CLAIM C

v_1:WHY C

u_2:C SINCE $says(e, C) \wedge expert(e, C)$

v_2:WHY $\neg biased(e)$

u_3:WHY $biased(e)$

v_3:BOP $(\neg biased(e), u)$ SINCE $\neg biased(e) \rightarrow trusted(e)$

u_4:WHY $\neg biased(e) \rightarrow trusted(e)$

v_4:WHY $\neg(\neg biased(e) \rightarrow trusted(e))$

u_5:$\neg(\neg biased(e) \rightarrow trusted(e))$ SINCE $presumed(\neg biased(e))$

v_5:RETRACT $\neg biased(e) \rightarrow trusted(e)$

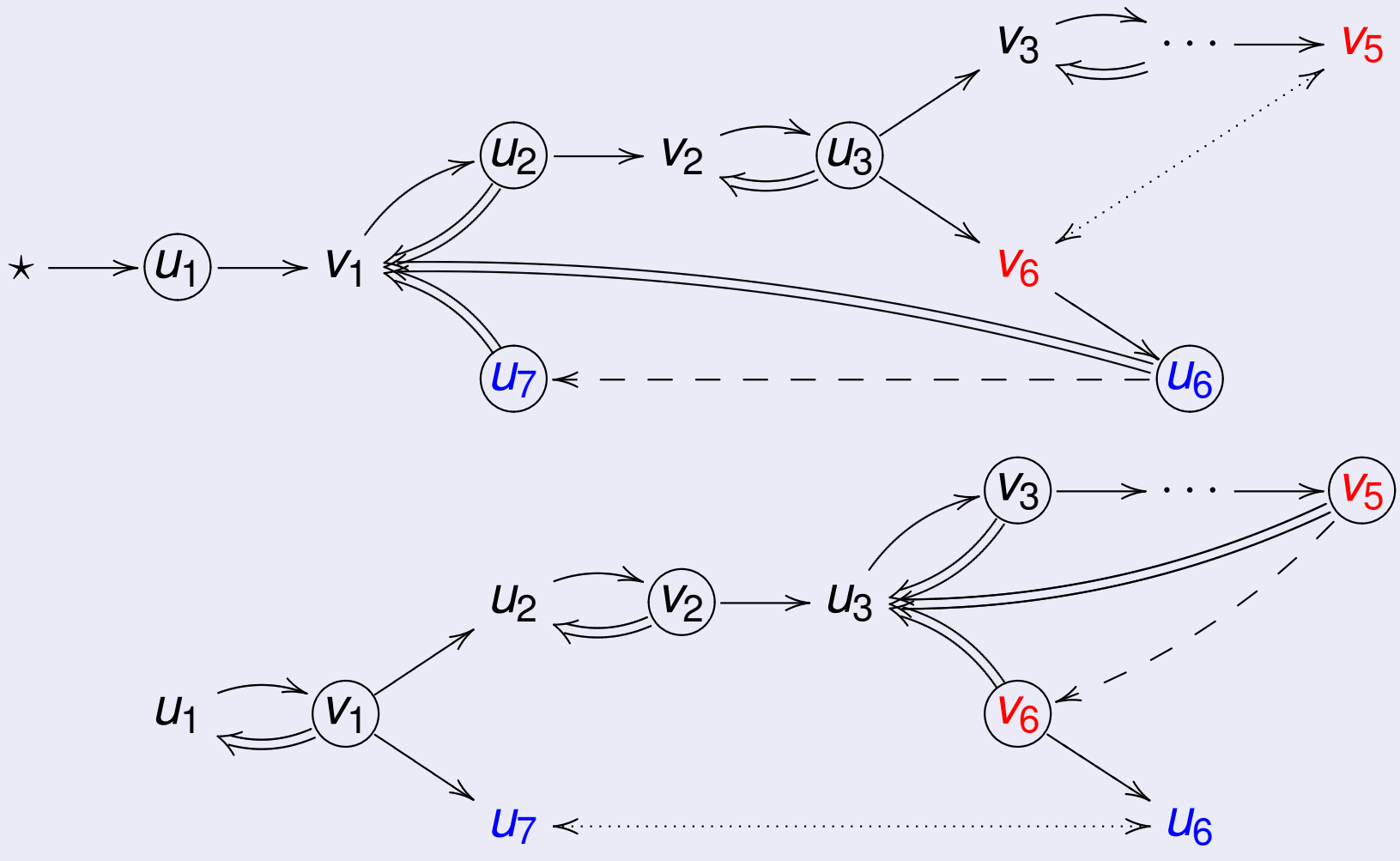
v_6:$biased(e)$ SINCE $paid(e, c) \wedge testifies(e, c)$

u_6:CONCEDE $biased(e)$

u_7:RETRACT C

Argumentation

u's & v's point of view



Conclusion

General considerations

- Dialogue regarded as product of interaction
- Pointer interaction crucial for backtracking
- Lots of other applications indeed possible

Conclusion

General considerations

- Dialogue regarded as product of interaction
- Pointer interaction crucial for backtracking
- Lots of other applications indeed possible

Ongoing and future work

- Syntax versus semantics
  - grammars as abstract Böhm trees
  - compositional theory of meaning?
- Analysis versus synthesis
  - modular approach to abstract Böhm trees
  - abstract Böhm trees as data structures?
- Rationality, decision making
  - implementation of selection functions?