Multimedia Information Extraction and Retrieval

A Probabilistic Abduction Engine for Media Interpretation based on Ontologies

Ralf Möller
Hamburg University of Technology

Application Context

- For multimedia interpretation and for the combined interpretation of information coming from different modalities a semantically well-founded formalization is required
- Images, Text, Video, Audio…
- Low-level percepts represent the observations (e.g., of an agent).
- Symbolic observations require interpretation
- Interpretations in turn are seen as explanations for the observations.

General Approach

- We propose an **abduction-based** formalism that uses description logics for the ontology and Horn rules for defining the space of hypotheses for explanations.

$$\Sigma \cup \Delta \models \Gamma$$

- Abduction example:

$\forall x,y$ *causes(x,y)* ← $\exists z$ *CarEnry(z), Car(x), DoorSlam(y), hasObject(z,x), hasEffect(z,y)*

$\forall x,y$ *causes(x,y)* ← $\exists z$ *CarExit(z), Car(x), DoorSlam(y), hasObject(z,x), hasEffect(z,y)*

- Multiple explanations possible with simple scores
  - Makes ranking difficult
- Why should an agent look for explanations?

Probabilistic Abduction

- Agent wants to minimize its uncertainty about observations
- Agent considers probability that observations are true given certain explanations
- Need to combine probability theory with first-order logic
- We use the Markov logic formalism to define the motivation for the agent to generate explanations and for ranking different explanations.

In Detail:

- **Idea of ranking:**

  Probability that the observations are true given the evidences.

  P(observation|explanation)

- **Idea of controlling the interpretation process :**

  Accept (additional) explanations only if the probability that observations are true (given the additional explanations) is significantly increased.

Markov Logic Networks

- A Markov Logic KB (ML-KB) is a set of pairs (F_i, w_i) where
  - F_i is a formula in first-order Logic
  - w_i is a real number weight
- Together with a finite set of constants it defines a Markov Logic Network (MLN) with
  - one node for each ground atom of predicates in ML-KB
  - one edge between two nodes ⇔ corresponding ground atoms appear together in grounding of some F_i

[Domingos et al. 2007]

Example

Weighted rules:

$$5\,\forall z\; CarEntry(z) \wedge hasObject(z,x) \wedge hasEffect(z,y) \rightarrow$$
$$Car(x) \wedge DoorSlam(y) \wedge causes(x,y)$$

$$5\,\forall z\; EnvConference(z) \wedge hasSubEvent(z,x) \wedge hasLocation(z.y) \rightarrow$$
$$CarEntry(x) \wedge Building(y) \wedge OccursAt(x,y)$$

Knowledge Representation in Markov Logic: Probability Distributions

- Log-linear model for specifying the probability distribution (probability of possible world *x*):

$$P(x) = \frac{1}{Z}\exp\left(\sum_{i=1}^{F} w_i n_i(x)\right)$$

Weight of F_i

Number of true groundings of F_i in x

- *Z* is the partition function given by:

$$Z = \sum_{x}^{X}\exp\left(\sum_{i=1}^{F} w_i n_i(x)\right)$$

Inference Problem 1: MLN Query Answering

- **Probability query**:

$$P_{MLN}(x_1 \wedge \ldots \wedge x_m \mid \vec{e})$$

Used for computing scores assigned
to the interpretation Aboxes (see below)

Inference Problem 2: Maximum A-Posteriori in MLN

- MAP approach determines the most probable world given the evidence.

- **Most-probable world query** (Maximum A-Posterior, MAP)

$$MAP_{MLN}(\vec{e}) := \vec{e} \cup argmax_{\vec{x}} \frac{1}{Z_e} \exp\left(\sum_i w_i n_i(\vec{x}, \vec{e})\right)$$

which can be slightly optimized s.th.

$$MAP_{MLN}(\vec{e}) := \vec{e} \cup argmax_{\vec{x}} \sum_i w_i n_i(\vec{x}, \vec{e})$$

Abduction Example

- For the explanation of *Causes(c1,ds1)* :

$$Causes(x, y) \leftarrow CarEntry(z), HasObject(z, x), HasEffect(z, y), Car(x), DoorSlam(y)$$

$$OccursAt(x, y) \leftarrow EnvConference(z), HasSubEvent(z, x), HasLocation(z, y), CarEntry(x), Building(y)$$

Abduction rules (new vars on the righthand side existentially quantified):

- Abduction requires consistent input

Prerequisites

Γ

1.3	$Car(C_1)$
1.2	$DoorSlam(DS_1)$
0.7	$EngineSound(DS_1)$
	Causes(C1,DS1)

MAP

Gound atoms	W
Car(C1)	1
DoorSlam(DS1)	1
EngineSound(DS1)	0
Causes(C1,DS1)	1

Combination of audio and video for this focus

DoorSlam $\sqsubseteq \neg$ EngineSound

Select

Γ'

1.3	$Car(C_1)$
1.2	$DoorSlam(DS_1)$
	$Causes(C_1, DS_1)$.

Concept-based Abduction Engine: Basic Idea

1. Forward chain rules on Abox A_i
2. Given a set of observations Γ , try to explain a selected assertion
3. Each explanation possibly introduces new assertions
4. Add new assertions to A_i
5. Continue with step 1. unless none of the explanations derived in this round cause the probability that the initial observations are true to increase "substantially"
6. Return the explanations (to be ranked afterwards)

Complete Example

Tbox:

$$CarEntry \sqsubseteq \neg DoorSlam$$
$$\ldots$$

Abduction rules (new vars on the righthand side existentially quantified):

$$Causes(x,y) \leftarrow CarEntry(z), HasObject(z,x), HasEffect(z,y), Car(x), DoorSlam(y)$$
$$OccursAt(x,y) \leftarrow EnvConference(z), HasSubEvent(z,x), HasLocation(z,y), CarEntry(x), Building(y)$$

Forward rules:

$$\forall x \;\; CarEntry(x) \rightarrow \exists y \;\; Building(y), OccursAt(x,y)$$

Weighted rules:

$$5\, \forall z\, CarEntry(z) \wedge hasObject(z,x) \wedge hasEffect(z,y) \rightarrow Car(x) \wedge DoorSlam(y) \wedge causes(x,y)$$

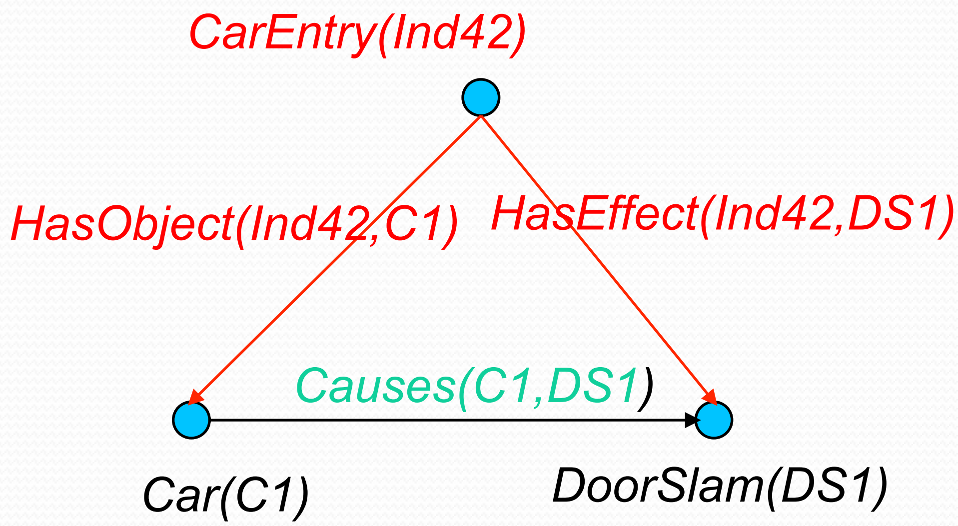
$$5\, \forall z\, EnvConference(z) \wedge hasSubEvent(z,x) \wedge hasLocation(z.y) \rightarrow CarEntry(x) \wedge Building(y) \wedge OccursAt(x,y)$$

Formulas are extremely simplified to make them fit on a slide.

Example (Backward rules)

$$Causes(x,y) \leftarrow CarEntry(z), HasObject(z,x), HasEffect(z,y), Car(x), DoorSlam(y)$$

$$OccursAt(x,y) \leftarrow EnvConference(z), HasSubEvent(z,x), HasLocation(z,y), CarEntry(x), Building(y)$$



Hypothesized assertions in red

Example (Backward rules)

Γ'

1.3	$Car(C_1)$
1.2	$DoorSlam(DS_1)$
	$Causes(C_1, DS_1)$.

$$\Delta_1 = \{CarEntry(Ind_{42}), HasObject(Ind_{42}, C_1), HasEffect(Ind_{42}, DS_1)\}$$

Abox A_1

1.3	$Car(C_1)$
1.2	$DoorSlam(DS_1)$
	$Causes(C_1, DS_1)$.
	$CarEntry(Ind_{42})$.
	$HasObject(Ind_{42}, C_1)$.
	$HasEffect(Ind_{42}, DS_1)$.

$$P(Car(C_1) \wedge DoorSlam(DS_1) \wedge Causes(C_1, DS_1) \mid \Delta_1) = 0.840$$

Example (Forward rules)

$$\forall x \;\; CarEntry(x) \rightarrow \exists y \;\; Building(y), OccursAt(x, y)$$

$$\Delta_f = \{Building(Ind_{43}), OccursAt(Ind_{42}, Ind_{43})\}$$

1.3 $Car(C_1)$
1.2 $DoorSlam(DS_1)$
$Causes(C_1, DS_1).$

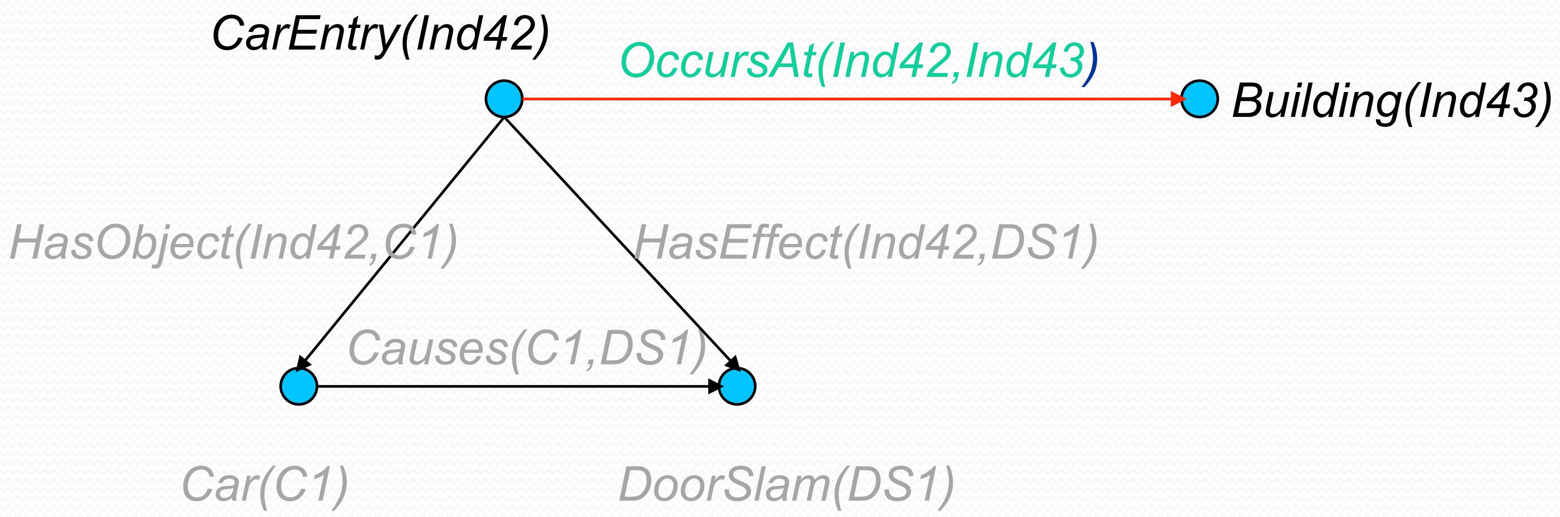
$CarEntry(Ind_{42}).$
$HasObject(Ind_{42}, C_1).$
$HasEffect(Ind_{42}, DS_1).$

$Building(Ind_{43}).$
$OccursAt(Ind_{42}, Ind_{43}).$

Example (Backward rules)

$$Causes(x,y) \leftarrow CarEntry(z), HasObject(z,x), HasEffect(z,y), Car(x), DoorSlam(y)$$
$$OccursAt(x,y) \leftarrow EnvConference(z), HasSubEvent(z,x), HasLocation(z,y), CarEntry(x), Building(y)$$

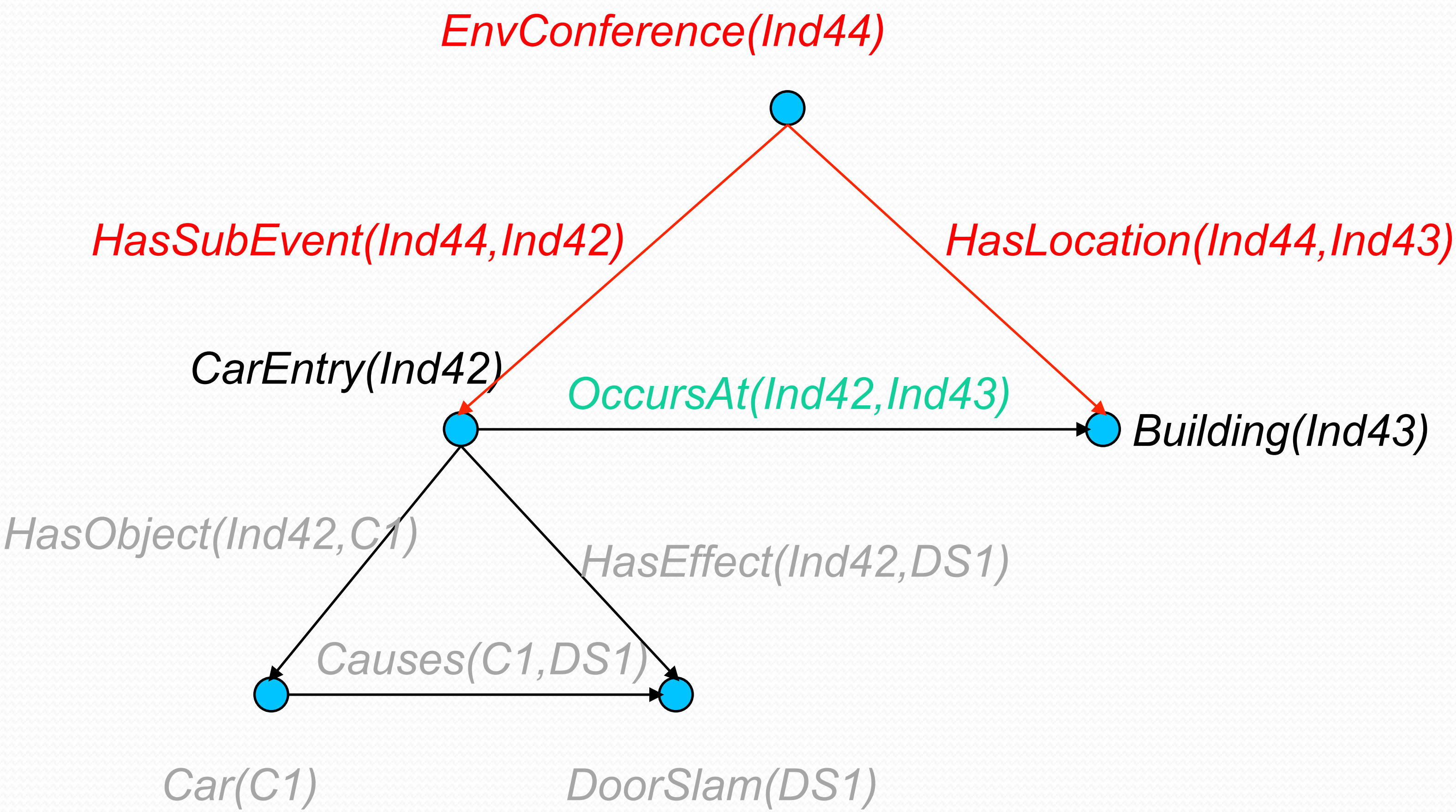
Abduction rules (new vars on the righthand side existentially quantified):



Example (Backward rules)

$$Causes(x,y) \leftarrow CarEntry(z), HasObject(z,x), HasEffect(z,y), Car(x), DoorSlam(y)$$

$$OccursAt(x,y) \leftarrow EnvConference(z), HasSubEvent(z,x), HasLocation(z,y), CarEntry(x), Building(y)$$



Example (Backward rules)

1.3	$Car(C_1)$
1.2	$DoorSlam(DS_1)$
	$Causes(C_1, DS_1)$.
	$CarEntry(Ind_{42})$.
	$HasObject(Ind_{42}, C_1)$.
	$HasEffect(Ind_{42}, DS_1)$.
	$Building(Ind_{43})$.
	$OccursAt(Ind_{42}, Ind_{43})$.

$$\Delta_2 = \{EnvConference(Ind_{44}), hasSubEvent(Ind_{44}, Ind_{42}), hasLocation(Ind_{44}, Ind_{43}), \ldots\}$$

Example (ranking step)

Abox $\mathcal{A}_2$

1.3	$Car(C_1)$
1.2	$DoorSlam(DS_1)$
	$Causes(C_1, DS_1)$.
	$CarEntry(Ind_{42})$.
	$HasObject(Ind_{42}, C_1)$.
	$HasEffect(Ind_{42}, DS_1)$.
	$Building(Ind_{43})$.
	$OccursAt(Ind_{42}, Ind_{43})$.
	$EnvConference(Ind_{44})$.
	$HasSubEvent(Ind_{44}, Ind_{42})$.
	$HasLocation(Ind_{44}, Ind_{43})$.
	…

Δ_1

Δ_f

Δ_2

$$P(Car(C_1) \wedge DoorSlam(DS_1) \wedge Causes(C_1, DS_1) \mid \Delta_1 \cup \Delta_f \cup \Delta_2) = 0.819$$

Example : Results

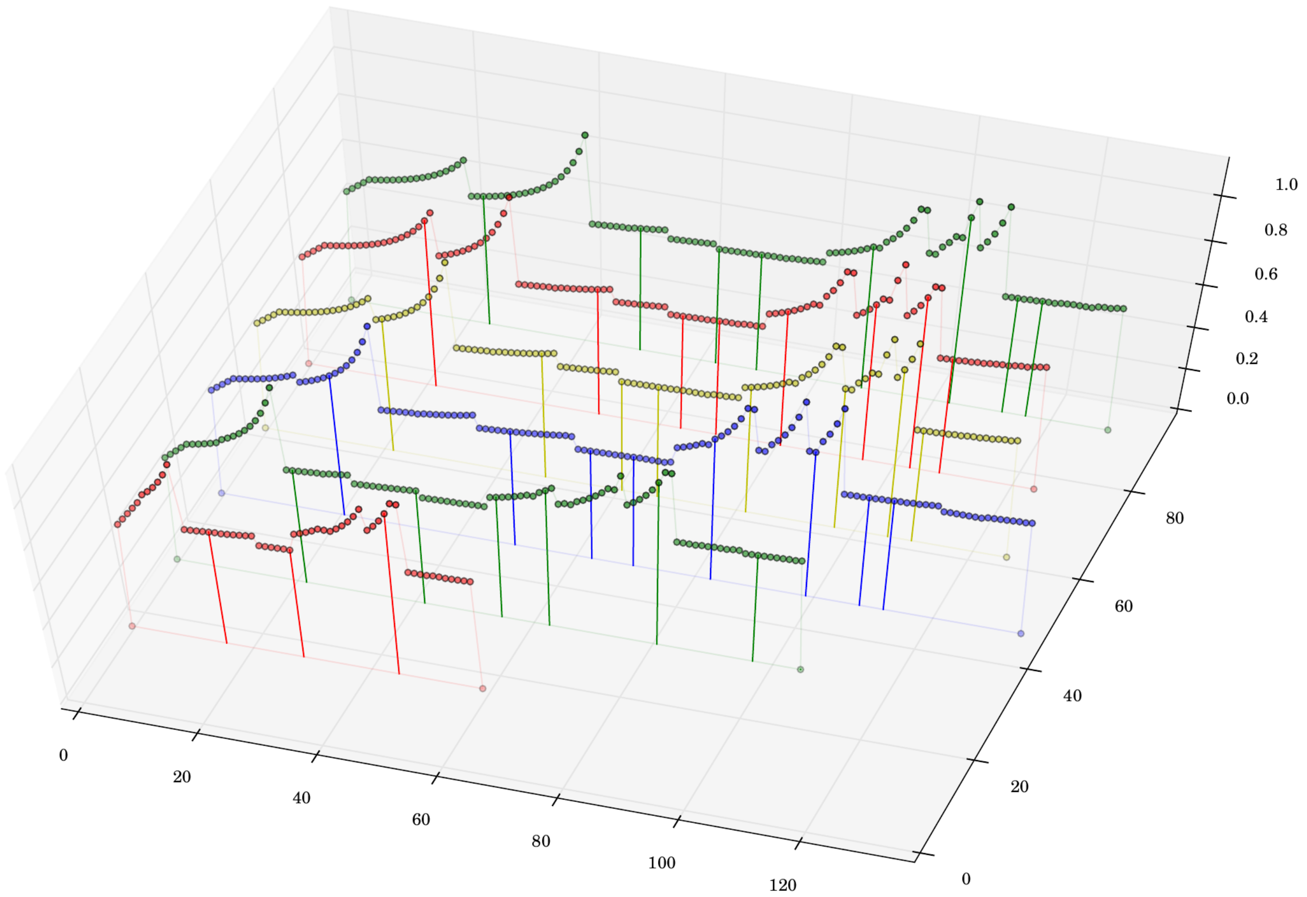
Prob.	Values
p_0	0.650
p_1	0.840
p_2	0.819

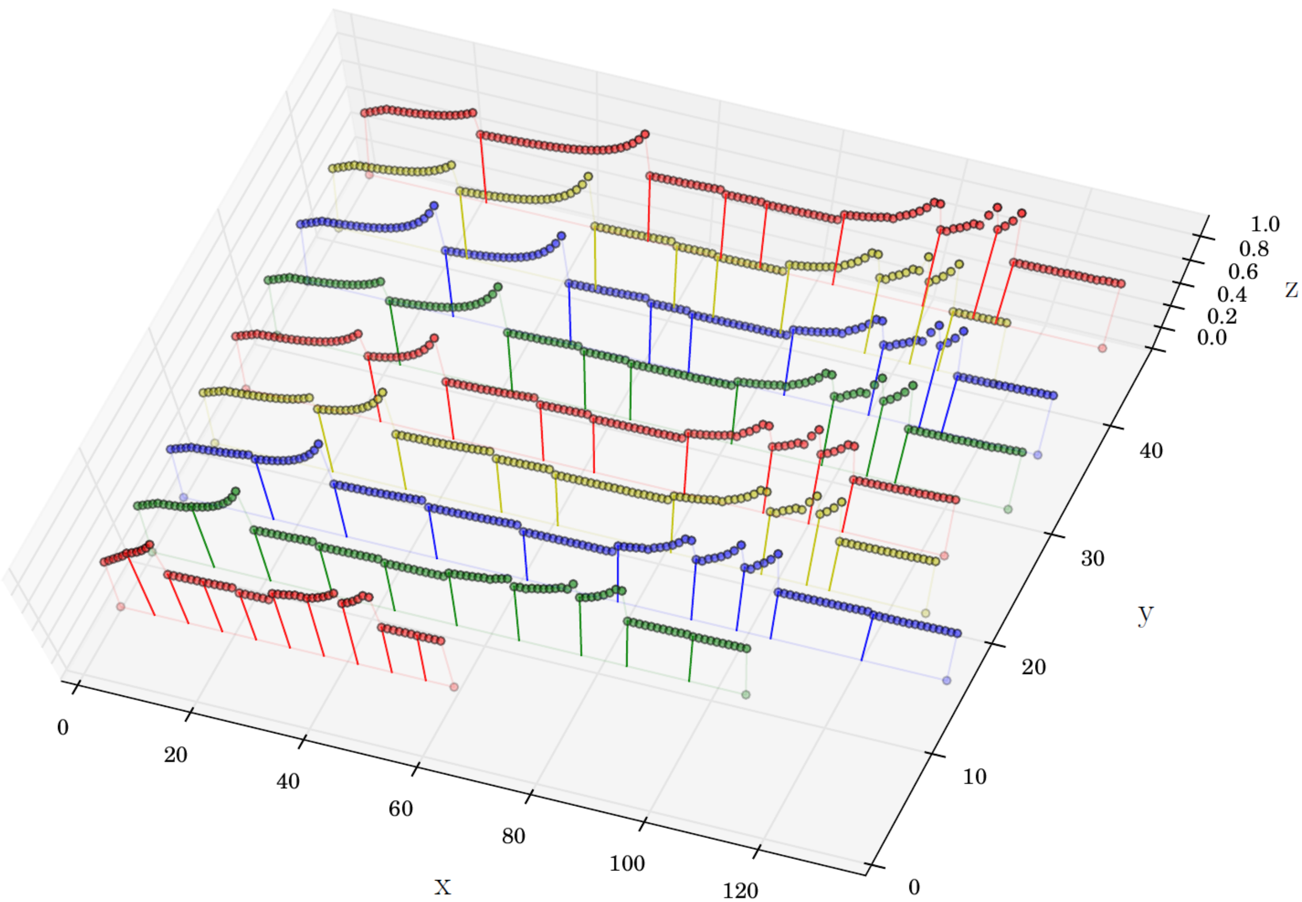
The termination condition is fulfilled.

→ **Abox A_1 is considered as the final interpretation Abox.**

Scoring

- For every interpretation (explained, non-explained)
  - For every explained add P(Obs | Interpretation)
  - For every non-explained add 0.5
  - Average w.r.t. number of assertions in interpretation


1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
120
0
20
40
60
80



1.0
0.8
0.6
0.4
0.2
0.0
z
0
10
20
30
40
y
0
20
40
60
80
100
120
x

Acknowledgements

- Funded by
  - European Commission (contract FP7-217061 CASAM)
  - Deutsche Forschungsgemeinschaft (contract MO-801-1 PRESINT)
- Thanks to the TUHH CASAM project members
  - Oliver Gries, Maurice Rosenfeld, Anahita Nafissi, Kamil Sokolski
- Thanks to the PRESINT project members
  - Prof. Bernd Neumann, UniHH, Dr. Michael Wessel, Reza Rasouli, Sebastian Wandelt