



European Union  
European Social Fund



MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS  
MANAGING AUTHORITY

Co-financed by Greece and the European Union



EUROPEAN SOCIAL FUND

# Towards formal representation and comparison of video content using algebraic semiotics

9th International Workshop on Semantic and  
Social Media Adaptation and Personalization

November 6-7<sup>th</sup>, 2014, Corfu, Greece

**Petros Stefaneas**

# Introduction

---

- ▶ **Main idea: use (an extension of) algebraic semiotics to:**
  - ▶ formally specify systems of moving pictures
  - ▶ compare their semantic annotated representations
- ▶ **Some first steps are presented here through an Illustrative example**



# Introduction

---

- ▶ Video data are more complex and larger than traditional text data
- ▶ Combine visual, audio and textual data
  - ▶ need to be appropriately annotated
  - ▶ indexed in an accessible form for search and retrieval
- ▶ This can be achieved based on *textual information, visual and audio features and semantic information*



# Introduction

---

- ▶ Text-based approach: the simplest
- ▶ Keywords or short sentences describe the content
  - ▶ big effort and time to accomplish the annotation task
  - ▶ possible accuracy issues
- ▶ Feature-based approach (visual or audio)
- ▶ Annotation by combinations of extracted features like intensity, color, texture, shape, motion, etc.
  - ▶ useful in doing a query-by example task
  - ▶ not suitable for searching for specific event



# Introduction

---

- ▶ Semantic-based approach: most promising
- ▶ Annotation with high-level information that represents the semantic meaning of the content
  - ▶ main difficulty: high variability of the semantic meaning
  - ▶ the meaning depends on many factors, including the purpose of the annotation, the domain, application and the annotator itself!
    - ▶ problem when the annotator is a human



# Introduction

---

- ▶ **Open research question: the quality of the annotation**
  - ▶ is it suitable for the specific application?
  - ▶ has it captured the most important information of the original video?
- ▶ **Also it may be difficult to choose *which* among different representations is the most appropriate – *compare annotations***



# Introduction

---

- ▶ We present *the first steps* of a methodology, based on algebraic semiotics, that can be used to
  - ▶ formally model video content
  - ▶ provide a means for measuring the quality of video annotated representations



# Algebraic semiotics

---

- ▶ Approach to reasoning about representations, which builds on the following insights;
  - ▶ Signs do not come in isolation but as elements of *systems of related signs*, including their structural aspects
  - ▶ *Representations* can be seen as *translations or maps* from one sign system to another
  - ▶ We can think of *a sign system as a set of signs*, grouped into *sorts and levels*, with *constructor functions* at each level that build new signs from old ones



# Algebraic semiotics - Definition

---

**A sign system  $S$  consists of:**

- 1) a set  $S$  of sorts for signs;
- 2) a partial ordering on  $S$ , called the subsort relation
- 3) a set  $V$  of data sorts, for information about signs, such as colors, locations, and truth values;
- 4) a partial ordering of sorts by level, such that data sorts are lower than sign sorts
- 5) a set  $C_n$  of level  $n$  constructors, used to build level  $n$  signs from other signs at levels  $n$  or less
- 6) a priority (partial) ordering on each  $C_n$
- 7) some relations and functions on signs
- 8) a set  $A$  of sentences (in the sense of logic), called axioms that constrain possible signs



# Algebraic semiotics

---

- ▶ The translation between sign systems is captured by *semiotics morphisms*;
  - ▶ they provide a *representation* from the source sign system into the target
  - ▶ they can be *partial*, i.e., they do not necessarily have to preserve all of the signs or all the structure of the source system
- ▶ *So the key point is that the quality of a representation can be examined in terms of what is preserved by its semiotic morphism!*



# Algebraic semiotics

---

A good semiotic morphism (*hence a good representation/annotation as well*) should preserve as much of the structure in its source sign system as possible.

- ▶ It should map sorts to sorts, subsorts to subsorts, data sorts to data sorts, constants to constants, constructors to constructors, etc.

**But some information is not preserved**

- ▶ Axioms should also be preserved!



# Algebraic semiotics

---

For computer scientists, it is natural to formalize:

- ▶ a sign system as a loose algebraic theory (consisting of a signature and some axioms) with further structure and
- ▶ the translations between the sign systems using theory morphisms
- ▶ with the help of algebraic specification languages
- ▶ here we use CafeOBJ



# Algebraic semiotics - extension

---

- ▶ Algebraic semiotics methodology has been mainly applied to the domain of user interface design
- ▶ Classical semiotics is concerned with static signs and does not allow for signs that change in response to user input
- ▶ Extend them - use instead of many sorted order algebra, hidden algebra - so that the dynamic parts of user interfaces can be specified



# Algebraic semiotics - extension

---

- ▶ The main characteristic of hidden algebra is that it introduces the notion of internal states
- ▶ Two kinds of sorts:
  - ▶ visible that represent the data part of a specification
  - ▶ hidden sorts that denote the state of an abstract machine (dynamic part)



# Proposed methodology

---

The extended methodology of algebraic semiotics, seems promising in the domain of video semantic annotated content

- ▶ suitable for addressing the problem of reasoning about the quality of annotations/representations
- ▶ Also, using algebraic theories we can specify an abstract video independently of a particular picture formalism (deformation monoids, wang tiles etc.) and
- ▶ reason in general about dynamic video systems and study/verify their properties of interest



# Picture theory - deformation monoids

---

- ▶ Some basic points of the picture theory used in the example are:
  - ▶ A picture of rank  $(\alpha, \beta)$  is a rectangular array of dimensions  $\alpha, \beta$  constructed by elementary rectangular pieces called pixels
- ▶ On the set of all pictures *two operations* are defined: the horizontal and the vertical concatenations
  - ▶ The first one is carried out over pictures with the same width and the other over pictures of the same length.



# Picture theory - deformation monoids

---

- ▶ **One more operation: picture deformation**
  - ▶ Associating to every  $(r, s)$  and every pixel  $x$  a new pixel  $x(r, s)$  which results from  $x$  by multiplying its dimensions by  $r, s$
- ▶ **Pictures can be composed in two ways:**
  - ▶ The horizontal concatenation of a picture  $p$  with a picture  $q$  is the picture  $pq$  obtained by writing  $q$  on the right of  $p$ .
  - ▶ The vertical concatenation of a picture  $p$  with a picture  $q$  is the picture obtained by writing  $q$  on the bottom of  $p$



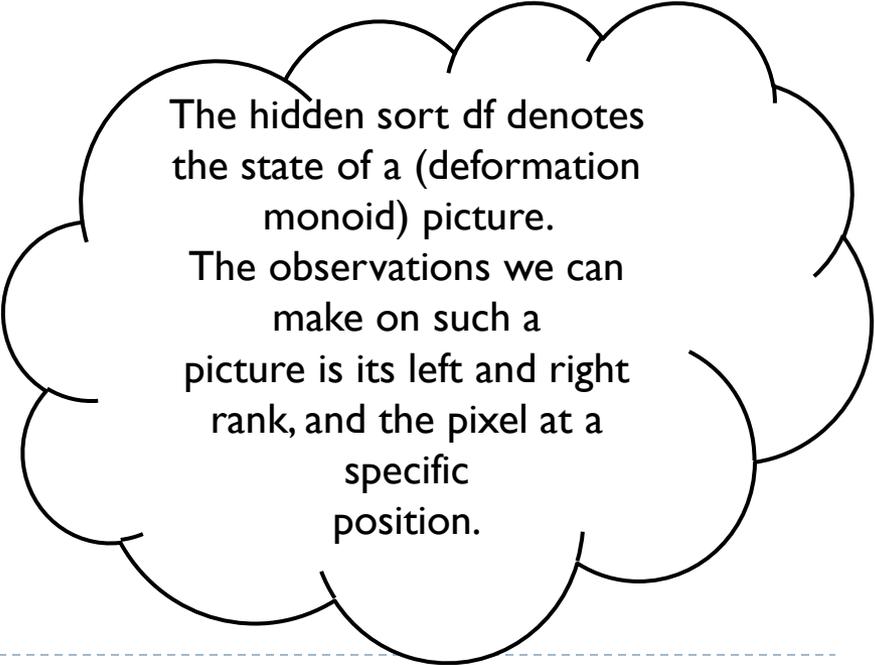
# Illustrative example

---

## Representing Pictures as algebraic theory

- ▶ We have specified an abstract picture (that uses deformation monoids as the underlying formalism) in **CafeOBJ** terms

```
mod* DF (X::TRIV) {  
  pr (NAT)  
  * [elt < dm] *  
  -- observation operators  
  bop lRank : dm -> Nat  
  bop rRank : dm -> Nat  
  bop pxAt : dm -> elt
```



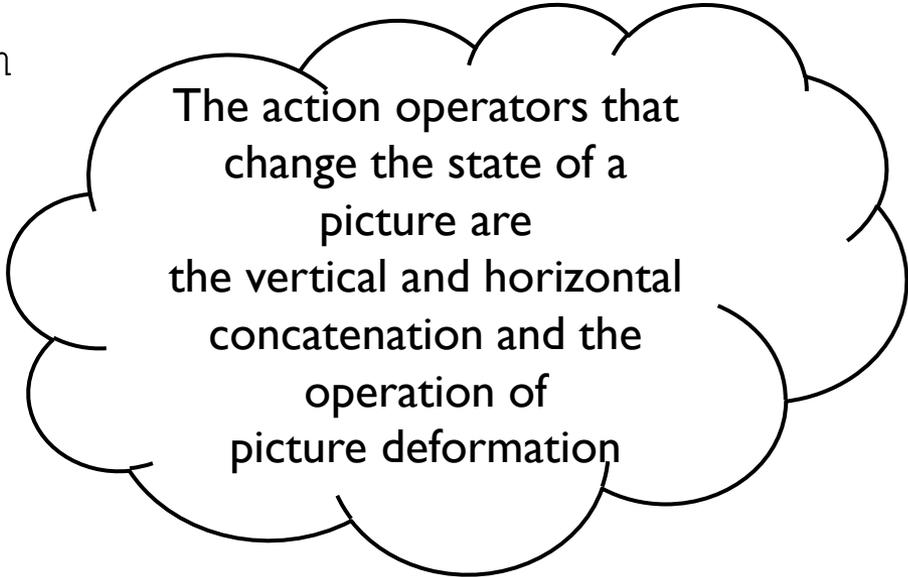
The hidden sort `df` denotes the state of a (deformation monoid) picture.

The observations we can make on such a picture is its left and right rank, and the pixel at a specific position.

# Illustrative example

---

```
-- action operators (constructors)
-- vertical concatenation
bop _+_ : dm dm -> dm {coherent, assoc}
-- horizontal concatenation
bop _|_ : dm dm -> dm {coherent, assoc}
-- picture deformation
bop def : dm Nat Nat -> dm
```



The action operators that change the state of a picture are the vertical and horizontal concatenation and the operation of picture deformation



# Illustrative example

---

- ▶ These (selected) axioms specify the behavior of our system, based on the previous picture definitions

```
-- axioms
```

```
ceq e[N1] + M = M if lRank(M) = N1 .
```

```
ceq rRank(M+N) = rRank(M) + rRank(N) if lRank(M) =  
lRank(N) .
```

```
ceq pxAt(M+N, N1, N2) = pxAt(N, N1, N2-rRank(M)) if N2 >  
rRank(M) and lRank(M) = lRank(N) and N1 <= lRank(M) .
```

```
ceq lRank(M|N) = lRank(M) + lRank(N) if rRank(M) =  
rRank(N) .
```

```
eq pxAt(def(M, N1, N2), N3, N4) = pxAt(M, ceil(N3/N1),  
ceil(N4/N2)) . }
```



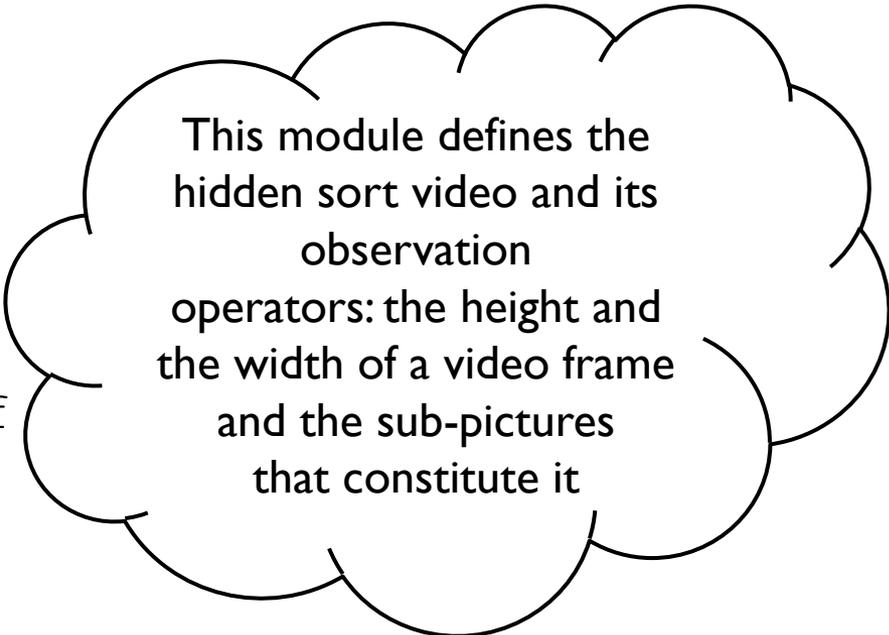
# Illustrative example

---

## Representing Video as algebraic theory

- ▶ We import our theory and extend it in order to define a video (i.e. a moving picture) theory

```
mod* VIDEO
pr (DM)
*[video]*
-- observation operators
bop height : video -> Nat
bop width  : video -> Nat
bop getSubPic : video -> df
```



This module defines the hidden sort video and its observation operators: the height and the width of a video frame and the sub-pictures that constitute it



# Illustrative example

---

-- single action operator (constructor)

bop changePix : video df Nat Nat -> video

- ▶ This transition can be used to define an algorithm that goes from one sub picture to another, intuitively like an encoding.

-- axioms

eq getSubPic (changePix (V, DF, N1, N2), N3, N4) = DF if N1 = N3 and N2 = N4 and

lRank (getSubPic (V, N3, N4)) = lRank (DF) and  
rRank (getSubPic (V, N3, N4)) = rRank (V, N3, N4) .

eq height (changePix (V, N1, N2)) = height (V) .

eq width (changePix (V, N1, N2)) = width (V) .

- ▶ The above axioms define how to change a "sub"- picture of a video frame with another one which has the same size. They also state that the size of the video frame does not change in that case.

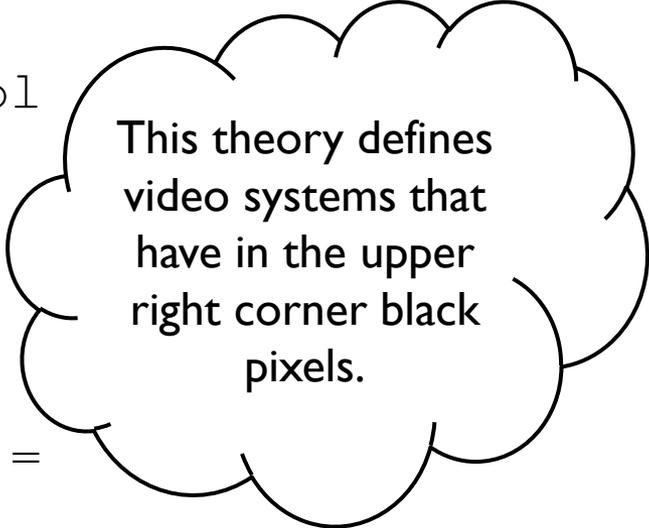


# Illustrative example

---

Assume now that as *source system* we have the theory (mod\* VIDEO) and as *target system* we have the following:

```
mod* TARGETVIDEO {  
  * [Tvideo] *  
  op height : Tvideo -> Nat  
  op width  : Tvideo -> Nat  
  op isblack : Tvideo Nat Nat -> Bool  
  -- variables  
  var T : Tvideo  
  vars N1 N2 : Nat  
  -- axiom  
  ceq isblack(T, N1, N2) = true if N1 =  
  height(T) and N2 = width(T) . }
```



This theory defines video systems that have in the upper right corner black pixels.



# Illustrative example

---

- ▶ Using the tools provided by algebraic semiotics it is easy to define a morphism between these two theories as follows:

```
view Source2Target from VIDEO to
```

```
TARGETVIDEO {
```

```
sort video -> Svideo
```

```
op height -> height
```

```
op width -> width }
```

- ▶ We can see that this annotation theory maps preserves some sorts and operators but is not a "good" representation for the source theory since it does not preserve its axioms!
    - ▶ no axiom is given for all relation instances
- 



## Future work

---

- ▶ Define an algebra/calculus for video, with appropriate operators
- ▶ Present how CafeOBJ can be used to mechanize up to a level the comparison of the representations
- ▶ Apply our methodology in real video systems and their semantic annotations



# Thank you!

---

▶ Questions?

