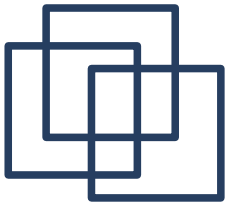


Overview of PRAESINT / CASAM Media Interpretation Agent

- A stream-based media interpretation agent
  - multiple modalities (speech, video, text, audio, video OCR, ...)
  - extended BOEMIE architecture, agent's interpretation loop
    - Assertions / „observations“ arrive via SOAP
    - get accumulated in an Abox (Abox gets bigger and bigger)

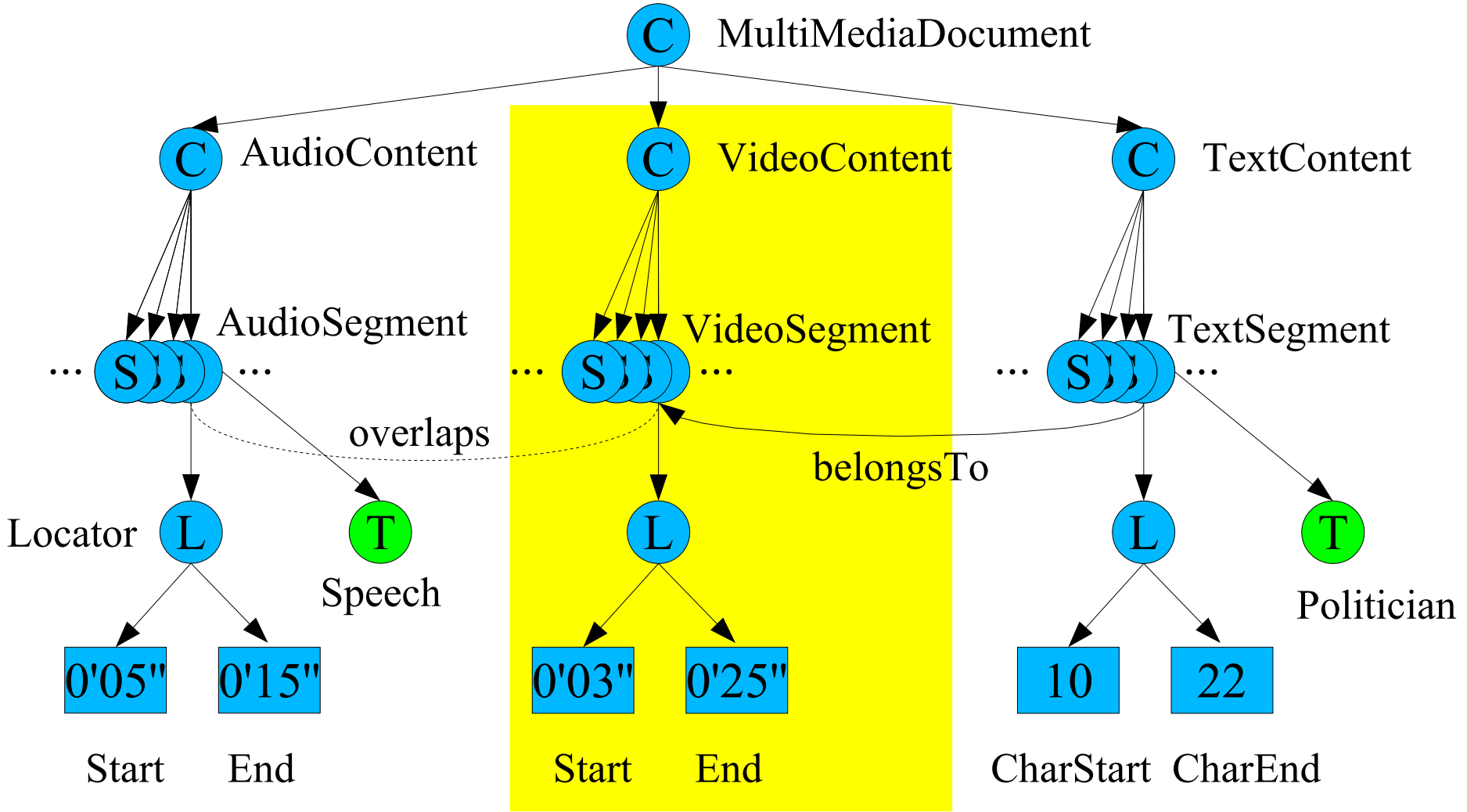
      → scalability?
    - Determine what to explain: FIAT generation rules (forward rules)

      → strategy?
    - Explain the FIAT assertion: abduction, extend best interpretation

      → very expensive on big Aboxes, optimization?
    - Collect explanations, probabilistic ranking of interpretations
    - Inform clients about changes in / of the best interpretation
    - Inform clients about alternative interpretations: queries!

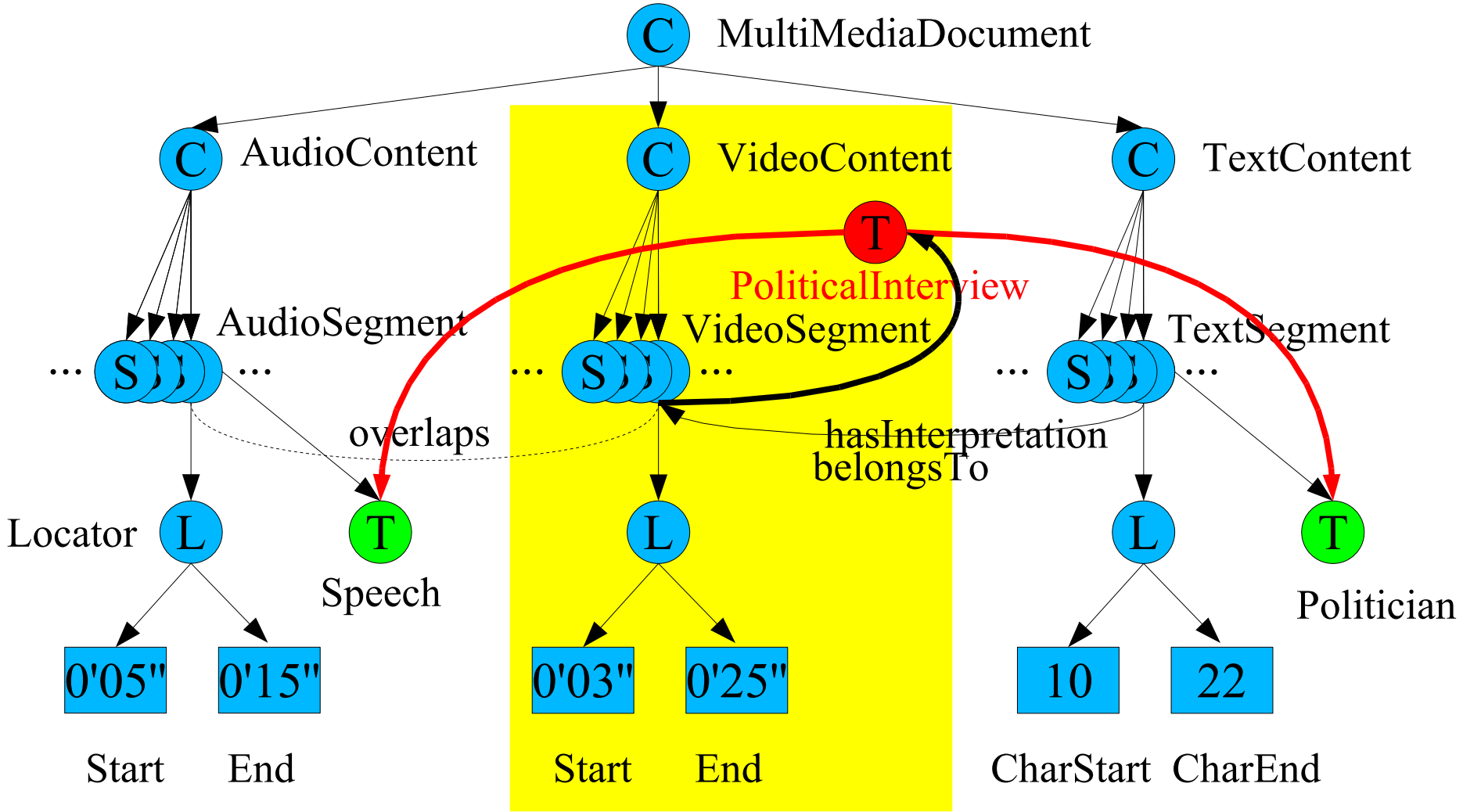
ABox Representation of MultiMedia Document

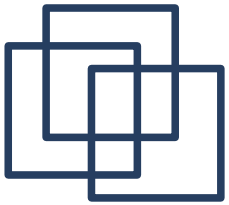
Example: „Text to Audio in local Video“ (Politician to Speech)



ABox Representation of MultiMedia Document

Example: „Text to Audio in local Video“ (Politician to Speech)



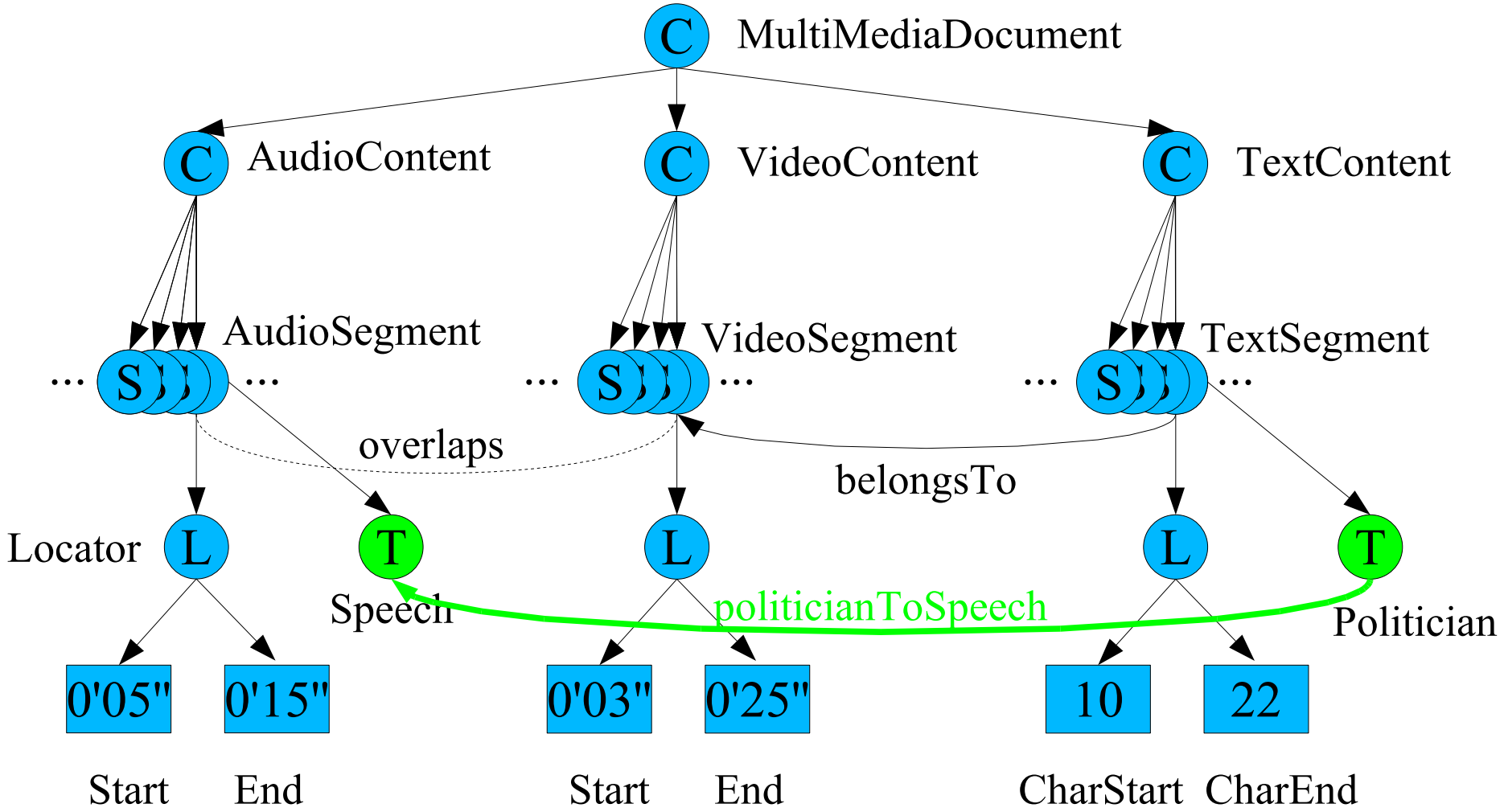


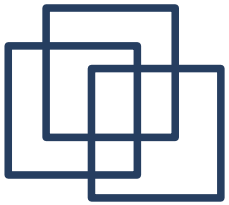
Identification of Coocurrences: Text X Audio

```
(defquery text-to-audio-in-local-video (?x ?y)
        (and
           (?mmd #!mco:MultimediaDocument)
           (?mmd ?vc #!mco:hasLogicalDecomposition)
           (?vc #!mco:VideoContent)
           (?vc ?vs #!mco:hasMediaDecomposition)
           (?vs #!mco:VideoSegment)
           (?vs ?vl #!mco:hasSegmentLocator)
           (not (?vs #!mco:GlobalVideoSegment))
           (?mmd ?tc #!mco:hasLogicalDecomposition)
           (?tc #!mco:TextContent)
           (?tc ?vs #!mco:belongsTo)
           (?tc ?ts #!mco:hasMediaDecomposition)
           (?mmd ?ac #!mco:hasLogicalDecomposition)
           (?ac #!mco:AudioContent)
           (?ac ?as #!mco:hasMediaDecomposition)
           (?as #!mco:AudioSegment)
           (?as ?al #!mco:hasSegmentLocator)
           (?al ?sm #!mco:overlaps)
           (?sm #!mco:SegmentLocator)
           (?sm ?vl #!mco:overlaps)
           (lambda (audio-near-video-p ?al ?vl "00:00:05,000"))
           (?ts ?x #!mco:depicts)
           (?as ?y #!mco:depicts)
           (not (?x ?y same-as))))
```

Fiat Generation

Coocurence of „Politician“ and „Speech“ → FIAT rule fires



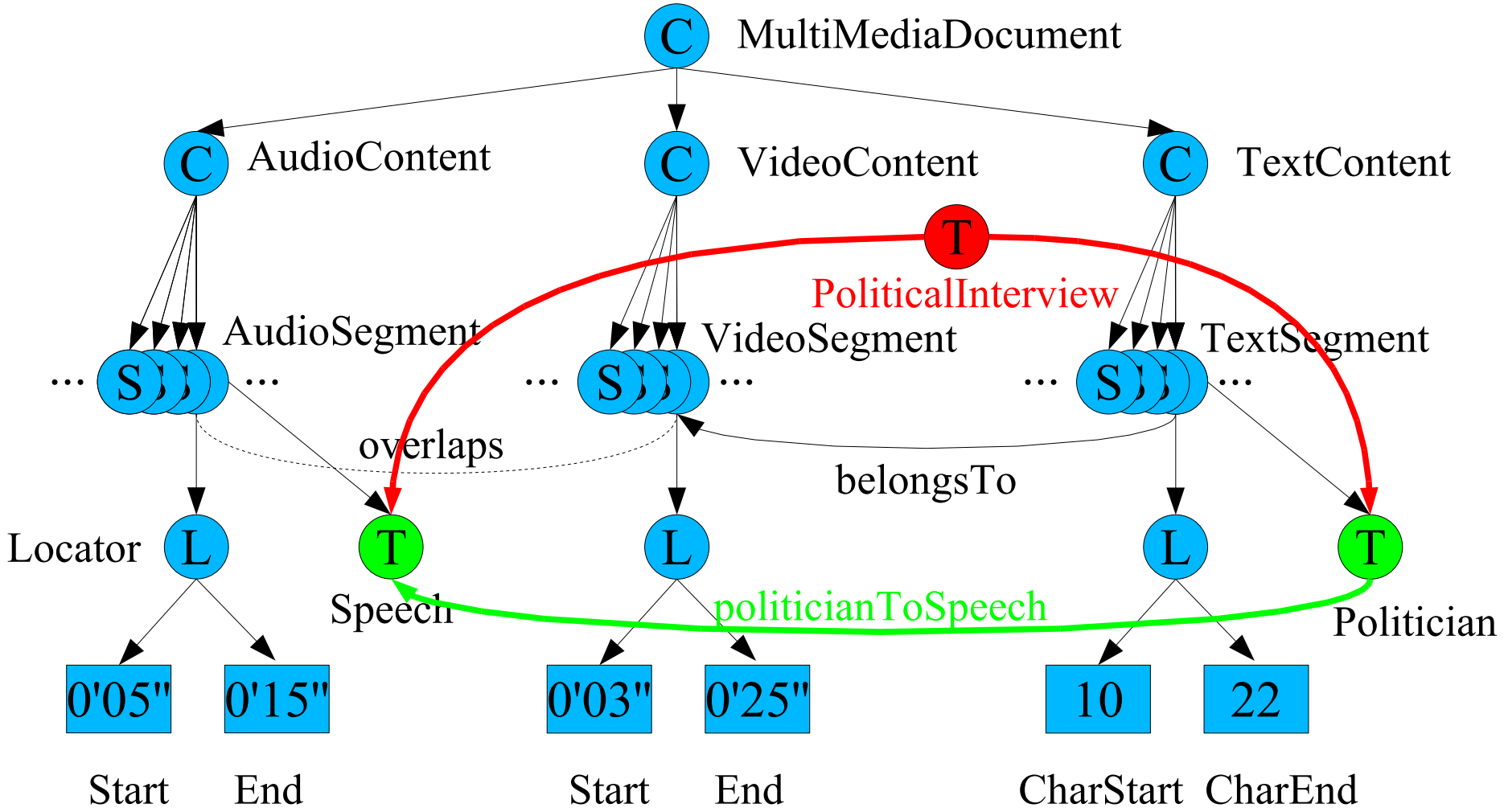


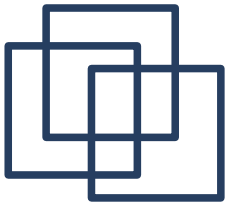
FIAT Generation Rule – Coocurence of Speech and Politician requires Explanation

```
(define-rule (?x ?y #!edo:politicianToSpeech)
             (and
              (?x #!edo:Politician)
              (?y #!edo:Speech)
              (?x ?y text-to-audio-in-local-video)
              (?x nil #!edo:politicianToSpeech)
              (nil ?y #!edo:politicianToSpeech))
             :backward-rule-p nil)
```

Explanation

Explanation of FIAT politicianToSpeech: PoliticalInterview



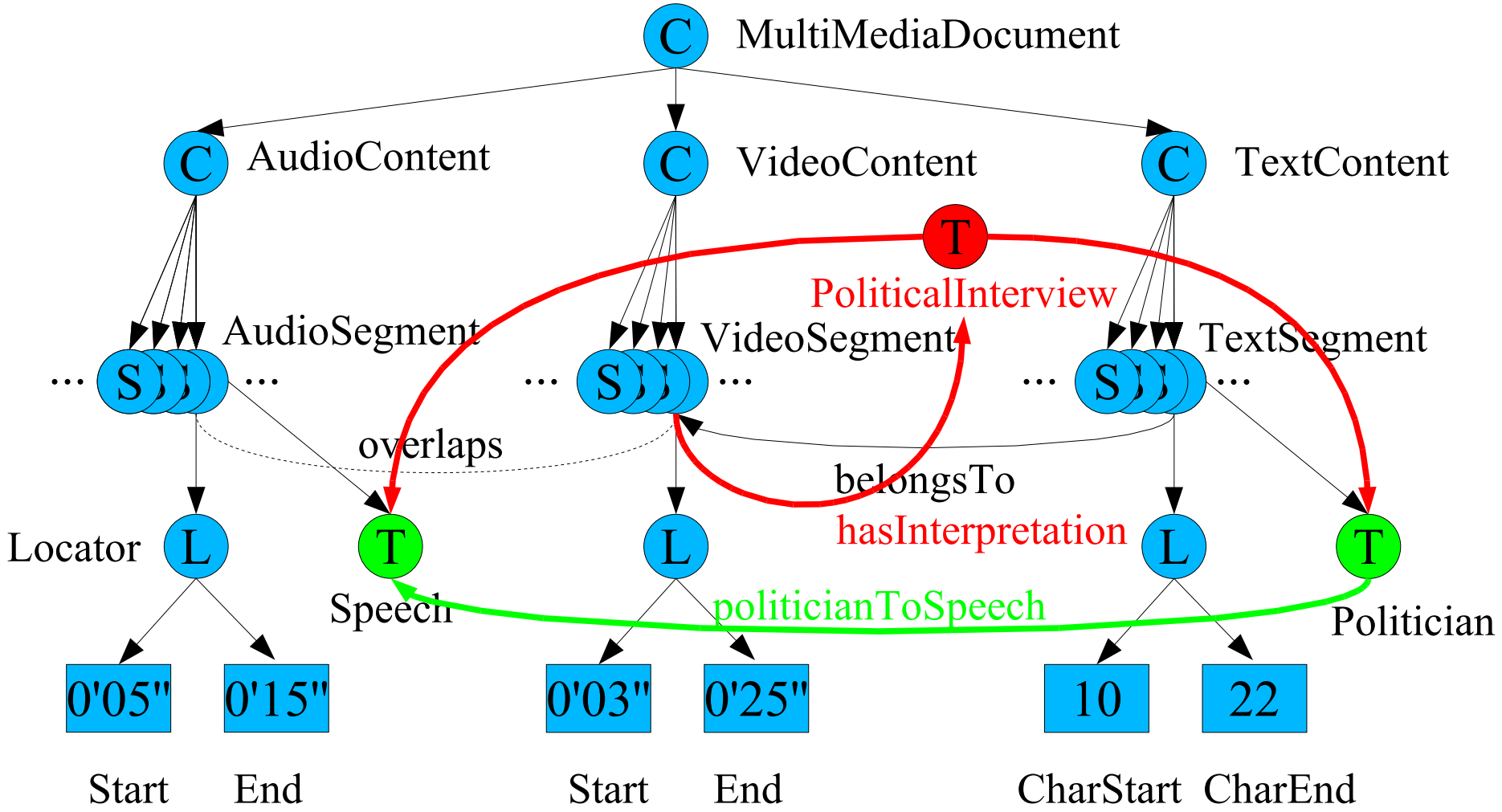


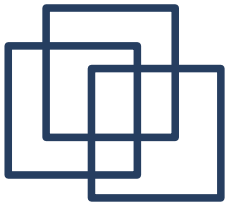
Abduction Rule for Fiat „politicianToSpeech“

```
(define-rule (?x ?y #!edo:politicianToSpeech)
             (and
                (?x #!edo:Politician)
                (?y #!edo:Speech)
                (?z #!edo:PoliticalInterview)
                (?z ?x #!mco:builtFrom)
                (?z ?y #!mco:builtFrom))
               :forward-rule-p nil)
```

Link Interpretation Result to VideoSegment

Explanation of FIAT politicianToSpeech: PoliticalInterview





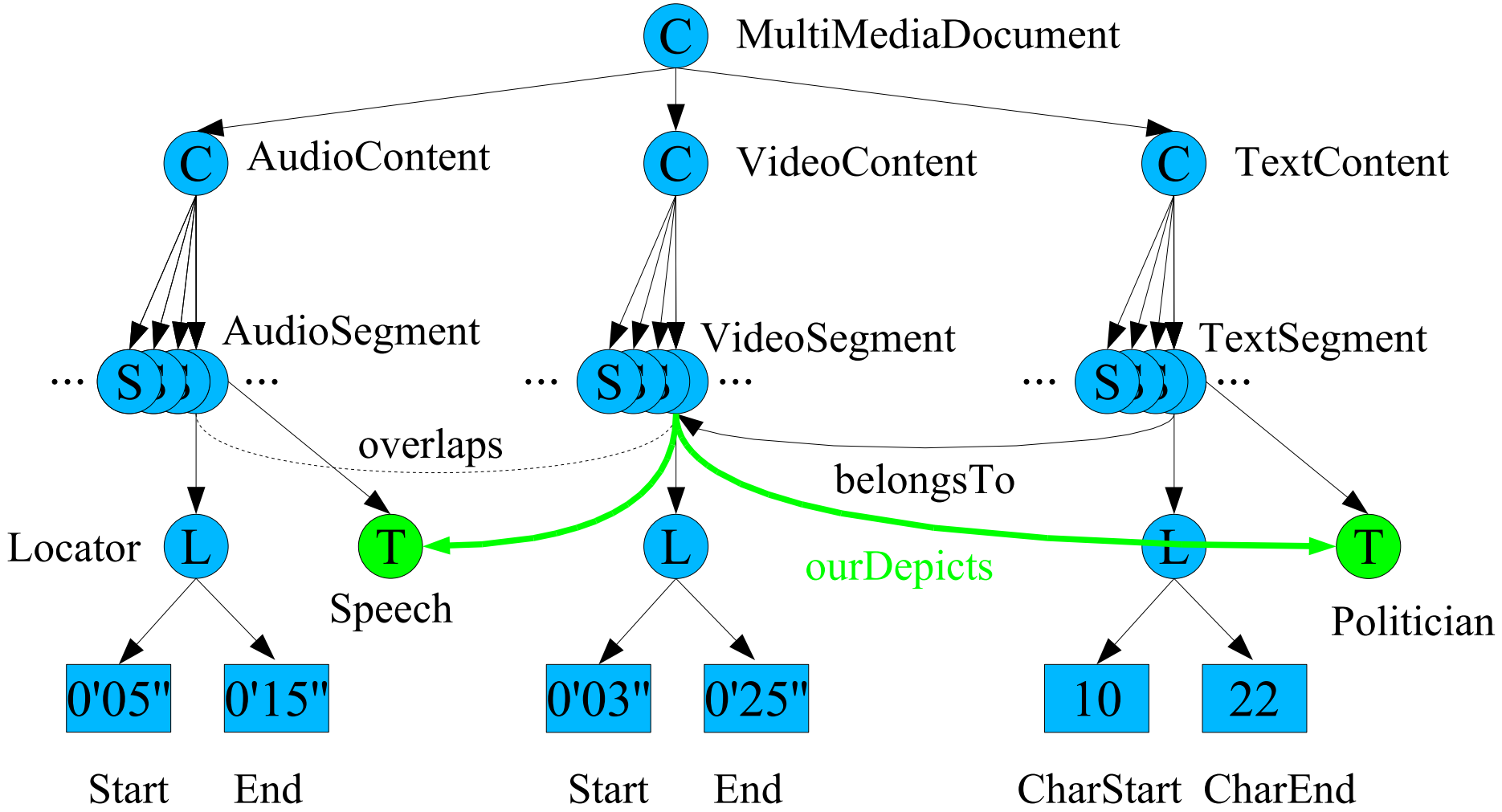
Problems & Solutions – 1

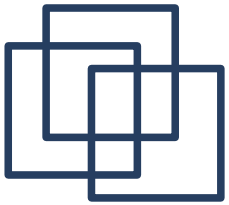
- Scalability
  - Coocurence identification with defined queries,
    e.g. text-to-audio-in-local-video
    - results are not cached
    - Fiat rules get very complex (def. queries treated by unfolding!)
    - relationships were found / computed again and again
    - coocurences have to be found again for linking the interpretation result to the VideoSegment

    Solution: prepare the Abox before processing
    - Establish links from VideoSegments to all „coocurring“ EDO concept instances
    - cooccurence only computed once and result stored
    - EDO information per Segment directly available and explicit

Link Interpretation Result to VideoSegment

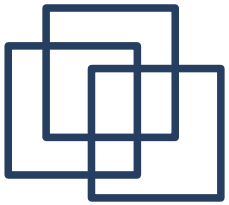
Explanation of FIAT politicianToSpeech: PoliticalInterview





Optimized Fiat Rule „politicianToSpeech“

```
(define-rule (?x ?y #!edo:speechToPolitician)
                  (and
                       (?vs ?x #!mco:ourDepicts)
                       (?vs ?y #!mco:ourDepicts)
                       (?x #!edo:Speech)
                       (?y #!edo:Politician))
  :backward-rule-p nil)
```



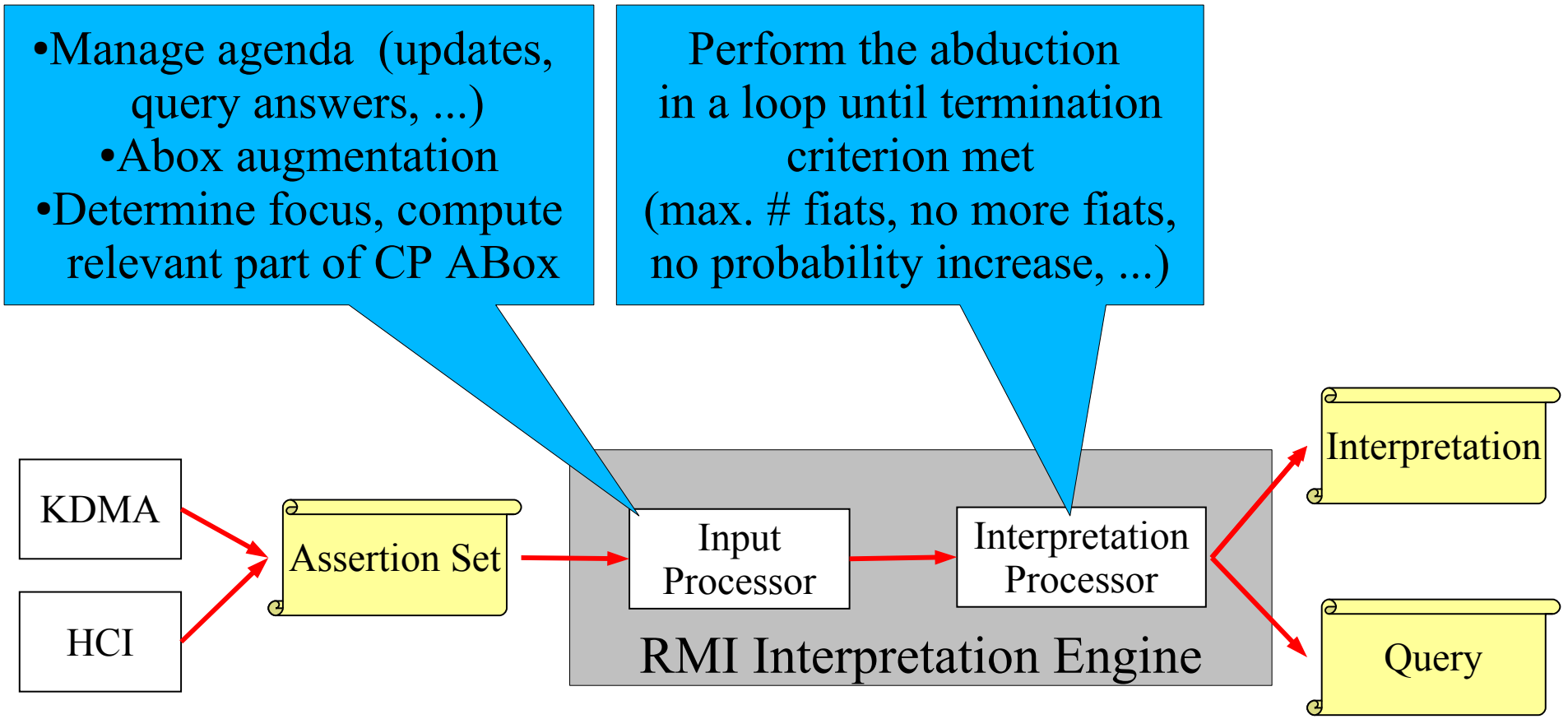
Problems & Solutions – 2

- Realization of Incrementality („stream-based“)
  - for new assertions, identify the affected part of the Abox!
    - add new assertions to global Abox
    - check if there are changes in the „ourDepicts“ relations
    - for affected segments, collect required assertions for interpret.

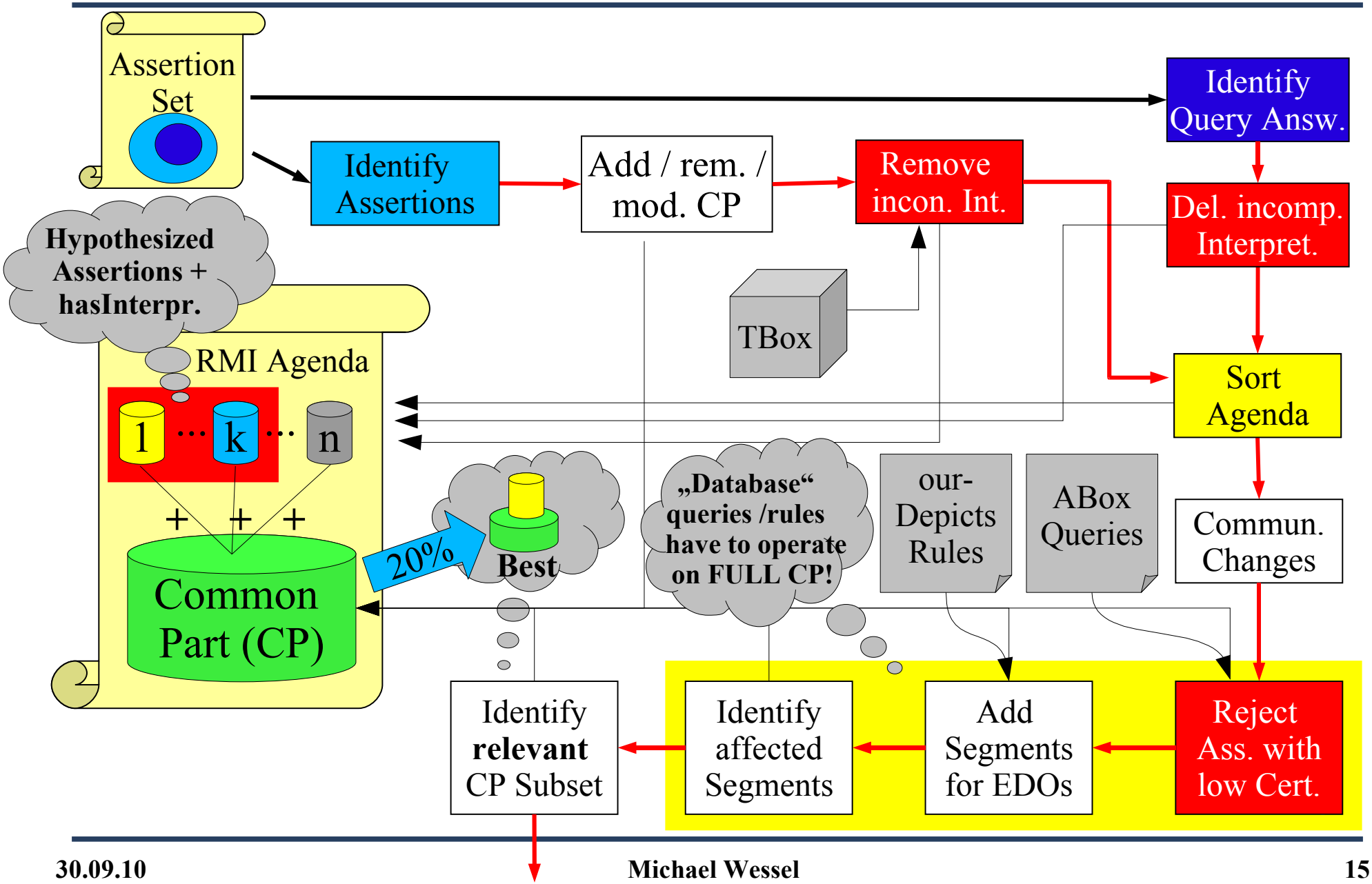
      → 80% reduction
- Multiple interpretations (many!)
  - how to keep the interpretation Aboxes small?

    → decouple the big common part of the intepretations
- Optimization of abduction
- Query generation problem
  - How to inform the client about alternative interpretations?

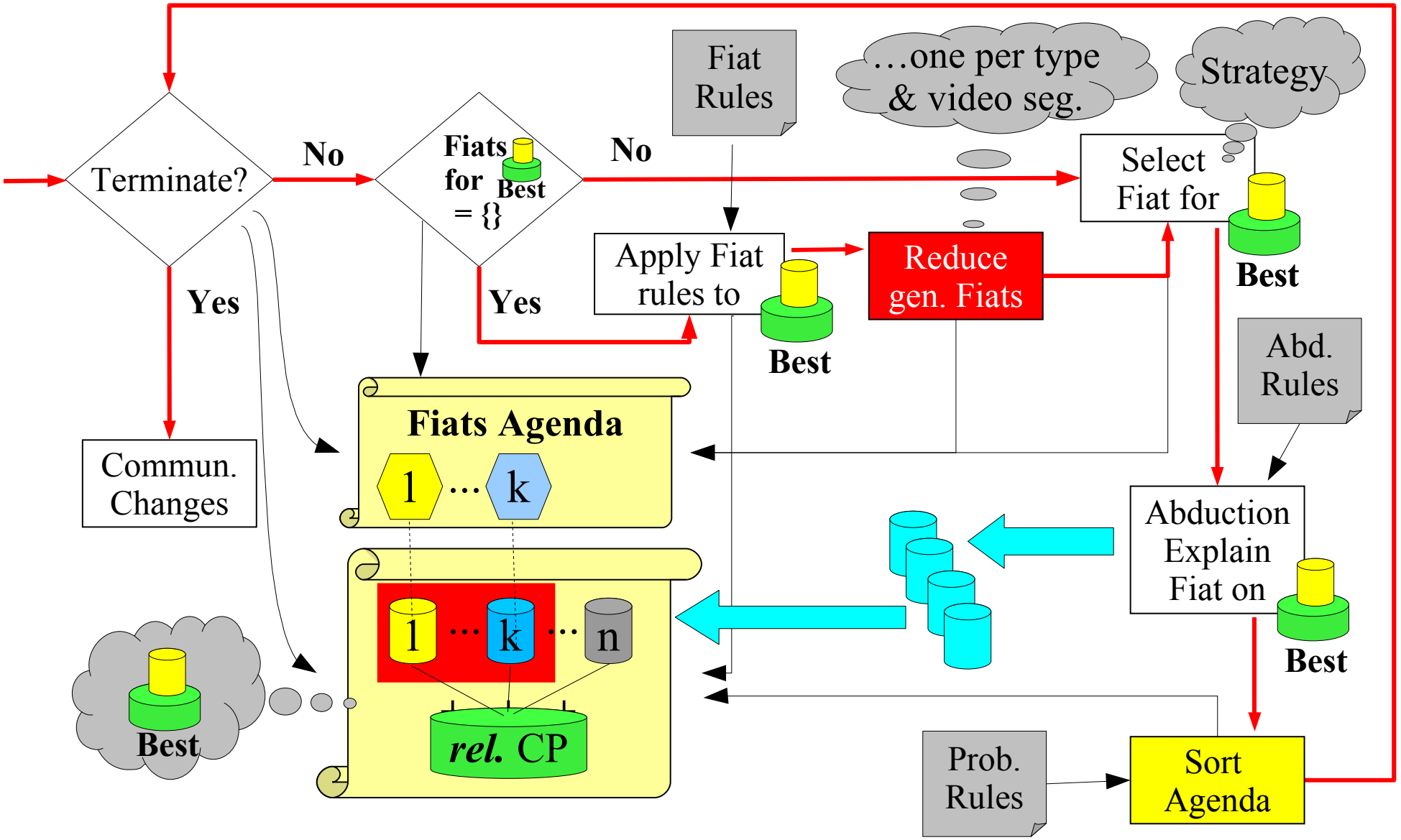
RMI Implementation of `receiveAssertions`



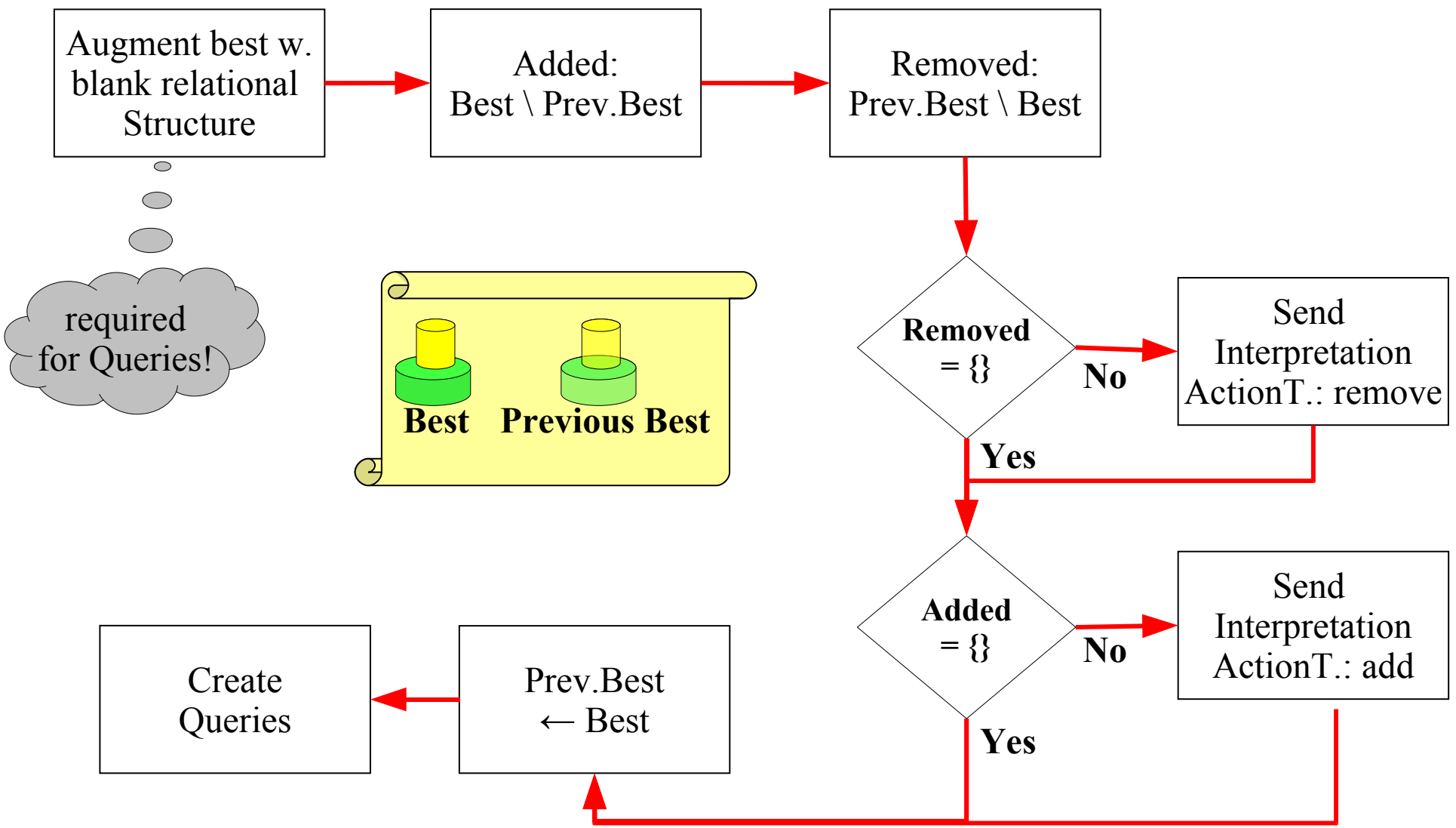
RMI Input Processor



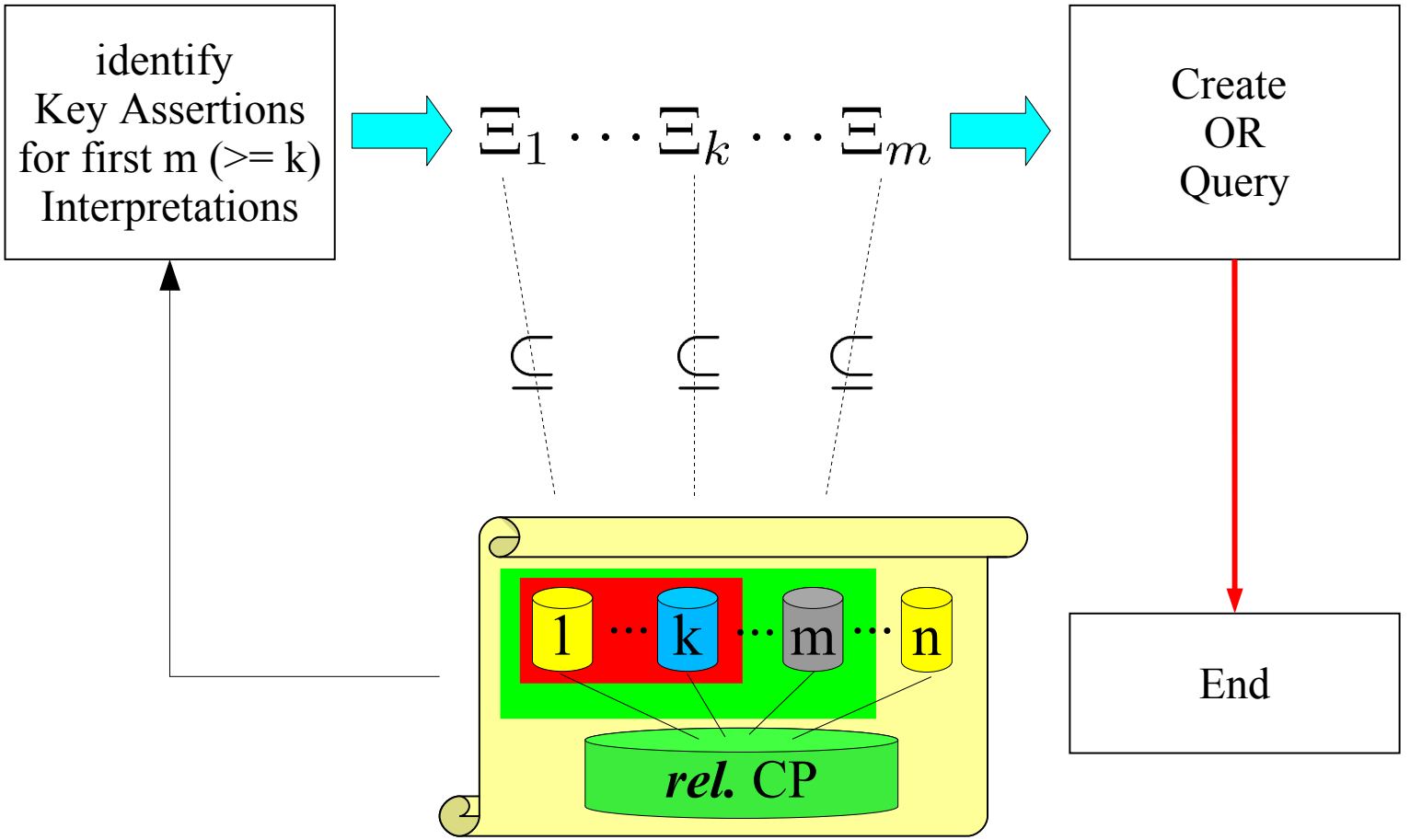
RMI Interpretation Processor



RMI Communicate Changes



RMI Create Queries



Computation of Queries

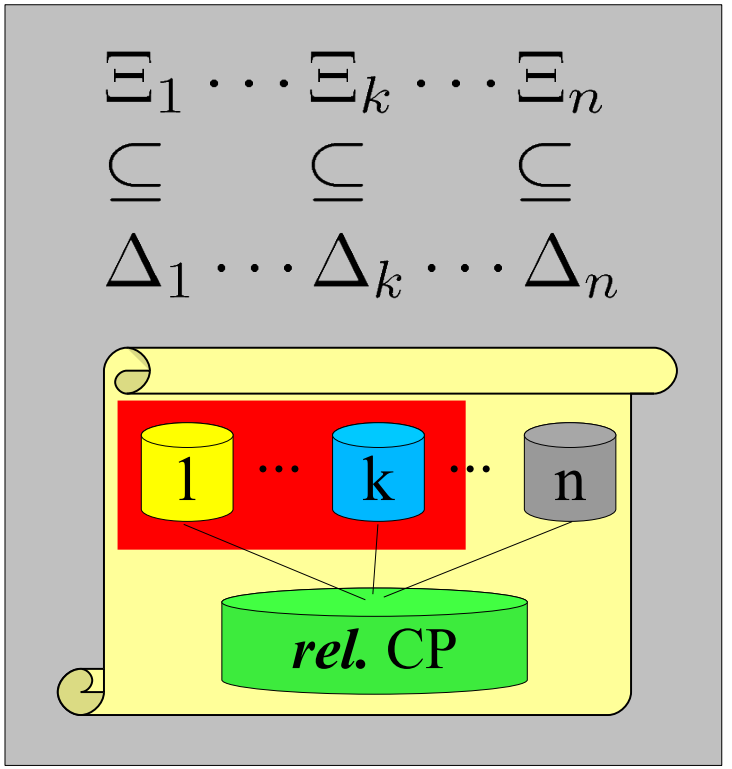
- Computation of characteristic („key“) assertions Ξ_i for $\Delta_i, 1 \leq i \leq n$
- Compute the „common differences“ by intersecting all differences to all other Δ_j

$$\Xi_i = \bigcap_{i \neq j, 1 \leq j \leq n} \Delta_i \setminus \Delta_j$$

- From each Ξ_i select an assertion (preferable an instance assertion)

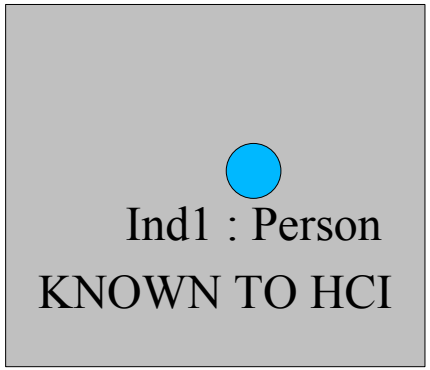
  → n disjuncts for OR query
  → simple score: 1 – 1 / *n*

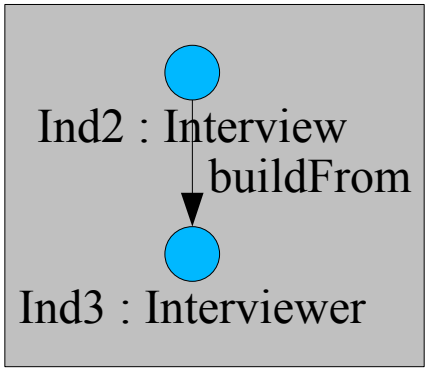
- „\“ may be ABox difference, but...



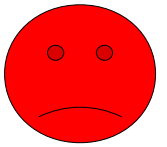
What is the blank relational structure and why is it required?

- Problem:
  - queries can only be formulated against the communicated „best“ interpretation: Δ_i
  - However, all but one query disjuncts come from $\Xi_j \subseteq \Delta_j$
  - the relational structures may be completely different
    - different hypothesized **RMI INDs**, different edges, etc.
- Example: how to communicate the difference between





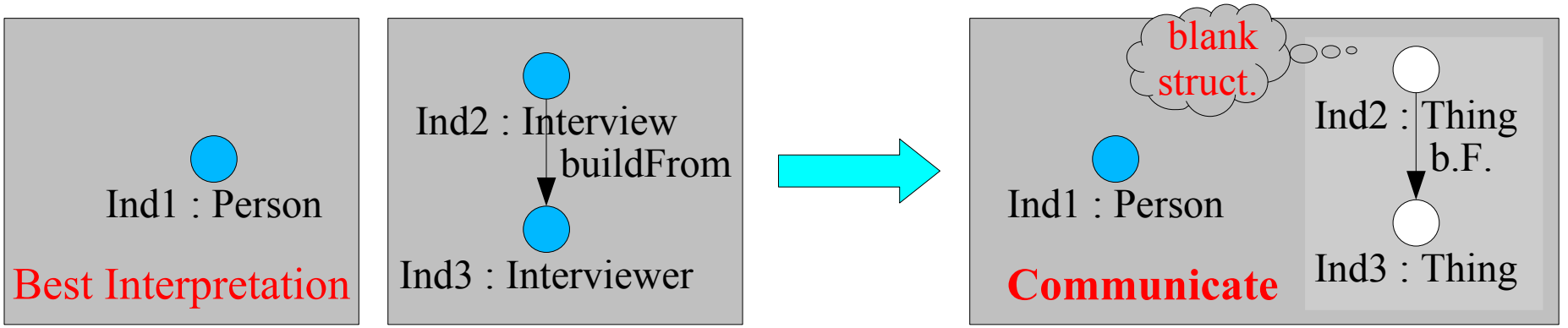
- HCI only knows **Ind1**!
- Q-Disjunct1: **Ind1 : Person**
- Q-Disjunct2: **Ind1 : Interview** ??
  Ind1 : Interviewer ??
- **Solution: avoid the problem in the first place!**

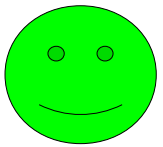


What is the blank relational structure and why is it required? (2)

- Instead of only sending the best interpretation, we also include the „blank relational structure“ of ALL other interpretations

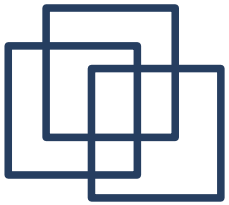
  → relational structure and all hypothesized INDs known to HCI





- HCI knows **Ind1, Ind2, Ind3**!
- Q-Disjunct1: **Ind1 : Person**
- Q-Disjunct2: **Ind2 : Interview**
  [Ind3 : Interviewer]
  [(Ind2, Ind3) : b.F.]
- **No „new-ind mapping“ needed**

Augment best w. blank relational Structure



Abductive Query Answering

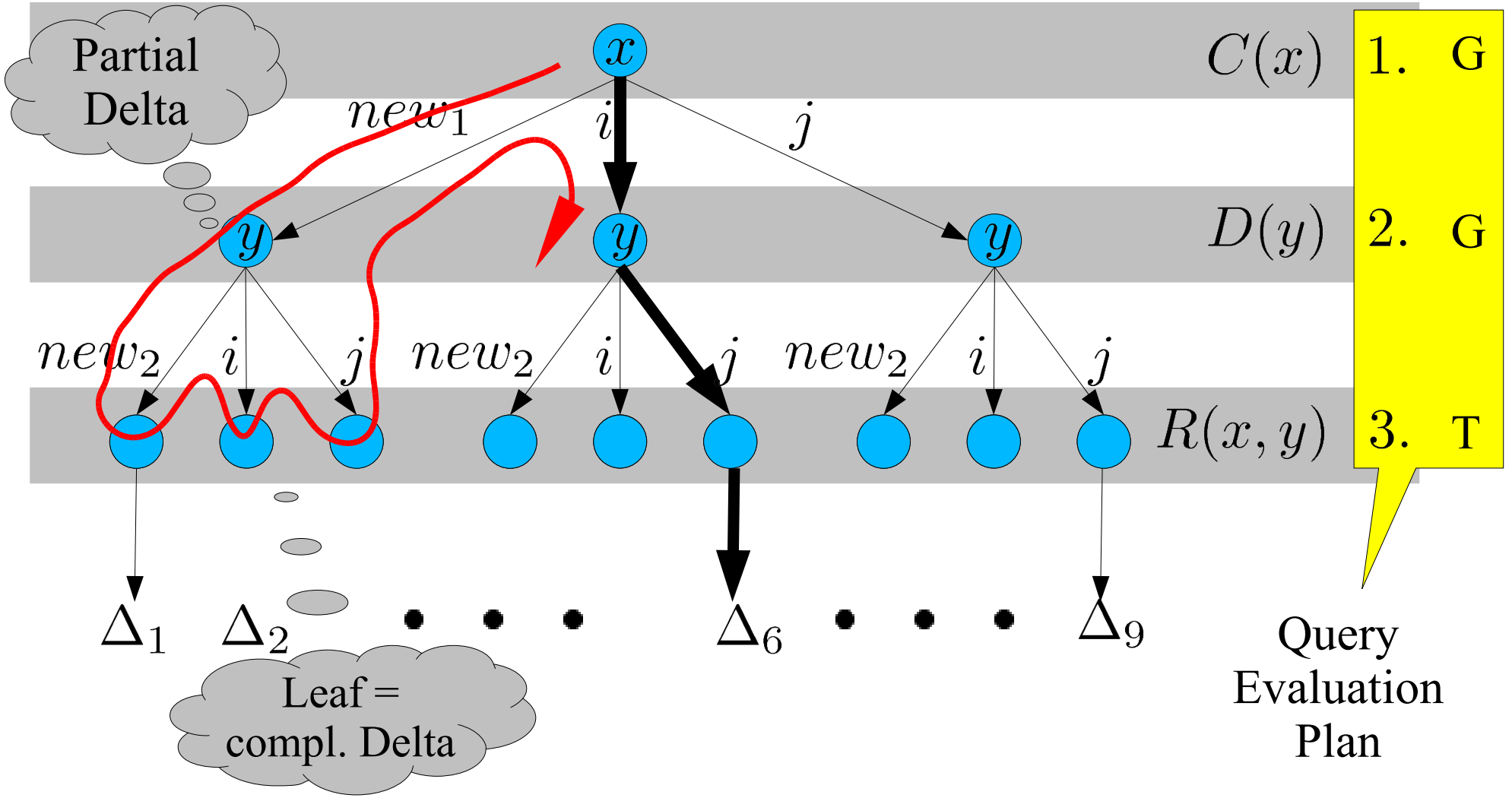
- Simple example
  - Query: $ans() \leftarrow C(x), D(y), R(x, y)$
  - Abox: $\{(i, j) : R, i : C\}$
  - **Preferred** solution (optimal, according to score defined below)

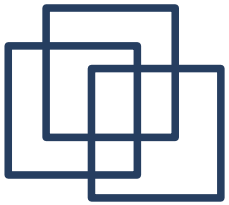
    $x \leftarrow i, y \rightarrow j :$
    $\Delta = \{j : D\}$
  - **Other** solution (plus 7 more, $3^2 = 9$), e.g.

    $x \leftarrow new1, y \leftarrow new_2 :$
    $\Delta = \{new_1 : C, new_2 : D, (new_1, new_2) : R\}$
- Exponential number of solutions has to be computed to find „the best“
  - **optimization idea:** early dynamic cutoff of search space based on score evaluation on partially computed explanations (deltas)

„Depth First“ Abductive Query Evaluation

$$\mathcal{A} = \{(i, j) : R, i : C\}$$





CASAM Preference Score

Very simple:

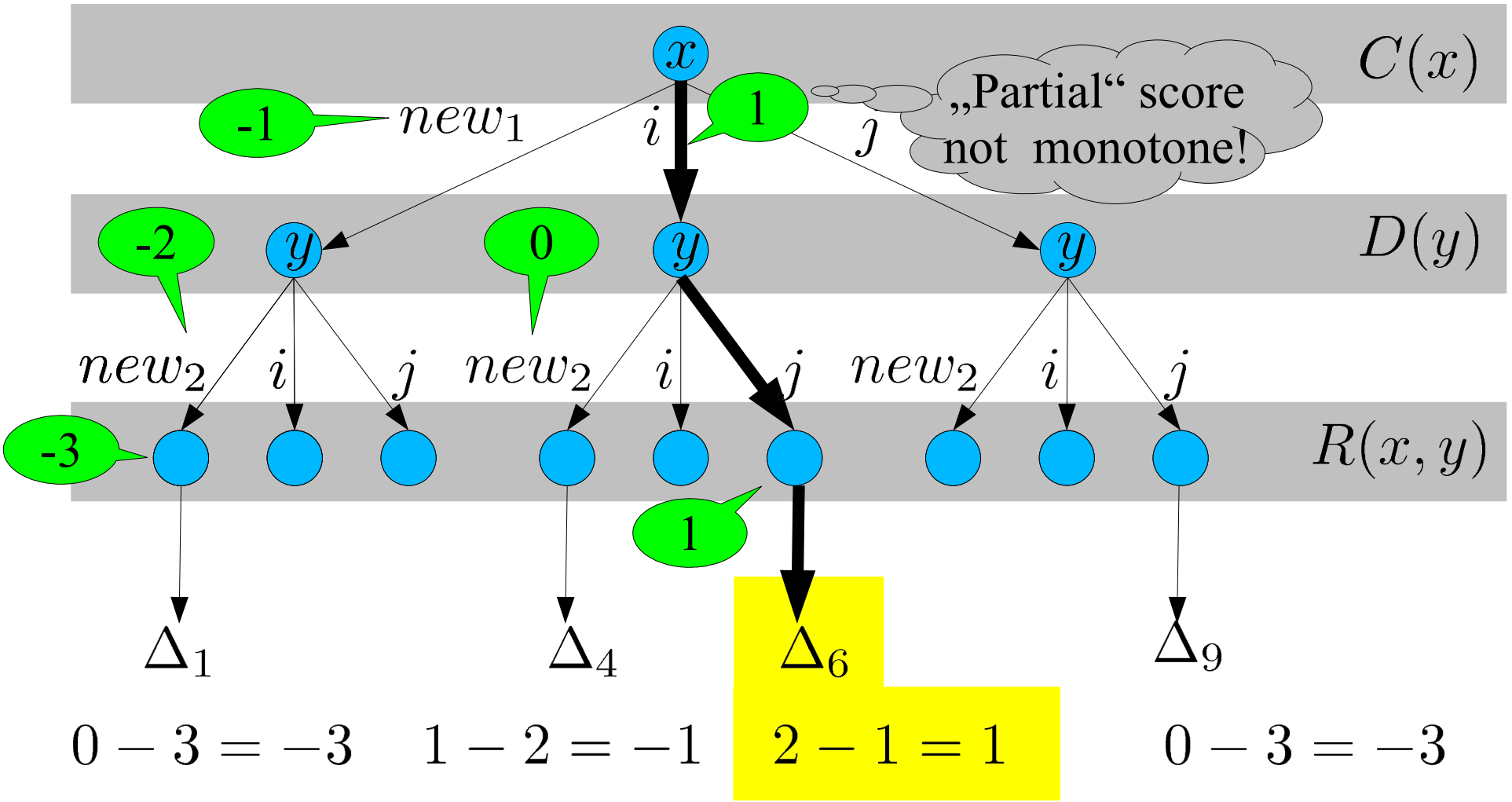
entailed Assertions minus hypothesized Assertions

$$\mathsf{score}(\Delta) =_{def} |\Delta^{+}| - |\Delta^{-}| \rightarrow \text{ maximize}$$

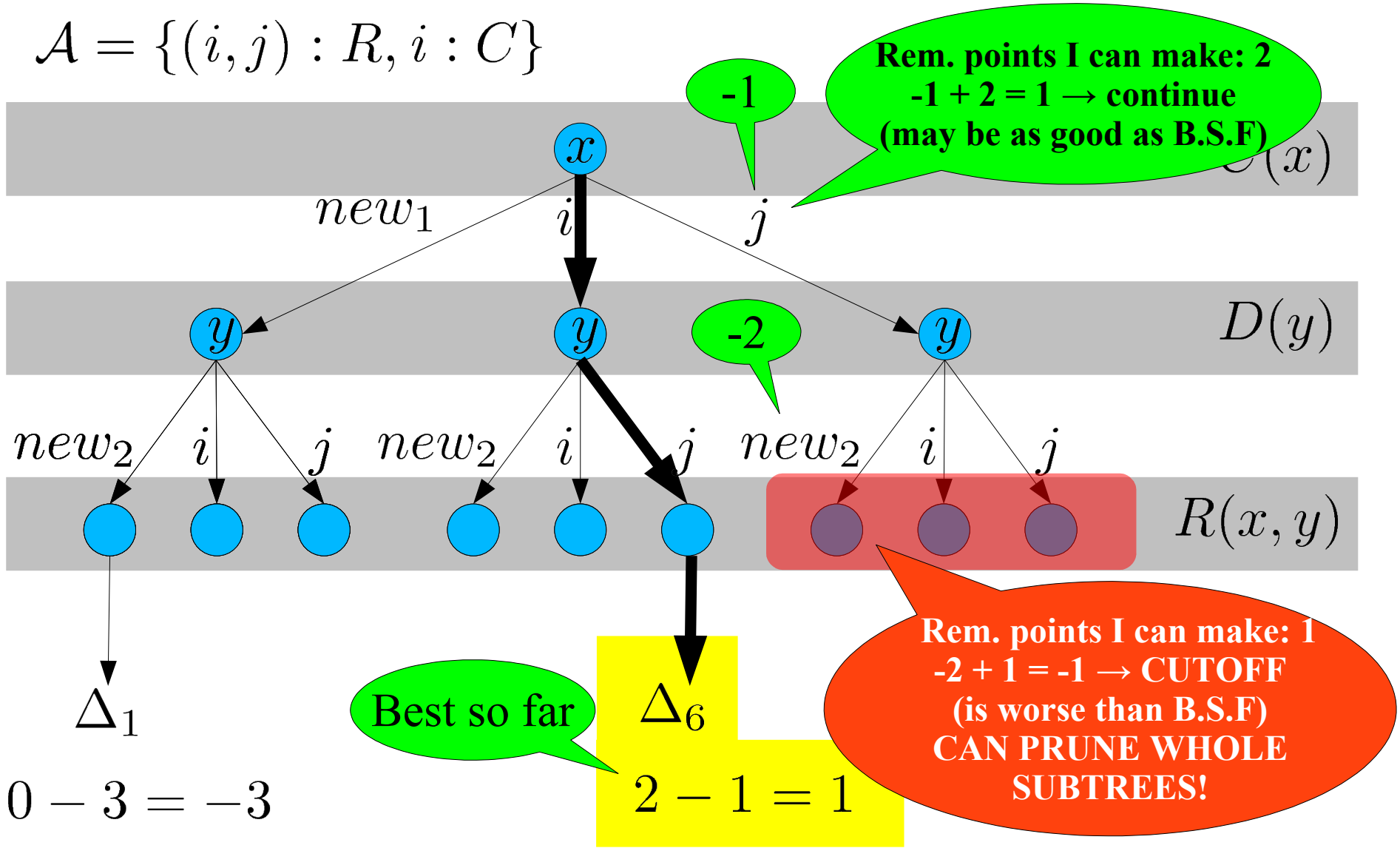
$$\Delta = \Delta^{+} \cup \Delta^{-} \text{ (entailed, hypothesized)}$$

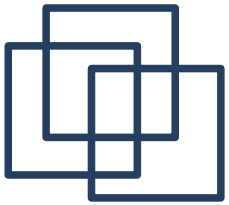
Illustrations of (Partial) Scores

$\mathcal{A} = \{(i,j) : R, i : C\}$ $\quad |\Delta^+| - |\Delta^-| = score \rightarrow max.$



Score-Based Cutoff of Search Space





More formally...

$n = |\Delta^+| + |\Delta^-|$ (n const. for each rule body)

$\mathsf{score}(\Delta) =_{def} |\Delta^+| - |\Delta^-| \rightarrow$ maximize (not monotone)

$n + \mathsf{score}(\Delta) = 2|\Delta^+|$

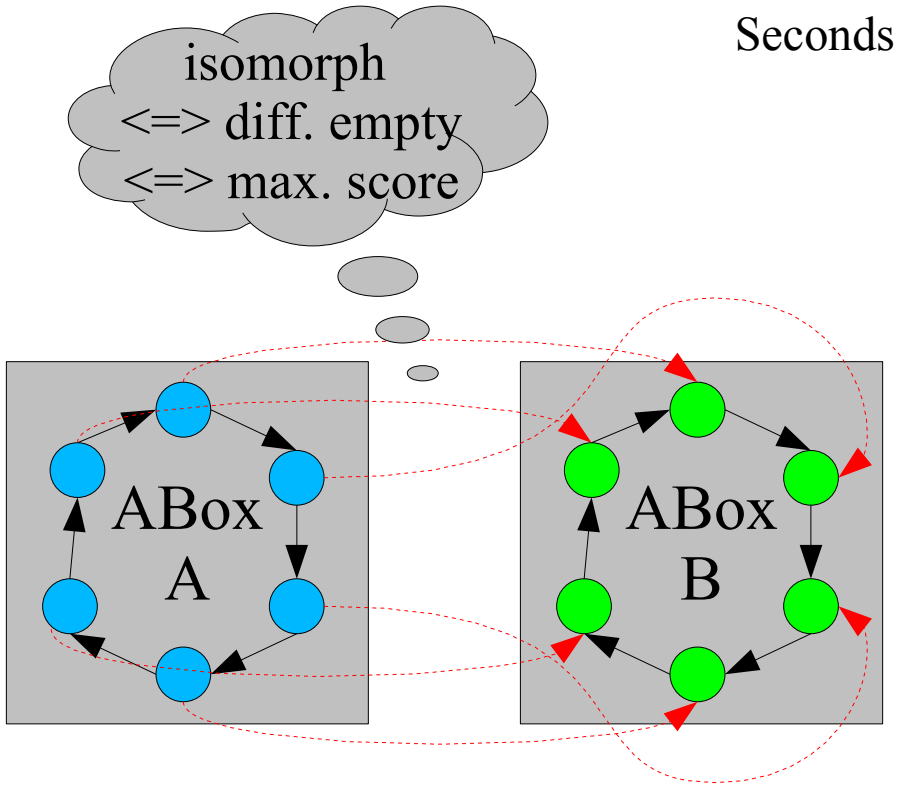
$\mathsf{score}(\Delta) = 2|\Delta^+| - n \rightarrow$ maximize (and monotone!)

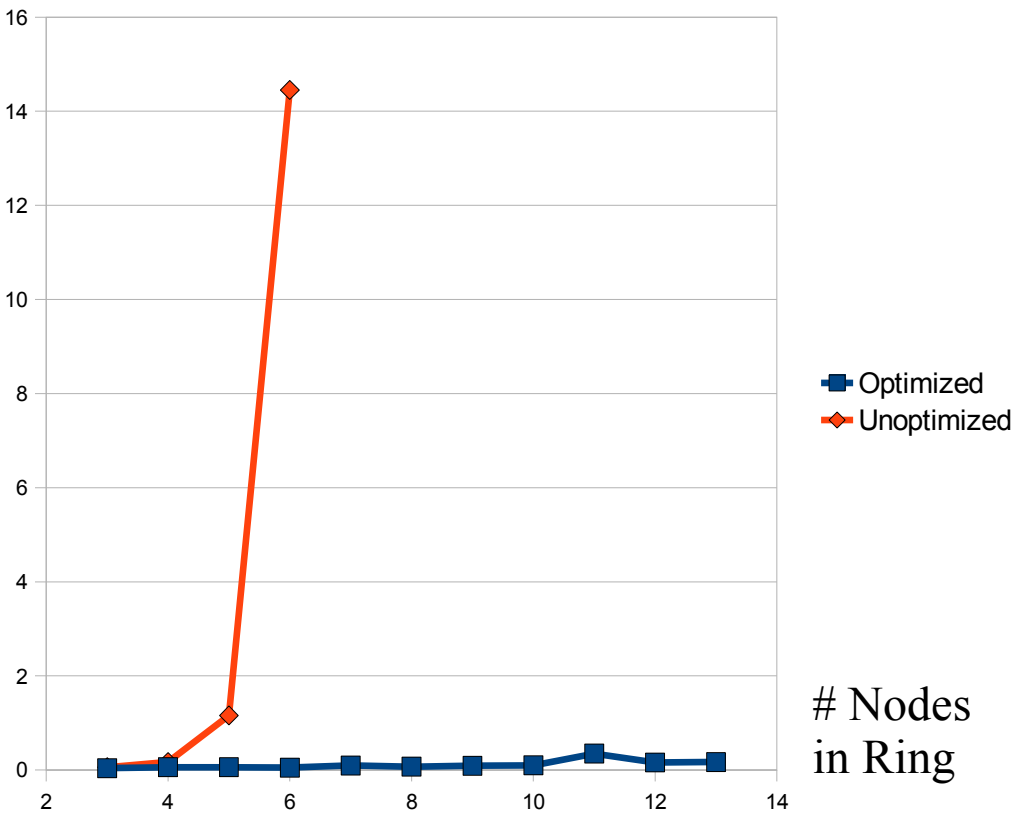
- Let $\Delta_p \subseteq \Delta, m_p = n - |\Delta_p|$ (remaining conjuncts)
  - If $\mathsf{score}(\Delta_p) + (n - |\Delta_p|) < \mathsf{score}(\Delta_{best\_so\_far})$
    $\mathsf{score}(\Delta_{best\_so\_far}) - \mathsf{score}(\Delta_p) > (n - |\Delta_p|)$
    reject Δ_p

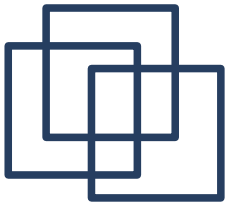
How Effective is this?

- Synthetic benchmark: finding graph isomorphisms (n nodes)
- Problem reductions:

  Graph Isomorphism → ABox Difference → Abduction







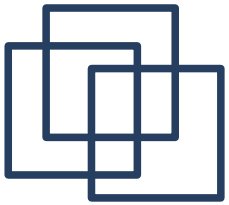
Appreciation of Complexity

- Some numbers
  - video 6, after bunch 3: 283 Fiats (new rule set)
    - potential quadratic number of Fiats (in terms of inds in the Abox)
    - after reduction „only one Fiat per type and shot“: 46 Fiats

Reduce gen. Fiats

  - „external complexity“ of interpretation loop
    - each Fiat may generate 2 to 3 explanations
    - branching will easily kill the system
  - „internal complexity“ of abduction (hidden in RacerPro)
    - in order to find these 2 to 3 best explanations PER FIAT, yet another exponential number of explanations has to be considered!
    - exponential in the number of indiviuals in the ABox

→ RMI handles **serious complex problems, more must be done for meta reasoning** (we stop after 30 Fiats per bunch)



Open Issues

- Reimplementation of probabilistic valuation and Sort Agenda
- React to removed / confirmed tags
- React to „negative“ query answers
  - only positive query answers considered so far
  - „shuffle“ the interpretations containing the answer assertions to the front of the agenda
- More specific Fiat generation rules
- Anytime / meta reasoning
  - reduce set of assertions if timeout occurs, etc.
  - some dumb strategies already implemented
- Q: do we really have to keep all interpretations on the agenda?