2025_09_03

Parcimonious knowledge-based summary of the annotations of a set of elements

- annotations are not independent; they are organized in an ontology
- elements are described using annotations
  - the structure of the ontology is taken into account: if an element is associated to an annotation, then it is also associated to all the ancestors of this annotation
  - each element can be associated (directly or indirectly) to 0..n annotations
  - likewise, each annotation can describe (directly or indirectly) 0..n elements
- elements of interest: subset of the set of elements
  - the set of the elements of interest can itself be also described using annotations.
  - this description can be inferred automatically, e.g. through GSEA
  - pb1: this usually results in numerous annotations, even after filtering
  - pb2: many of these annotations are interdependent, so there is some redundancy

Hypothesis: we can exploit the relations between the annotations to identify the most important ones and discard the others.

Greedy algorithm (v2)

Intuition: initially, the summary is empty and the candidates are the annotations over-represented in the set of the elements of interest

(2) identify the annotation(s) that have the best compromise between the information they convey (the more informative, the better) and the number of elements of interest it annotates that are not annotated by the summary (which favors general so less informative annotations)

(3) add them to the summary, and discard them from the candidates. Also discard all their descendents from the candidates, as they necessarily annotate a subset of the elements of interest annotated by the annotation(s) we have chosen

(7) discard all the remaining candidates that do not annotate any element of interest that is not annotated by the summary (*)

- repeat the 3 previous steps until candidates is empty or the remaining candidates do not annotate any element of interest that is not annotated by the summary

  ? check that δ guarantees that the 2nd condition would result in candidates == {} at the next step.

(8) prune redundant annotations from the summary (*)
i.e.

- annotations from the summary that have an ancestor in the summary and the ancestor annotates ≥ 1 element of interest that is not annotated by any other annotation from the summary $\Rightarrow$ discard the descendant

- annotations from the summary that have an ancestor in the summary and all the elements of interest are annotated by ≥ 1 annotation from the summary that is different from the ancestor $\Rightarrow$ discard the ancestor

⚠ if > 1 couple (ancestor, descendant) in the summary, the order of pruning probably matters
$\Rightarrow$ figure out the best strategy
? try all the combinations and choose the one that minimizes the size of the summary and and maximize the sum of IC ?

Greedy algorithm (v2)

// initialization

$\forall$ annotation, compute a score that reflects (1) its general relevance and (2) the number of elements of interest it annotates

todo: try

cumulative $IC(a) = IC(a) \times |annotatedBy(a) \cap EOI|$

or cumulative Entropy $= H(a) \times | \ldots$

with $\begin{cases} p(a) = \dfrac{|annotatedBy(a)|}{|Elements|} \\ IC(a) = -\log_2(p(a)) \\ H(a) = -p(a)\log_2(p(a)) \end{cases}$

candidates = {annotations over represented in the elements of interest}

summary = {}

el^{ts} AnnotatedBy Candidates = {el^{ts} of interest annotated by ≥ 1 over-represented annot}

el^{ts} AnnotatedBy Summary = {}

// incremental aggregation of the summary by pruning
// the candidates

while (candidates ≠ {} and eltsAnnotatedByCandidates \ eltsAnnot.BySum ≠ {})

compute scoreMax = max({score(c) | c ∈ candidates})
cWMS = {c ∈ candidates | score(c) == scoreMax}

remove the annotations from cWMS that have a descendant in cWMS with the same coverage (*)

remove the descendants of cWMS from candidates

summary = summary ∪ cWMS

candidates = candidates \ cWMS

update eltsAnnotatedByCandidates
eltsAnnotatedBySummary

candidates = candidates ∩ annot⁻¹(eltsAnnotByCandidates \
eltsAnnotBySummary)
(remove from candidates the annots that are not associated with any elt not yet annotated by summary)

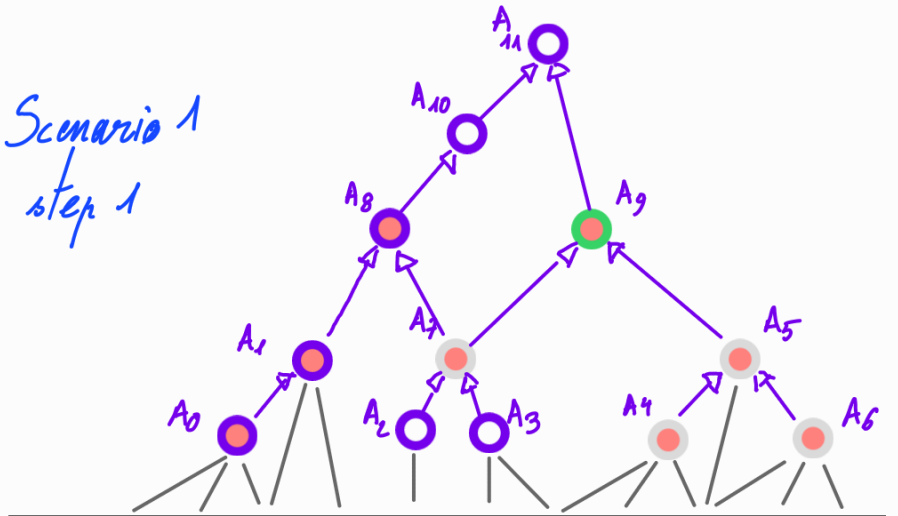
summarize the summary (remove the redundant annotations, cf &) (*)

Greedy algorithm

2 ontology-based tricks (*)

- at each step, if multiple candidates with same coverage and one is a superclass of the other, keep the most precise (subclass) ⇒ informativeness
- at the end, prune the annotations that are covered by 1+ other annotation of the summary (i.e. all its direct superclasses are descendants of the other annotations of the summary) ⇒ concision

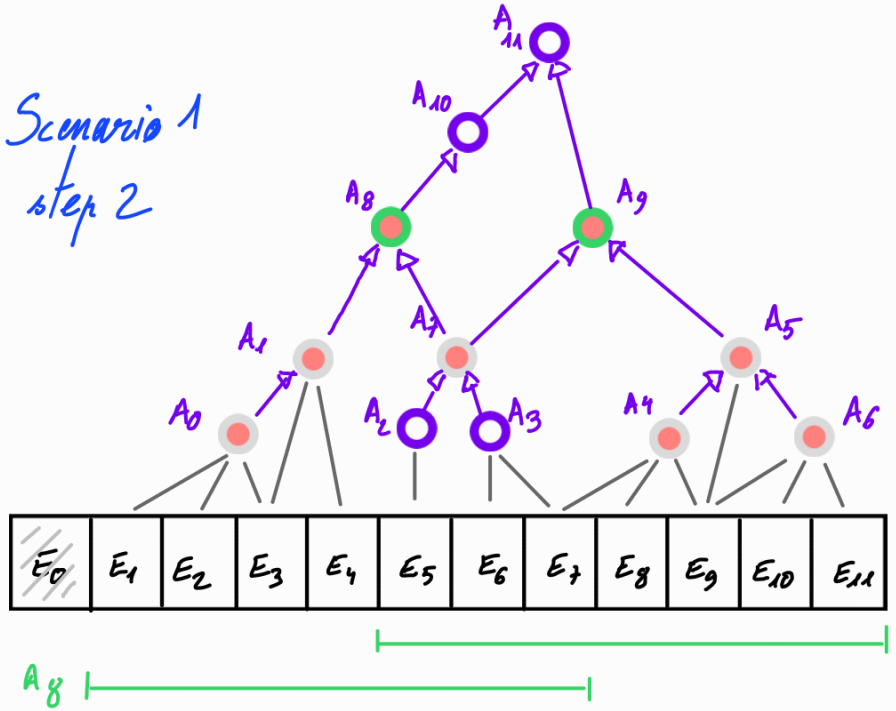
Scenario 1
step 1



- annotation candidate for the summary
- annotation removed from the candidates
- annotation in the summary

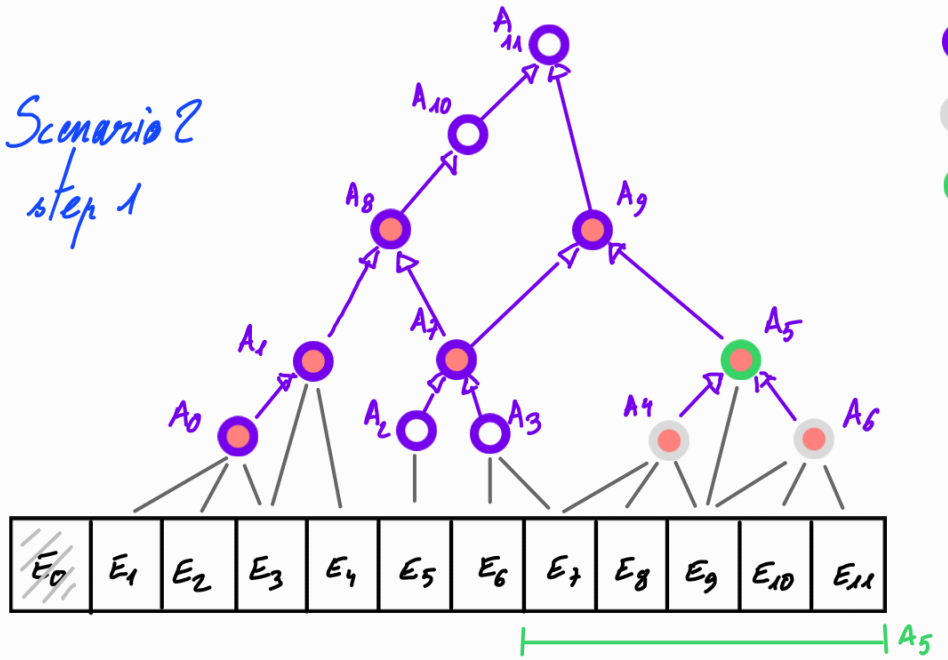
- A_9 selected as best candidate and added to summary
- its descendants are removed from the pool of candidates

Scenario 1
step 2



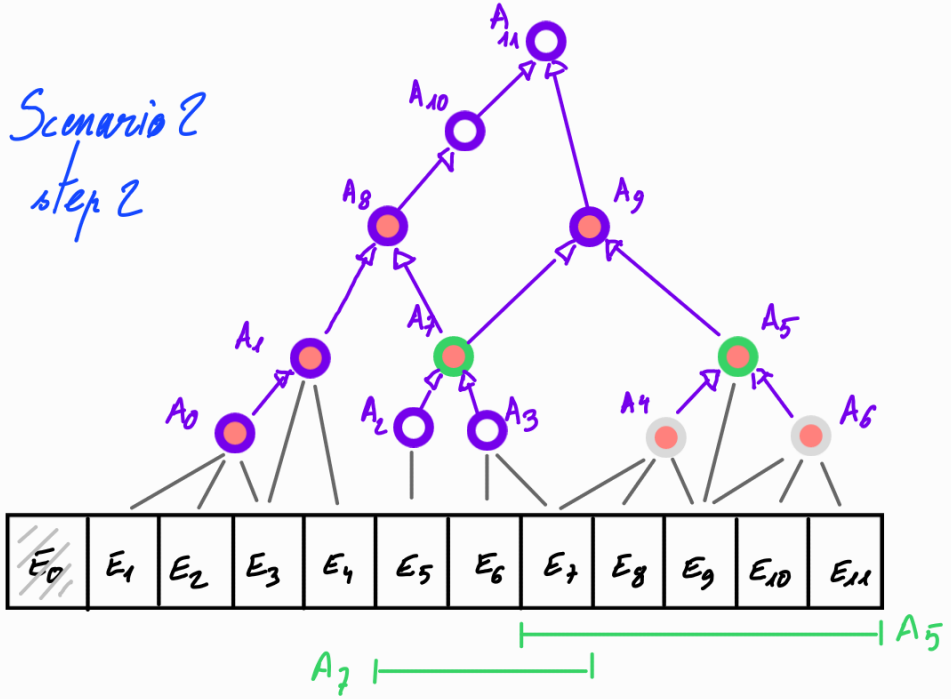
- A_8 selected and added to the summary
- no new candidate can reach new elements ⇒ end of the algorithm

Scenario 2
step 1



- annotation candidate for the summary
- annotation removed from the candidates
- annotation in the summary

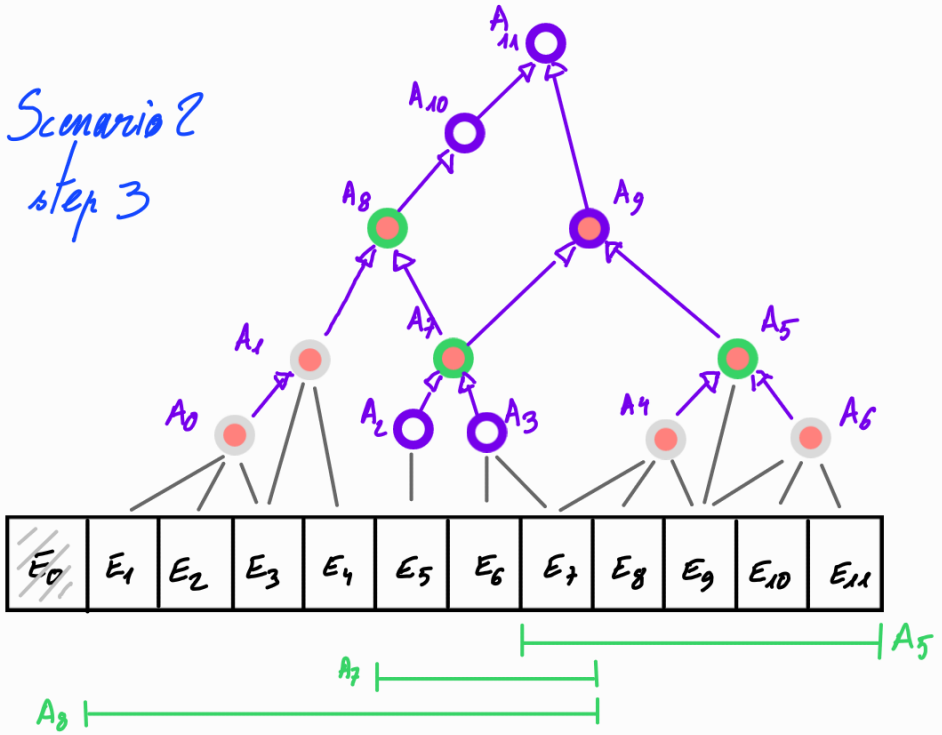
Scenario 2
step 2



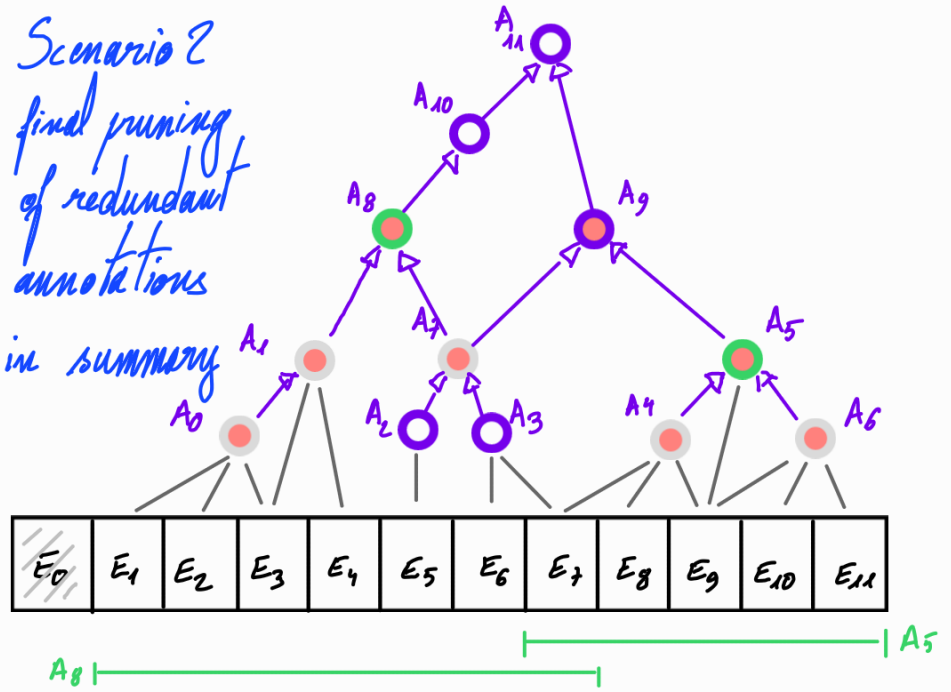
scenario 2 : A_7 better candidate than A_9

A_9 still a possible candidate in future steps but does not annotate new el^{ts} so will never be selected

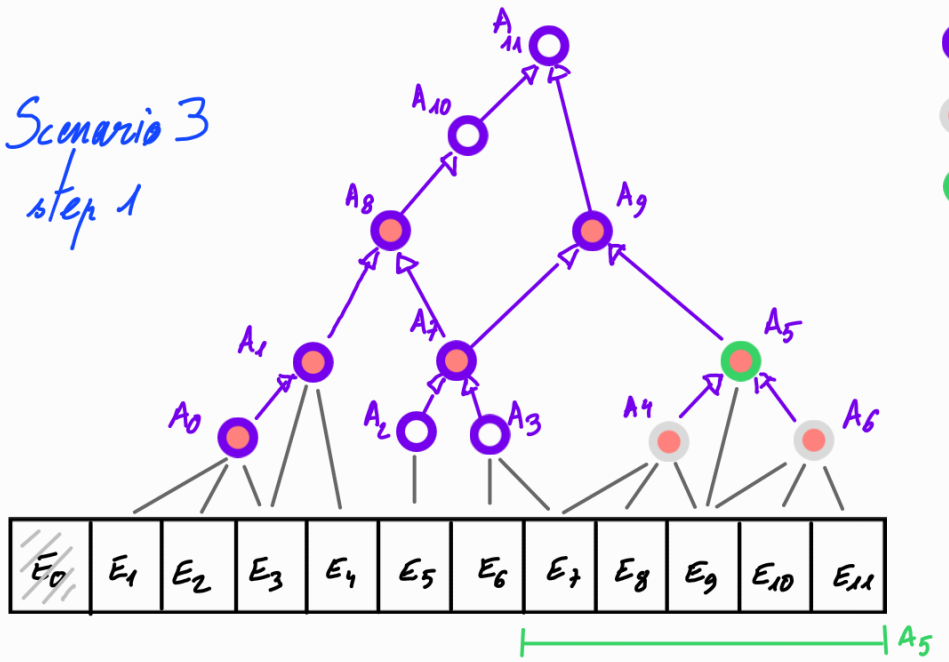
Scenario 2
step 3



Scenario 2
final pruning of redundant annotations
in summary

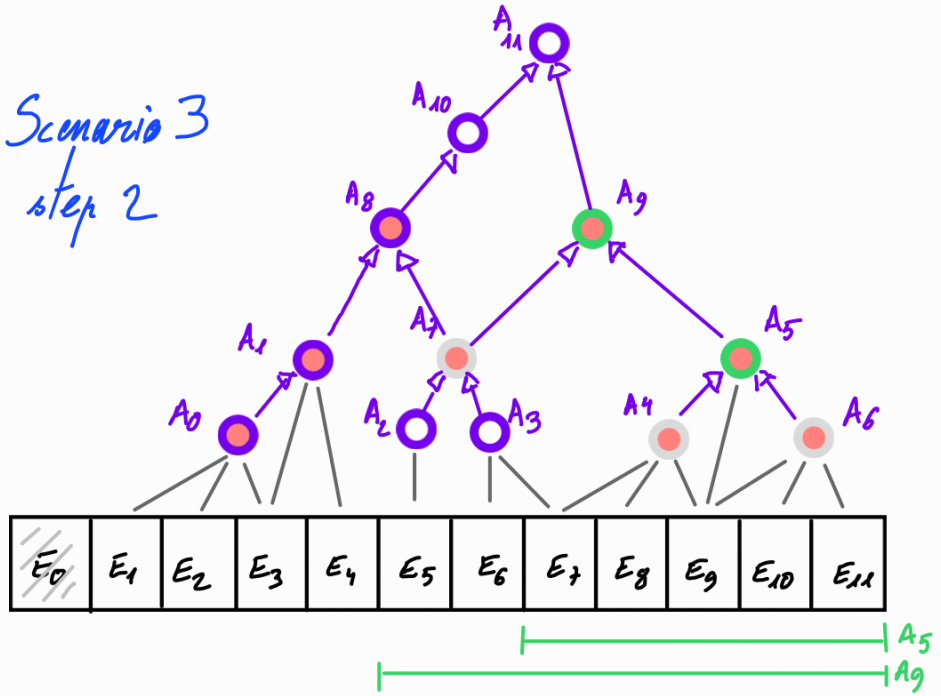


Scenario 3
step 1



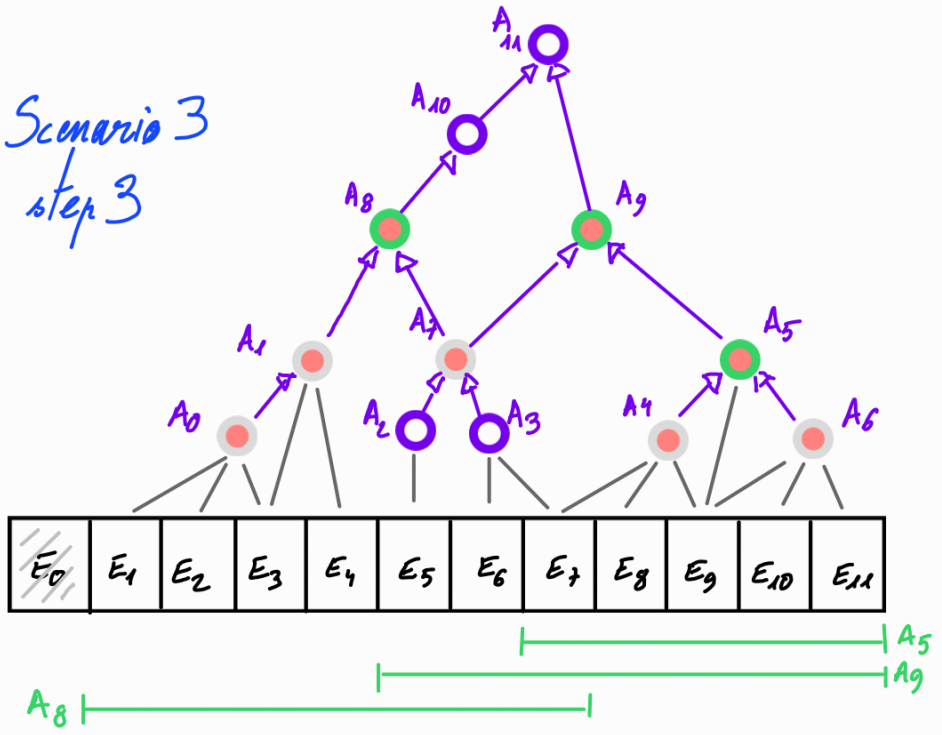
annotation candidate for the summary
annotation removed from the candidates
annotation in the summary

Scenario 3
step 2



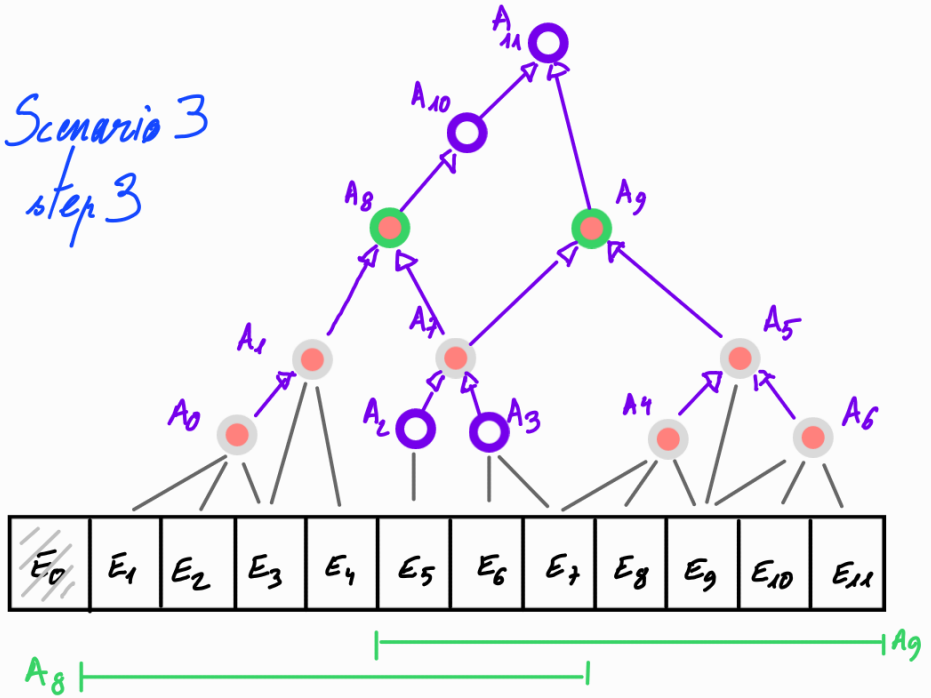
- contrary to scenario2 step 2, A_9 is a better candidate than A_7

Scenario 3
step 3



- select A_8 (or A_1)

Scenario 3
step 3



- find pruning of redundant annotations in summary: discard A_5 (or A_9)