Subtext Refactoring

- extending the tree
  - add new
  - override where it was not inherited
  - introduce indirection
    - before source (indirect all references)
      - compound:
      - add new "i" with ref to source "s" into s's parent
      - change all reference to "s" to "i'
    - after caller
      - compound:
      - add new "i" with ref to caller's source into caller's parent
      - change caller's source to "i"
    - for free vars
      - replace all references to source "s" leaving scope "n" with new child "i" of scope "n" pointing to "s
      - ...watch out for s' capture level.
  - pull up
    - compound:
    - override non-inherited (in source)
    - remove redundant (in copy)
  - override with more specific
    - (watch out for contra-variance ???)
- mapping the (integrated) tree
  - illu
    - A <-> B C
    - A B C
  - move to component
    - mind the capture level!
  - move to aggregate
    - without free var (in B scope)
    - with parameter
    - with overrides
- label morphism
  - "structure of the tree stays the same, only labels change"
  - replace label with fresh label globally
    - isomorphism (not necessary because name is a matter of presentation)
  - replace label l1 with existing l2 label globally
    - "conjoin, subsume, identify'
    - only possible syntactically if l1 and l2 do not both occur on the same node in the root inclusions auftreten
    - changes tree structures if there is a path p with {l1,l2} c inclusions(p
  - replace label l1 with fresh label l2 locally
    - only possible if it is evident at all root overrides with l1 whether l1 is copied from this locatio
    - or possibly introduce new root overrides?
    - possibly introduce indirection l1:l2 to make it read compatible
    - or introduce dynamic forwarder and if possible replace by static ("inline delegate")
- shrinking the tree
  - remove unreferenced override
    - i.e. never occurs as source, or only after further overrides
  - merge equal siblings
    - change source for all uses of sibling 2 to sibling 1 ??
    - remove unreferenced sibling 2
  - extract common source
    - 1 introduce indirection (sibling 1)
    - 2 pull up (sibling 1),
    - 3 change source (sibling 2),
    - 4 remove redundant (sibling 2)
  - push down
    - add an override to each copy of structural ancestors that does not have an override already it
    - remove unreferenced