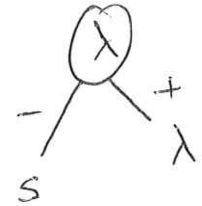
0th: MQ on λ to get the start state of λ.

Initial EQ 

→ get counterexample s, −

→ init T 

classification tree

Operation: Sift(T, s) = figure out what leaf it reaches. (the path)

string (arbitrary)

1st: MQ: sλ → − left, + right

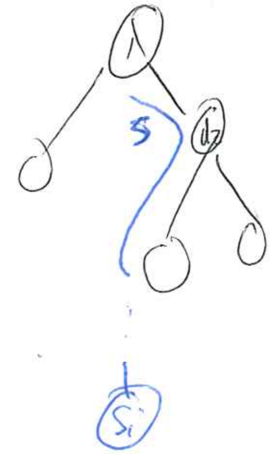
~~...~~

2nd) MQ: sd_2 → − left, + right

3rd: MQ: sd5 etc.

⋮

nth: Reach some leaf s_i, the result.

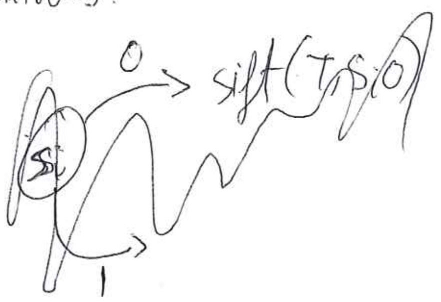


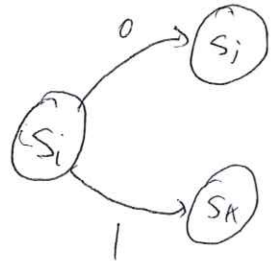
↳ Remark: i) T generates a partition of the space of strings by assigning them the leaf they reach.

ii) Strings that reach the same state of the machine must sift to the same leaf.

Going from T → hypo $\hat{M}$.

- States named by leaves of T
- transitions:





$S_j = \text{sift}(T, S_i 0)$

$S_k = \text{sift}(T, S_i 1)$

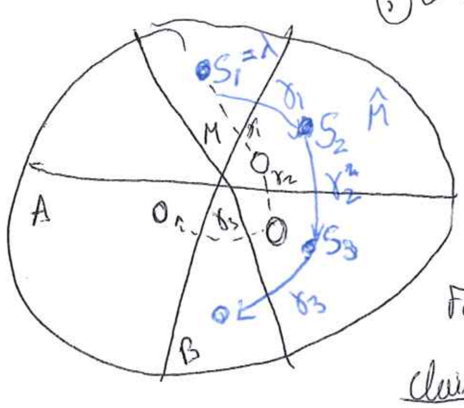
- EQ on $\hat{M}$.
  - → Either they are same and done ✓
  - → There is counterexample. We need to show that we can make progress.

Suppose EQ on $\hat{M}$ gives counterexample $\gamma = \gamma_1 \gamma_2 \cdots \gamma_m$ where $\gamma_\ell \in \{0,1\}$

Assume wlog γ is + in M and − in $\hat{M}$.

Our current tree induces a partition on $\{0,1\}^*$ (on states of M)

(!) Each piece corresponds to one of the current leaves.



(!) We can assume that $S_1 = \lambda$, which corresponds to the start state of the machine.

Follow γ under $\hat{M}$ and M and see where they diverge.

<u>claim</u>: At some point we must end up in different equiv classes / partitions (Bc. λ labels the root of T)

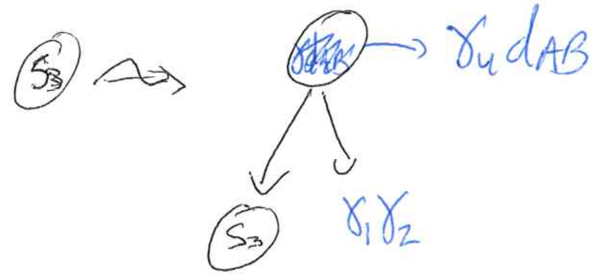
Consider the last place where ~~they~~ are in the same equiv. class.

In the example above, it is after following $\gamma_1 \gamma_2$.

This implies that $\gamma_1\gamma_2$ is a string that reaches a <u>new state</u> that used to be in the same equiv. class as S_3.

than S_3

$\Rightarrow$ We split S_3.



where d_{AB} is the distinguishing string of the least common ancestor of A, B, which are also leaves of our tree.

We continue this process until we have as many leaves as the true machine has states.