2. (3 points) Suppose you are wanted to build a correct morphological analyzer for English.

a) Answer *Correct/Incorrect* to the following proposals. Justify your answers briefly.

(1) Use Finite State Automata (FSA) combined with a POS tagger.

(2) Use FSA combined with a list of word forms with their corresponding analyses.

(3) Use FSA.

(4) Use a list of word forms with their corresponding analyses.

(5) Finite State Transducers (FST) are more effective than the previous proposals.

To analyze the proposed methods for building a correct morphological analyzer for English, let's consider each one individually:

1. **Use Finite State Automata (FSA) combined with a POS tagger.**
   * **Assessment:** Correct, but with limitations.
   * **Justification:** Finite State Automata can effectively model the morphological rules of a language like English, handling regular patterns in word formation. Combining this with a Part-Of-Speech (POS) tagger can enhance the system's ability to deal with contextual variations in morphology. However, FSAs are limited in handling irregularities and exceptions common in English morphology.
2. **Use FSA combined with a list of word forms with their corresponding analyses.**
   * **Assessment:** Correct, and more effective.
   * **Justification:** This approach builds on the strengths of FSAs in modeling regular morphological patterns and supplements it with a comprehensive list of word forms for handling exceptions and irregularities. This combination can effectively deal with both regular and irregular morphological forms in English.
3. **Use FSA.**
   * **Assessment:** Incorrect, if used alone.
   * **Justification:** While FSAs are good at modeling regular patterns, English morphology is replete with irregularities and exceptions that FSAs alone cannot handle. Therefore, relying solely on FSAs would result in a morphological analyzer that is accurate for regular forms but inadequate for irregular ones.
4. **Use a list of word forms with their corresponding analyses.**
   * **Assessment:** Correct, but not efficient.
   * **Justification:** This method can handle both regular and irregular forms by explicitly listing them. However, it might not be the most efficient approach, as it relies heavily on exhaustive lists, making it less adaptable to new words or forms that are not pre-listed.
5. **Finite State Transducers (FST) are more effective than the previous proposals.**
   * **Assessment:** Correct.
   * **Justification:** Finite State Transducers are a more powerful tool for morphological analysis than simple FSAs. FSTs can perform the functions of an FSA but also map between levels of analysis (such as surface forms to lexical forms). This makes them particularly well-suited for handling the complexities of English morphology, including irregular forms, and they can integrate with POS taggers and word lists for even better performance.

b) Given the following forms of verb *take* with their corresponding morphological analyses:

form analysis

*taking* take+VBG

*took* take+VBD

*takes* take*+*VBZ

1. Provide the expressions corresponding to the surface level, the intermediate level and the lexical

level of a FST for each form. Identify clearly each expression with its respective level.

2. Draw the intermediate FST for those forms.

3. Draw the lexical FST for those forms.

4. How are both FSTs combined to produce the result?

1. Surface: taking, took, takes; Intermediate: take^ing#, took#, take^s#; Lexical: take+VBG, take+VBD, take+VBZ
2. Intermediate forms

T – a – k --- e^ing#:ing 🡪 end

| |--- e^s#:s------ --> end

|- o – o – k ---- #: -🡪 end

1. Lexical forms

----Irreg\_V-- +VBZ:^s# --🡪 end

|------ +VBG:^ing# --🡪 end

----Irreg\_V\_P -- +VBD: -🡪 end

1. Combination:

*Taking* gets inside the gerund intermediate FST and becomes take^ing# and, after that, goes inside the lexical FST where ^ing# is removed and +VBG is appended.

*Takes* passes through the intermediate FST and gets its intermediate form take^s# due to spelling rules for 3rd person. Then through the lexical FST where the ^s# is replaced by +VBZ

*Took* passes through a simple intermediate FST where # is appended to its end. After that, passes through the lexic FST where the o is mapped into a, the second o into k, the k into e and finally the # is removed and +VBD is appended.