Question 1 (4 marks)

Give a proof for one of the following:

1. $A_{TM}$ is undecidable (do not use a reduction from another undecidable problem).
2. A language is in NP iff it is decidable by some nondeterministic polynomial time TM
3. SAT is NP-Complete (not using a reduction from another NP-complete problem)
4. 3SAT is polynomial reducible to CLIQUE
Question 2 (2 marks)
Draw a picture showing how the following classes of languages relate to each other: regular, CFL, Turing-Recognizable, Turing-decidable, P, NP, EXPTIME. On the side, indicate which relationships embeddings are known to be strict subsets.

Question 3 (2 marks)
Which of the above classes of languages are known to be closed under complementation? Which ones are known not to be closed under complementation? Which ones are we not sure about? (no explanation is needed)

Question 4 (1 marks)
List 3 undecidable languages that we discussed in class. Give the languages in their full set notation (not their abbreviated name, such as $A_{TM}$). Do not use any language mentioned in this exam.
**Question 5** (2 marks)

Give a proof that $EQ_{CFG}$ is undecidable by using a mapping reduction $f$. You can make use of that $ALL_{CFG}$ is undecidable.

**Question 6** (1 marks)

What does it mean to say that $A$ is NP-complete (what are the two conditions that need to be satisfied)

**Question 7** (2 marks)

Give a proof that $E_{DFA} = \{\langle A \rangle | A \text{ is a deterministic finite automata and } A \text{ accepts no string} \}$ is decidable. Prove this directly, do not use a reduction with another language.
Question 8  (3 marks)
In class, we gave a reduction of $A_{TM}$ (acceptance problem for TM) to MPCP (modified Post Correspondence Problem, in which a specific tile must be used to begin). We map computation histories of a TM to MPCP, where there is an accepting computation history iff MPCP has a solution.

Here, give a reduction from $A_{DFA}$ (acceptance problem for a DFA) to MPCP. In your proof, define what a configuration for a DFA is (which is a simpler version of a computation history for a TM). The configuration should be as simple as possible, but fully capture everything about the computation that impacts whether the DFA’s upcoming computation.

Once you know what the configurations are, a computation history is just a legal sequence of configurations, separated by #.

Create a set of tiles to map computation history to the MPCP problem. As a hint, just as in the reduction of $A_{TM}$ to MPCP, include a special start tile: \(#q_0w_1w_2\ldots w_n#\). Make sure you explain how each $\delta$ transition of the DFA is modeled for in MPCP.

Argue why there will be an accepting computation iff your MPCP problem has a solution.

Question 9  (1 marks)
In the previous question, we reduced $A_{DFA}$ to $MPCP$. However, we know that $A_{DFA}$ is decidable, and $MPCP$ is undecidable. Why is this not a contradiction? (Normally, when we reduce one language to another, it is to show that both are decidable or both are undecidable.)
**Question 10** (1 marks)
What does it mean to say that language $A$ is polynomial time reducible to $B$?

**Question 11** (2 marks)
Show that NP is closed under union. You may use the definition that NP problems have a polynomial time verifier, or the theorem that a nondeterministic TM decides them.