UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

11.3 Computability

Name: _____ Class: _____ Date: _____ **Total: 24 marks**

KEY WORDS Turing machine • halting problem

Objective

Run the diagonal argument and know what stays decidable.

1 Key knowledge and worked examples

■ **The machine model**

Tape, head, states – the formalisation of algorithm ✓. The halting problem: no decider exists; D flips H's answer about itself ✓.

■ **Worked example: the decidable cousin**

DFA emptiness: mark states reachable from q_0; empty iff no accepting state is marked – finiteness makes it searchable ✓.

2 Practice

2.1 State the halting problem in one sentence. [3]

2.2 What does $D(M)$ do in the proof? [3]

2.3 What does Rice's theorem cover? [3]

3 Exam-style questions

3.1 Suppose $H(M, w)$ decides halting.

Write the construction of D from H precisely (both branches). Trace what happens when D runs on its own description. State the contradiction and what it proves about H. State the two mistakes in the claim "so no program ever stops" ✓. [8]

3.2 Three questions, three verdicts:

(1) Does DFA D accept any string at all? (2) Does TM M accept any string at all? (3) Does program P print banana on input x?

Give each verdict (decidable / undecidable) with a one-line reason. For the decidable one, give the algorithm. For (3), name the theorem and give the reduction's wrapper logic. [7]

exercises.lol exercises.com.cn

Solutions

2.1

- Does machine M halt when run on input w? ✓.

2.2

- Runs $H(M, M)$: if it answers halts, loop forever; otherwise halt ✓.

2.3

- Every nontrivial semantic property of programs (what they compute, not how they are written) ✓.

3.1

- $D(M)$: run $H(M, M)$; if the answer is "halts", enter an infinite loop; if "loops", halt immediately ✓.
- $D(D)$: $H(D, D)$ must answer something; either answer makes D do the opposite of what the answer says about itself ✓.
- Contradiction: no consistent H exists – halting is undecidable ✓.
- Mistakes: undecidable is about a general decider for ALL pairs, and specific instances often are answerable – the theorem is not about any program stopping ✓.

3.2

- (1) Decidable: finite graph – check whether any accepting state is reachable from the start ✓.
- (2) Undecidable: $L(M) \neq \emptyset$ is a nontrivial property of what M computes – Rice, or reduce halting by making M accept exactly when the halting computation finishes ✓.
- (3) Undecidable by Rice: wrap P so banana prints only after the wrapped computation halts – a banana-detector would decide halting ✓.

exercises.lol exercises.com.cn