Advanced Undecidability

Rice’s Theorem and Post Correspondence Problem

Semantic properties • Reductions • String matching

Learning outcomes

Recognize semantic properties

Distinguish language properties from machine-description properties.

Apply Rice’s theorem

Prove undecidability using nontrivial properties and mapping reductions.

Analyze PCP

Find matching sequences and explain recognizability and undecidability.

Revision: mapping reductions

Known source

$A_{TM} = \{\langle M,w\rangle : M \text{ accepts } w\}$ is recognizable but undecidable.

Reduction $A \leq_m B$

Construct a total computable f such that $x \in A$ iff $f(x) \in B$.

Direction

If A is undecidable and $A \leq_m B$, then B is undecidable.

Semantic vs. syntactic properties

Semantic

Depends only on L(M). Example: L(M) is finite.

Syntactic

Depends on machine description. Example: M has exactly five states.

Test

Machines recognizing the same language must agree on every semantic property.

Rice’s theorem: formal statement

Theorem

Every nontrivial property P of Turing-recognizable languages gives an undecidable language of machine encodings.

Decision problem

$R_P = \{\langle M \rangle : L(M) \text{ has property } P\}$.

Nontriviality

Some recognizable language has P and some recognizable language does not.

Rice’s theorem: proof setup

Case 1: empty language lacks P

Choose a recognizer T whose language has P. Assume a decider for R_P exists.

Given ⟨M,w⟩

Construct N that, on input x, first simulates M on w.

Gate the simulation

If M accepts w, simulate T on x. If M rejects w, reject x; if M loops, N loops.

Rice’s theorem: proof conclusion

M ACCEPTS w	M DOES NOT ACCEPT w
L(N) = L(T) Property P holds	L(N) = ∅ Property P fails

Thus $A_TM \leq_m R_P$ (map malformed inputs to an empty-language machine). If ∅ has P, apply the proof to the complementary property.

Worked example 1: emptiness

Problem

$E_TM = \{\langle M \rangle : L(M) = \emptyset\}$.

Nontrivial semantic property

A machine rejecting every input recognizes $\emptyset$. A machine accepting every input recognizes Σ^*, which is nonempty.

Conclusion

Rice’s theorem implies E_TM is undecidable, unlike DFA and CFG emptiness.

Worked example 2: regularity and finiteness

Regularity

∅ is regular; $\{a^n b^n : n \geq 0\}$ is recognizable but not regular. Testing regularity of L(M) is undecidable.

Finiteness

∅ is finite; a* is infinite and recognizable. Testing finiteness of L(M) is undecidable.

Lesson

Familiar language classes cannot generally be detected from arbitrary TM descriptions.

When Rice’s theorem does not apply

Machine structure

Number of states is syntactic and decidable by inspecting the encoding.

Halting behavior

Does M halt on every input? Rejecting all inputs and looping on all inputs both recognize $\varnothing$.

Important distinction

Rice’s theorem does not determine decidability for nonsemantic properties.

Post Correspondence Problem (PCP)

Instance

A finite collection of pairs (u_i, v_i) of strings over a finite alphabet.

Question

Does a nonempty sequence of indices $i_1, \ldots, i_k$ exist such that $u_{i1} \ldots u_{ik} = v_{i1} \ldots v_{ik}$?

Rules

Tiles may be reused. Use the same index sequence on both rows. The empty sequence is not a solution.

Worked PCP example: matching tiles

Tile 1	Tile 2
a	ba
ab	a

Choose indices 1, 2

Top: a · ba = aba

Bottom: ab · a = aba

The strings match: (1, 2) is a valid PCP solution.

Worked PCP example: no solution

Instance

Tile 1 = (a, b); tile 2 = (aa, bb).

First-symbol argument

Every nonempty top concatenation starts with a. Every nonempty bottom concatenation starts with b.

Conclusion

No index sequence can match. This proves no solution for this instance, not a general decision algorithm.

PCP is recognizable

Enumeration

Enumerate all nonempty finite index sequences by increasing length.

Check each sequence

Concatenate top and bottom strings and compare them. Accept if they match.

Limitation

If no solution exists, enumeration runs forever. Recognizability does not imply decidability.

Why PCP is undecidable: proof outline

Modified PCP (MPCP)

Require the first tile in the sequence to be a designated tile.

Reduction from A_TM

Encode an accepting computation history with tiles for the initial configuration, legal successors and acceptance.

Reduction to PCP

Fresh separator and boundary symbols enforce the designated first tile. Thus $A_TM \leq_m MPCP \leq_m PCP$. Full tile construction omitted.

Applications and comparison

Rice’s theorem

A general tool for semantic questions about Turing-recognizable languages.

PCP

An undecidable string matching problem; useful as a reduction source for CFG intersection-emptiness.

Shared principle

A finite description does not imply a finite search bound or an algorithm that always halts.

Classroom exercises

Exercise 1

Can Rice’s theorem prove that testing $L(M) = \Sigma^*$ is undecidable? Supply witnesses.

Exercise 2

Find a PCP solution for tiles (ab,a) and (a,ba).

Exercise 3

Does Rice’s theorem apply to testing whether M has exactly ten states? Explain.

Solutions and recap

Exercise 1

Yes: Σ^* has the property and $\emptyset$ does not. Both are recognizable.

Exercise 2

Choose (1,2): top ab·a = aba; bottom a·ba = aba.

Exercise 3

No: state count is syntactic. Count states in the description. PCP is recognizable but undecidable.