Computational Complexity

Systems Thinking · Module 6 — Complexity

Node 6.2

Position in Knowledge Graph

This node formalizes computational complexity as a resource-based measure of problem difficulty.

It builds on: - Complex Adaptive Systems (6.1)

It prepares for: - Information-Theoretic Complexity (6.3) - Kolmogorov Complexity (6.4)

Computational complexity quantifies the cost required to compute system outcomes or solve decision problems.

Formal Statement / Definition / Construction

Let Σ be a finite alphabet.

A **decision problem** is a language:

$$L \subseteq \Sigma^*,$$

where Σ^* denotes all finite strings over Σ.

A deterministic Turing machine M decides L if:

$$x \in L \Rightarrow M(x) = 1, \quad x \notin L \Rightarrow M(x) = 0.$$

Time Complexity

The **time complexity** of M is:

$$T_M(n) = \max_{|x|=n} \text{number of steps } M \text{ takes on } x.$$

Complexity Class P

$$\mathbf{P} = \{L \subseteq \Sigma^* : \exists M \text{ deciding } L \text{ with } T_M(n) \leq Cn^k \text{ for some } C, k > 0\}.$$

Problems solvable in polynomial time.

Complexity Class NP

$$\mathbf{NP} = \{L : \exists \text{ polynomial } p, \exists \text{ verifier } V, \text{ such that}$$

$$x \in L \iff \exists y, |y| \le p(|x|) \text{ with } V(x,y) = 1\}.$$

Verification is polynomial-time.

Assumptions and Conditions

1. Computation model: deterministic Turing machine.
2. Input size measured as string length $|x|$.
3. Asymptotic bounds refer to $n \to \infty$.
4. Resource measured is time; space complexity analogous.

No assumption is made about $\mathbf{P} = \mathbf{NP}$ (open problem).

Proof or Mathematical Development

Polynomial Closure

If:

$$T_1(n) = O(n^{k_1}), \quad T_2(n) = O(n^{k_2}),$$

then sequential composition satisfies:

$$T(n) = O(n^{\max(k_1,k_2)}).$$

Thus class $\mathbf{P}$ is closed under composition.

Reduction

Let $L_1, L_2 \subseteq \Sigma^*$.

If there exists polynomial-time computable function f such that:

$$x \in L_1 \iff f(x) \in L_2,$$

then $L_1 \le_p L_2$.

If $L_2 \in \mathbf{P}$, then $L_1 \in \mathbf{P}$.

Thus polynomial reductions preserve tractability.

Structural Role

Computational complexity:

- Quantifies feasibility of prediction and control.
- Separates tractable from intractable coordination problems.
- Links adaptation to resource constraints.
- Provides formal boundary between solvable and computationally prohibitive dynamics.

In complex adaptive systems, agents operate under computational constraints.

Structural Compression

Invariant

Problem difficulty measured by asymptotic time:

$$T(n) \leq Cn^k.$$

Mechanism

Resource growth under composition and reduction.

Cross-System Echo

- Economics: bounded rationality.
- AI: algorithmic efficiency.
- Networks: routing complexity.
- Control Theory: real-time computability constraints.

Systems Thinking · Module 6 — Complexity · Node 6.2
Concepts: computational-complexity
Prerequisites: 6.1
Enables: 6.3, 6.4

hbar.university — own.your.cognition