Parting
Finite Automata (40s-50s):
Very Simple Model (constant memory)
• Characterize what can be computed (through closure properties)
• First encounter: non-determinism (power of verified guessing)
• Argue/characterize what cannot be computed
• Optimization, learning

More modern (algorithmic and complexity-theoretic) perspective:
streaming algorithms, communication complexity
Chapter II

Computability Theory 30’s – 50’s

Very Powerful Models: Turing machines and beyond

(Un)decidability – what cannot be computed at all

• Foot in the door – an unrecognizable language
• Many more problems, through reductions
• Hierarchy of exceedingly harder problems

The foundations of mathematics & computation

Kolmogorov complexity (universal theory of information)
Chapter III

Complexity Theory: 60’s –
Time complexity, P vs. NP, NP-completeness
• Non-determinism comes back
• Our foot in the door – SAT, a problem that is likely hard to compute
• Many more problems through (refined) reductions
• An hierarchy of hard problems
Other Resources: space, randomness, communication, ...
Crypto, Game Theory, Computational Lens
“evolution of an environment via repeated application of simple, local rules”

~Avi Wigderson
Computational Lens
Hitchhiker's Guide to the Galaxy
Computational Game Theory

Markets computing an equilibrium. Simple dynamics (best response)?

Bounded Rationality:
Prisoners Dilemma
Repeated Games, infinite, finite
Backward Induction
Finite Automata
Always Cooperate, Always Defect, Tit for Tat, Trigger

<table>
<thead>
<tr>
<th></th>
<th>Cooperation</th>
<th>Defection</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>A: 2 years</td>
<td>A: 10 years</td>
</tr>
<tr>
<td></td>
<td>B: 2 years</td>
<td>B: 0 year</td>
</tr>
<tr>
<td></td>
<td>A: 0 year</td>
<td>A: 5 years</td>
</tr>
<tr>
<td></td>
<td>B: 10 years</td>
<td>B: 5 years</td>
</tr>
</tbody>
</table>
Limited Resources
Factoring & One-Way Functions

Given two primes P and Q easy to compute N=PQ. For random such N, assume it is hard to find P and Q. Special case of One-Way Functions (the most basic cryptographic primitives).

Random Instances of SAT that are hard
Zero-Knowledge Proofs
Hardness of learning
Pseudorandom Generators
Deterministically increasing entropy
Randomness is weak
Parting thoughts:

• Computation is a powerful notion, becoming increasingly central.
• Theory allows us to model and analyse computation, reaching non-trivial understanding.
• Much is still open – waiting for you
That's all Folks!