AOC

Rachid Guerraoui, EPFL

This Talk

Advertisement

Complexity of a computation

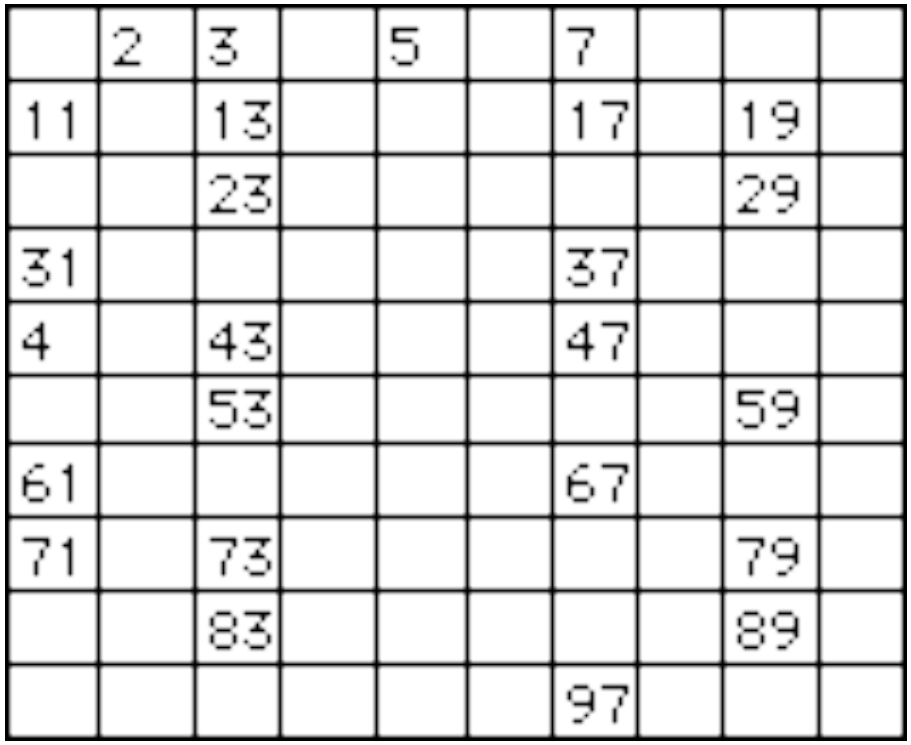
Complexity of a distributed computation

Complexity of a distributed system

Adversary-Oriented-Computing


Lausanne
Vevey
Villeneuve
Nyon
Genève

AOC



NP P

Complexity

Number of steps (cells) of a (batch) program on a Turing machine

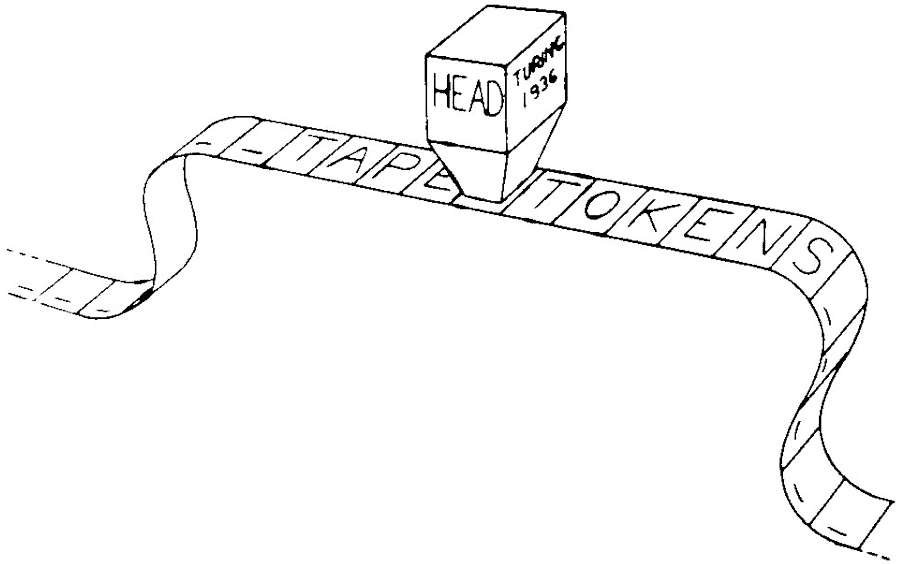


Figure 1 Diagram of a Turing Machine

This Talk

Complexity of a computation C(P)


Isac G.
Tolkien's
The Fellowship of the Ring

What distributed computation?

	2	3		5		7			
11		13				17		19	
		23						29	
31						37			
4		43				47			
		53						59	
61						67			
71		73						79	
		83						89	
						97			

Complexity: C(P)/n ?

Naive distribution is not good enough

Synchronization is needed
(e.g., a counter object)

Distributed Computation

C(O,A)

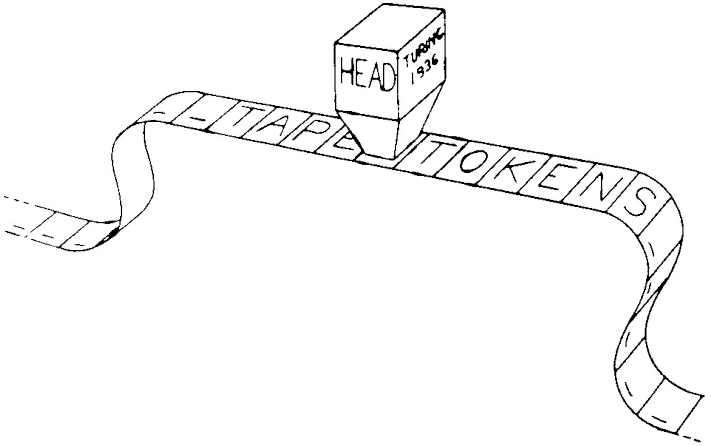
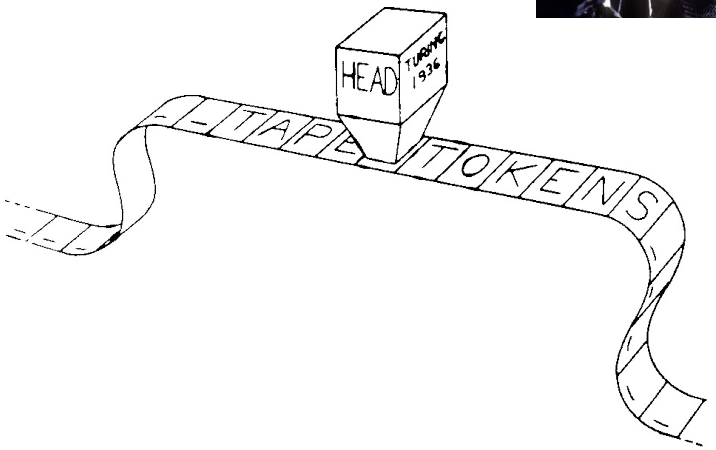
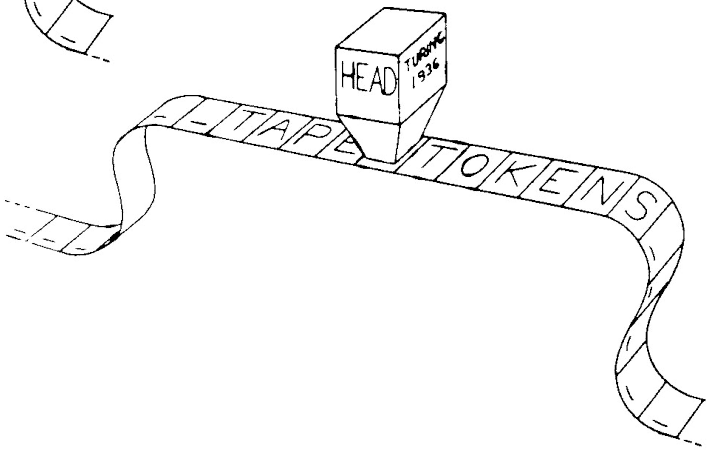


Figure 1 Diagram of a Turing Machine

Figure 1 Diagram of a Turing Machine





What complexity?

No contention: 0 step

Contention: k steps

Node failures: n steps

Link failures: infinity

This Talk

Complexity of a computation C(P)

Complexity of a distributed computation C(P,A)

from centralized to distributed computation

C(P) -> C(P,A)

from distributed computation to systems

C(P,A) -> C(P,A1,A2,..An)

Polymorphic adversary

What complexity?

Polymorphic adversary

C(O,A1,A2,A3,A4) = (1,O(k),O(f),infinity)

The Distributed System's Nightmare

Polymorphic adversary

Paxos saga

STM saga

This Talk

Complexity of a computation C(P)

Complexity of a distributed computation C(P,A)

Complexity of a distributed system C(P,A1,A2,..An)

Adversary-Oriented-Computing

Reduce the pb of addressing several adversaries at a time to the "classical" problem of addressing one adversary at a time

Adversary-Oriented-Computing

Two fundamental questions

(1) How to prove polymorphic lower bounds?

(2) How to develop polymorphic algorithms?

How to prove polymorphic lower bounds?

Algorithmic reductions of adversaries
(STOC 05, ..SPAA 12)

K steps with 

Can be simulated with 

t rounds of a t-resilient synchronous algorithm can be simulated by

a 1-resilient asynchronous algorithm

Two fundamental questions

(1) How to prove polymorphic lower bounds?

Algorithmic reductions of adversaries

(2) How to develop polymorphic algorithms?

AOC (PODC00, ..Eurosys 10, PLDI12, Spaa 12)

Switch(adversary)

Case A1: algorithm_1();

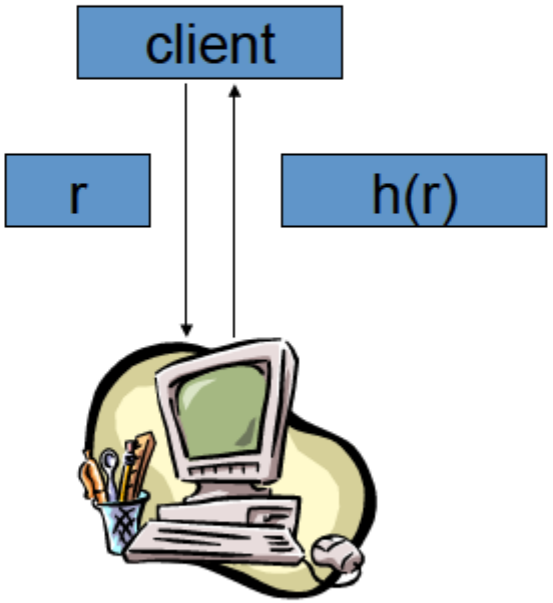
Case A2: algorithm_2();

…

Case Ak: algorithm_k()

…

State machine



Shared Object

Safety: if c1 delivers history h1 and c2 delivers history h2, then one is the prefix of the other

Liveness: if a correct client c invokes a request req, then c eventually delivers response h (req)

AOC Object (A)

Liveness (1): if a correct client c invokes a request req, then c commits or aborts h (req)

Liveness (2) : h (req) is **committed** if the adversary is weaker than A

AOC Object

Safety (1): if c1 commits history h1 and c2 commits h2, then one is prefix of the other

Safety (2): if c1 commits history h1 and c2 aborts h2, then h1 is prefix of h2

Composition Theorem (PLDI 12)

AOC Object A1 + AOC Object A2

=

AOC Object (A1 U A2)

Two fundamental questions

(1) How to prove polymorphic lower bounds?

Algorithmic reductions of adversaries

(2) How to develop polymorphic algorithms?

Modular/dynamic switching of adversaries

This Talk (AOC)

Complexity of a computation C(P)

of a distributed computation C(P,A)

of a distributed system C(P,A1,A2,..An)

Adversary-Oriented-Computing

C(O,A1) C(O,A2) .. C(O,An)

Thank you for your attention