Roots of Distributed Systems
Computation in Space & Time

Distributed Systems

Andrea Omicini
andrea.omicini@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna

Academic Year 2025/2026

Next in Line...

Our Motivation Here I

Impossibility results are theorems

- they are essential
- they *define* the fields—first of all, by properly delimiting the space of the admissible "moves" in the field of distributed systems

! yet, they are *theorems*

- which means, there has to be a well-defined scientific ontology, a shared and non-ambiguous notation, a common definition of what a proof is, ...

? do we have that?

?? do we have that in *computer science*?

"definiscono il campo" di studio e di applicazione.

eppure sono teoremi

Un insieme di concetti e termini definiti in modo rigoroso e accettati universalmente

Una metodologia rigorosa per convalidare i teoremi in modo inequivocabile.

Our Motivation Here II

considerata una congettura (un'affermazione senza prova rigorosa)

Is computer science anything like math?

- Brewer's theorem[Brewer, 2000] was stated without a proper proof
- a first proof came a couple of years later, with some limitations[Gilbert and Lynch, 2002]
- if we go looking through the "hundreds of impossibility results"[Fich and Ruppert, 2003], we will find heterogeneous notations, diverse ontologies, different notions of proof

Nello studio accademico dei sistemi distribuiti, c'è una mancanza di standardizzazione.

- nothing like, say, *math* as we know it today

? should computer science be like that?

e fino a che punto?

? is computer science *supposed to be like that*, and to what extent?

Our Motivation Here III

The point being...

- do we know how computer science should look like?
- do we actually know *what computer science is*, and should be?

Next in Line...

Our Foundations

As computer *scientists* and *engineers*...

- ...we are supposed to ground our *technical knowledge* upon some solid foundations
- accordingly, the answers to the *basic questions* should come to us without any apparent effort

Foundational Questions

What is

- science?
- engineering?
- a machine?
- a computer?
- a system?
- a computational system?
- computer science?
- computer engineering / software engineering?
- computation?

What is Science?

E.g., from the Science Council

http://sciencecouncil.org/about-us/our-definition-of-science/

> *Science is the pursuit and application of knowledge and understanding of the natural and social world following a systematic methodology based on evidence*

- well played, not really surprising
- in the overall, for all the basic questions we have answers that require some deep thinking, nevertheless they more or less belong to our background
- so, first of all, we have to find out *which* one is *the* basic question for us here

Phenomena vs. Noumena

Phenomenon

http://www.britannica.com/topic/phenomenon-philosophy

> *Phenomenon, in philosophy, any object, fact, or occurrence* ***perceived*** *or* ***observed****. In general, phenomena are the* ***objects of the senses*** *(e.g., sights and sounds) as contrasted with what is apprehended by the intellect. . . .*

Noumenon

http://www.britannica.com/topic/noumenon

> *Noumenon, plural Noumena, in the philosophy of Immanuel Kant, the* ***thing-in-itself*** *(das Ding an sich) as opposed to what Kant called the phenomenon—the thing as it appears to an observer. . . .*

So, What Does Science Do, Actually? I

Explanation

- we *observe* phenomena, we record, track, measure them—we call them *facts*
- we a need to understand what we observe, to provide an explanation for the facts
- science looks for the noumena that give reasons to phenomena, the models that explain the facts observed
- there might exist many different models of a given set of facts
  - they are *not* equivalent, in general
- typically, we prefer models with less assumptions and more facts explained—more *expressive* scientific models

? is explanation all that we ask to science?

So, What Does Science Do, Actually? II

Prediction

- a working scientific model does not just *explain* observations
- it also tells us what happens next
  - it does *predict* not-yet-observed phenomena, e.g.
    - the gravitational redshift for Einstein's general theory of relativity
    - the Higgs boson, predicted in 1964 and observed in 2012
- the predictive power of a *scientific theory* is an essential part of what makes a theory a scientific one

What is Computer Science?

"Is there such a thing as computer science, and if there is, what is it?"

Wherever there are phenomena, there can be a science to describe and explain those phenomena. Thus, the simplest (and correct) answer to "What is botany?" is, "Botany is the study of plants." And zoology is the study of animals, astronomy the study of stars, and so on. **Phenomena breed sciences**.

There are computers. Ergo, computer science is the study of computers. The **phenomena** *surrounding computers are varied, complex, rich.* [Newell et al., 1967]

What is the Object of Study of Computer Science? I

What if *computer science is the study of computers*, yet...

The term "computer" is not well defined, and its meaning will change with new developments. [Newell et al., 1967]

A science may bear a shifting object of study

The phenomena of all sciences change over time; the process of understanding assures that this will be the case. Astronomy did not originally include the study of interstellar gases; physics did not include radioactivity; psychology did not include the study of animal behavior. Mathematics was once defined as the "science of quantity." [Newell et al., 1967]

What is the Object of Study of Computer Science? II

Whatever a computer *is*, what does a computer *do*?

A computer computes

What is the Object of Study of Computer Science? III

More generally...

- a computer is a *machine* that computes
- *computer systems*, or computational systems are systems made of computers
- computers and computational systems produce the *evidence*, the *facts*, the *phenomena* that are studied by computer science
- computation is then a core *object of study* of computer science
- the *essence* of *what computation is* represents then the noumenon at the core of computer science

What is the Object of Study of Computer Science? IV

Definitions of *computation* and *computer science* go hand in hand

> *Over time, the definition of computer science has been a moving target. These stages reflect increasingly sophisticated understandings of computation.* [Denning, 2010]

What is the Object of Study of Computer Science? V

"The history of computer science reveals an interesting progression of *definitions* for computer science"[Denning, 2008]

- study of automatic computing (1940s)
- study of information processing (1950s)
- study of phenomena surrounding *computers* (1960s)
- study of *what* can be automated (1970s)
- study of computation (1980s)
- study of information processes, both *natural* and *artificial* (2000s)

Next in Line...

What is Computation? I

Question "What is Computation?" considered as harmful[Freeman, 2011]

It is important to have a common understanding *of fundamentals in order to make progress in any field, but a rigid "standard" that is adopted too early is almost always an* impediment to progress*. Just think of where we would be today if computer science had remained merely a branch of mathematics or engineering or experiment-based science.*

. . . said that, *"What is computation?"* is the basic question

- around which all Computer Science revolves
- where any well-founded study of Distributed Systems should be grounded on

What is Computation? II

Computation is *symbol manipulation*

A computation is a sequence *of simple, well-defined* steps *that lead to the* solution *of a* problem*. The problem itself must be defined exactly and unambiguously, and each step in the computation that solves the problem must be described in very specific terms.*[Conery, 2010a]

. . . a problem, and its solution, must be encoded in the form of symbols; a step is a symbol manipulation that transforms one set of symbols into a new set of symbols[Conery, 2010b]

What is Computation? III

Computation is *process*

> *. . . the essence of computation can be found in any form of process* [Frailey, 2010]

→ so, computation is a *process*, and every process is also a computation [Frailey, 2010]

What is Computation? IV

Process

http://www.oxforddictionaries.com/definition/english/process

process,

1. *A series of actions or steps taken in order to achieve a particular end...*
   1. *A natural series of changes...*
   2. *A systematic series of mechanized or chemical operations that are performed in order to produce something...*
   3. *An instance of a program being executed in a multitasking operating system, typically running in an environment that protects it from other processes...*

What is Computation? V

When defining computation... [Denning, 2011]

- computational model matters
- many important computations are natural
- many important computations are non-terminating
- many important computations are continuous
- computational thinking can be defined

Computational Model

Computing machines

- Turing's computing machine?[Turing, 1937]
- von Neumann's computing machine?[Burks et al., 1982]
- ? they are *artificial*, do they work when we include *natural* computations?
- ? they are *discrete*, do they work when we include *continuous* computations?
- ? they represent one *single* computing device, do they work when we deal with *computational systems*?

What is a Machine? I

Machine

http://www.britannica.com/technology/machine

Machine, device, having a unique purpose, that augments or replaces human or animal effort for the accomplishment of physical tasks. . . . The operation of a machine may involve the transformation of chemical, thermal, electrical, or nuclear energy into mechanical energy, or vice versa, or its function may simply be to modify and transmit forces and motions. All machines have an input, an output, and a transforming or modifying and transmitting device. . . .

What is a Machine? II

Input, output & state of a machine

- *input* is what affects a machine from the outside
- *output* is how a machine affects the outside
- *state* at time t is whatever is necessary to understand the evolution of a machine after t given some input—or, more generally, given the *context* where the machine operates

What is a Computing Machine?

A computing machine is a different sort of machine...

- whose task is *cognitive* instead of physical
- whose input and output are basically *information*—in some form
- whose *context* is...?

Abstracting away from (computing) machinery

- if we choose not to stick with one specific computational model, it might be appropriate to *abstract away* from the *machinery*
- by focussing instead on the *computational process*

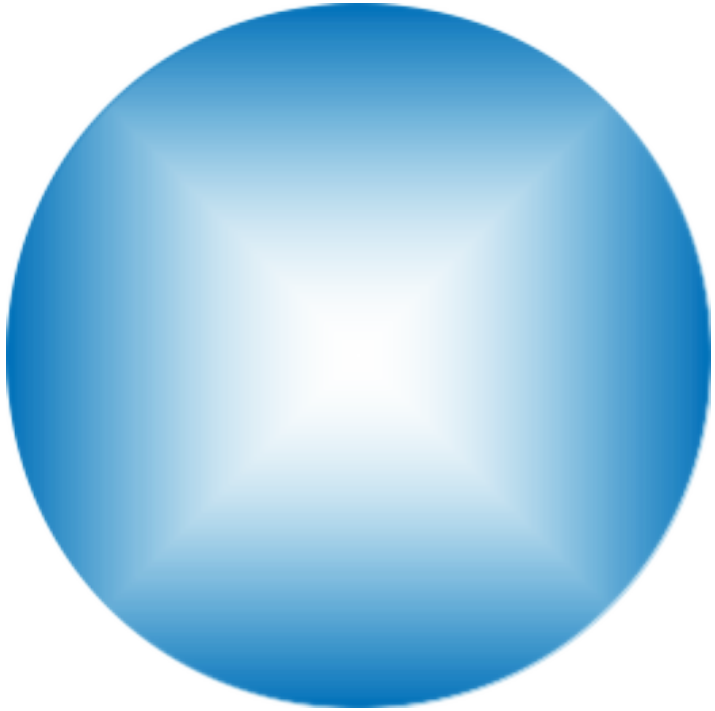
Next in Line...

Computational Process I

Assumptions

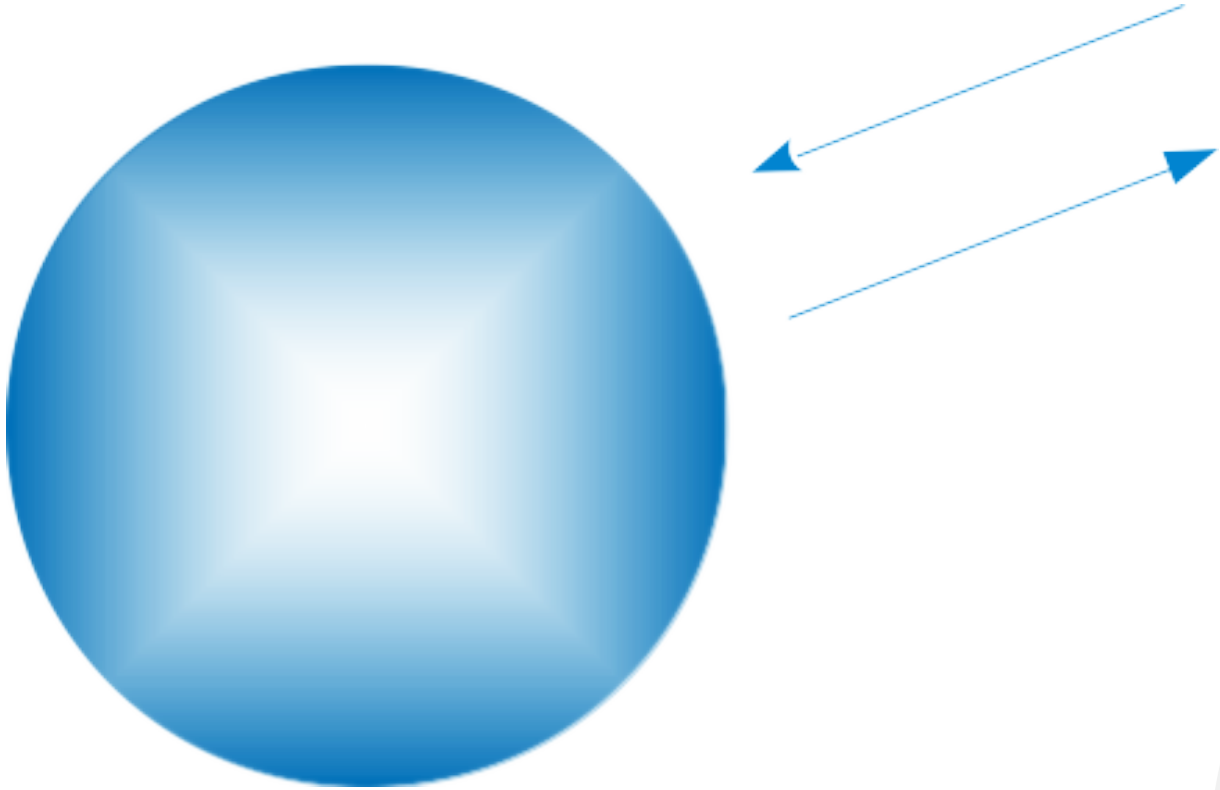
- the *elementary* computational process is sequential
- since it represent the *phenomenal* expression of the dynamics of a computing machine, it has both
  - *input / output*
  - *context*
- as a result, in the following a *computing machine* is the place where a *computational process* occurs

Computational Process II



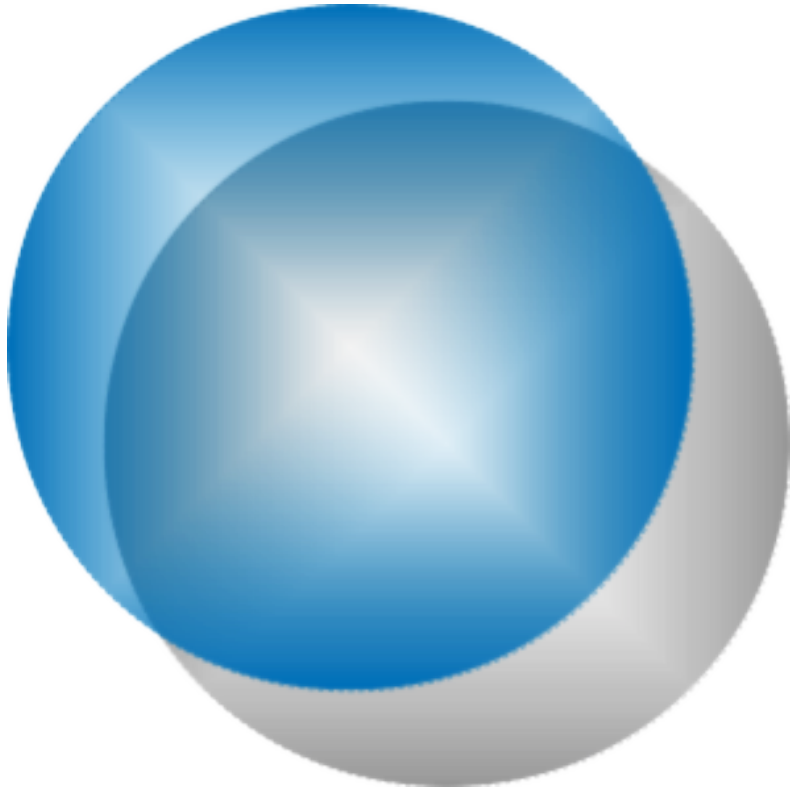
Our representation for a computational process:
sequential computation occur inside the blue circle

Computational Process III



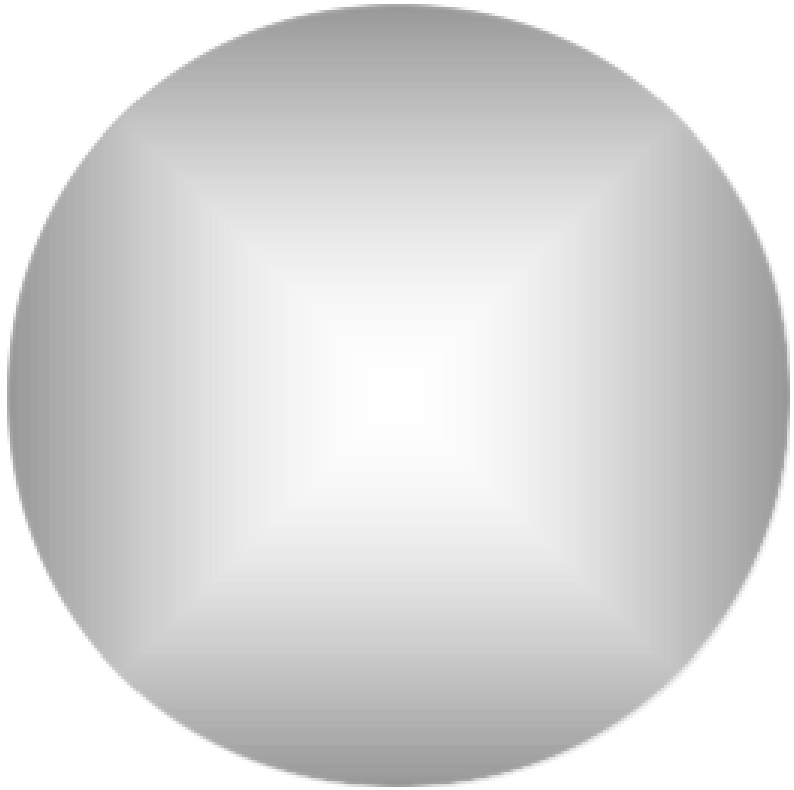
Computational process with *input* and *output*

Computational Process IV



Computational process with *context*

Context for Computation I



Computational *context*

Context for Computation II

What is context when computation is concerned?

- *computing machine*
- resources
- time
- space

Context for Computation III

When do we need to *represent* context for computation?

Whenever either

- computing machine
- resources
- time
- space

or, all of them, are *relevant* to model / represent / understand computation—that is,

- *understanding the dynamics* of the computational process

Context for Computation IV

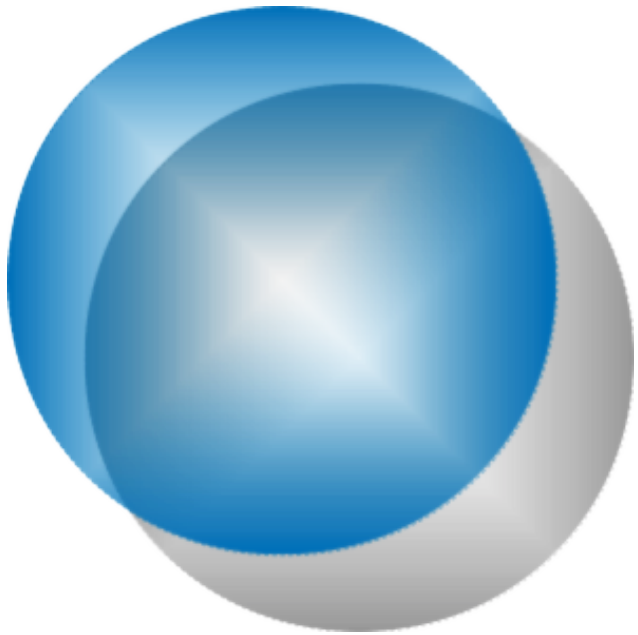
Sorts of computations

- timed computation, whenever the *time* of the computational machine is relevant / essential for the computing process
- spatial computation, whenever the *spatial features* of the computational machine are relevant / essential for the computing process
- more generally, situated computation,[Suchman, 1987] whenever the *environment* of the computational machine is relevant / essential for the computing process
  - where the environment is any meaningful combination of temporal and spatial features with the *resources* required by the computation

Context for Computation V

Representing context for computation

- so, understanding a computational process requires the precise definition of its computational context
- graphically, a computational process (*blue area*) depends on how we define the features of the context (*grey area*)



What is a System?

System

http://www.oxfordlearnersdictionaries.com/definition/english/system

. . . a group of things, pieces of equipment, etc., that are connected or work together . . .

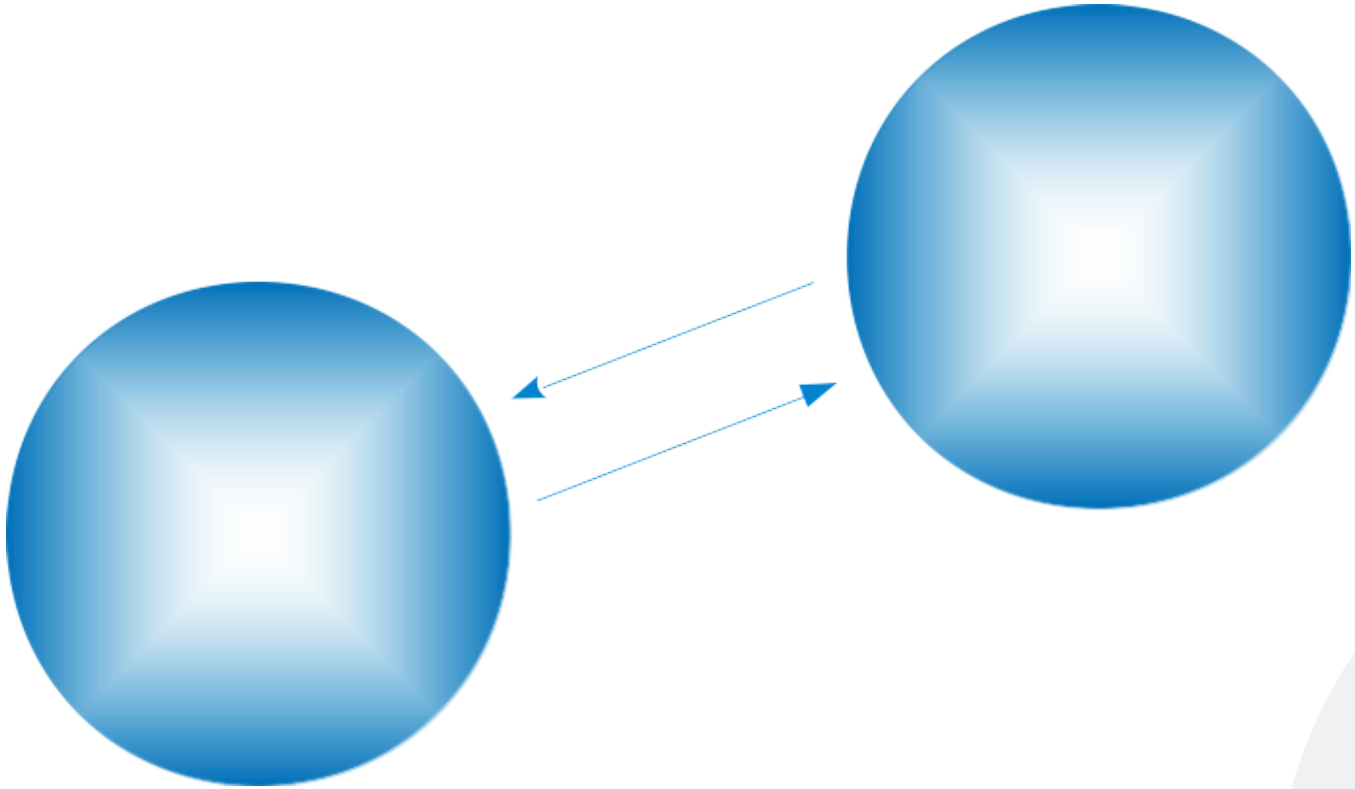
(Interacting) Computational System[Goldin et al., 2006] I

Computational system

In a computational system, two or more computational processes

- *behave* (by computing), and
- *work together* (by interacting)

(Interacting) Computational System[Goldin et al., 2006] II



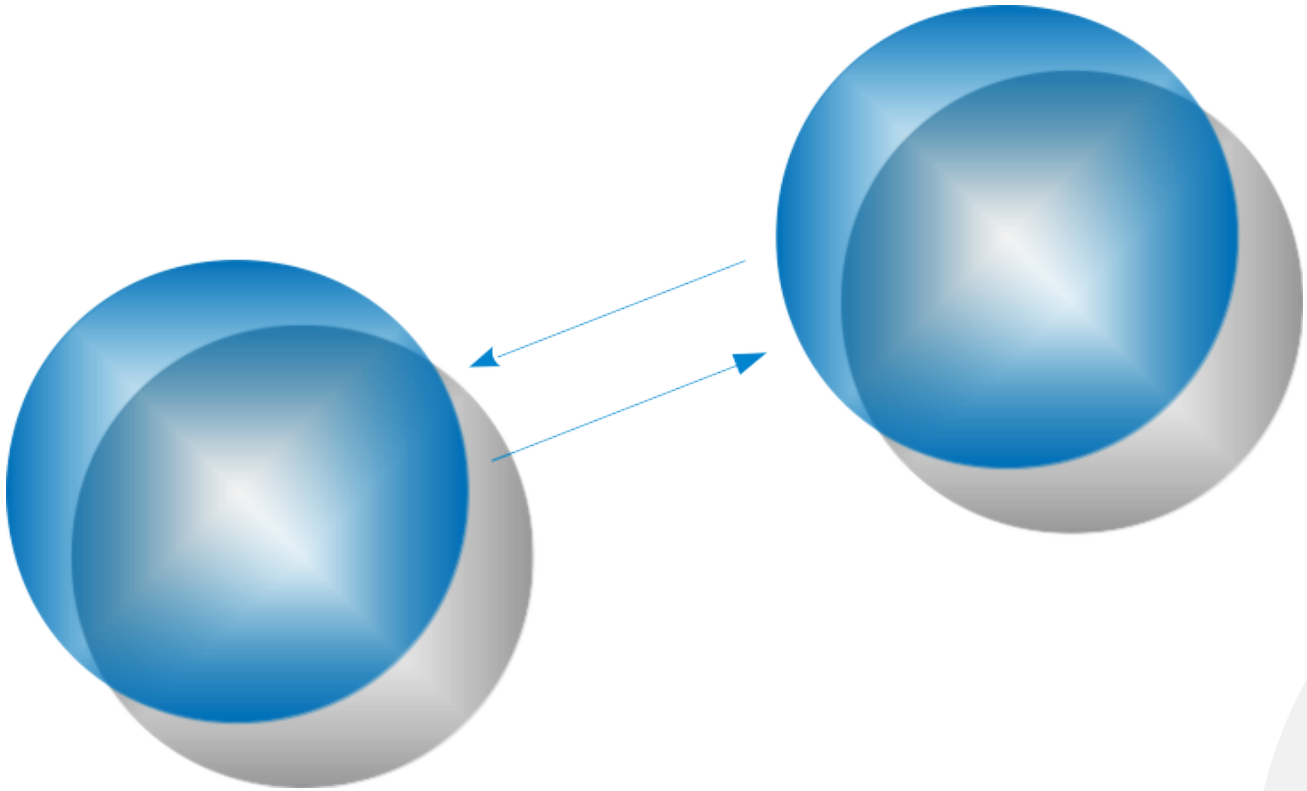
Computational *system*

What is Context for a Computational System? I

How should we represent the context for a computational system?

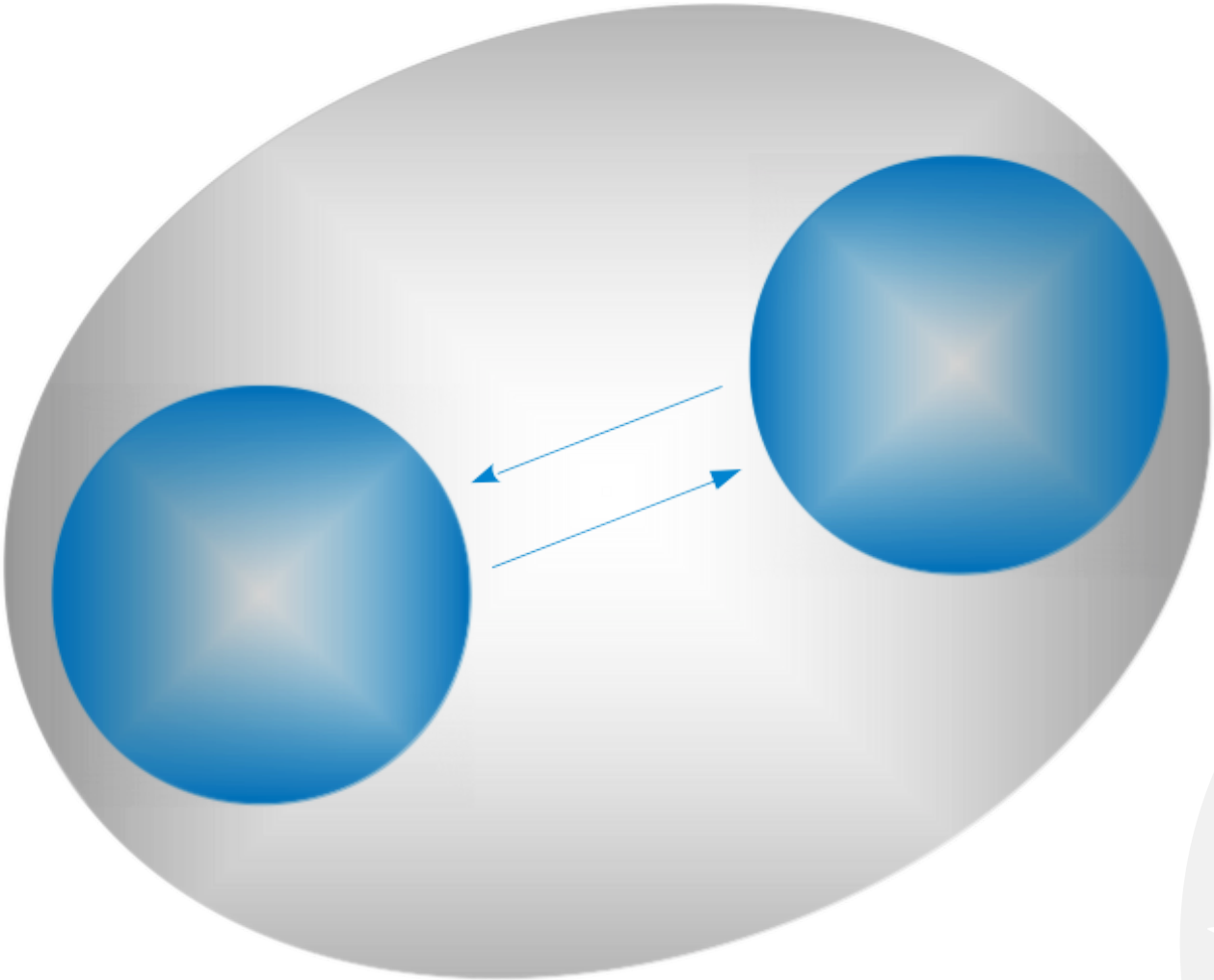
- as one separate, different context for each computational process?
- as a single context for the overall computational system?
- as a combination of the above choices?

What is Context for a Computational System? II



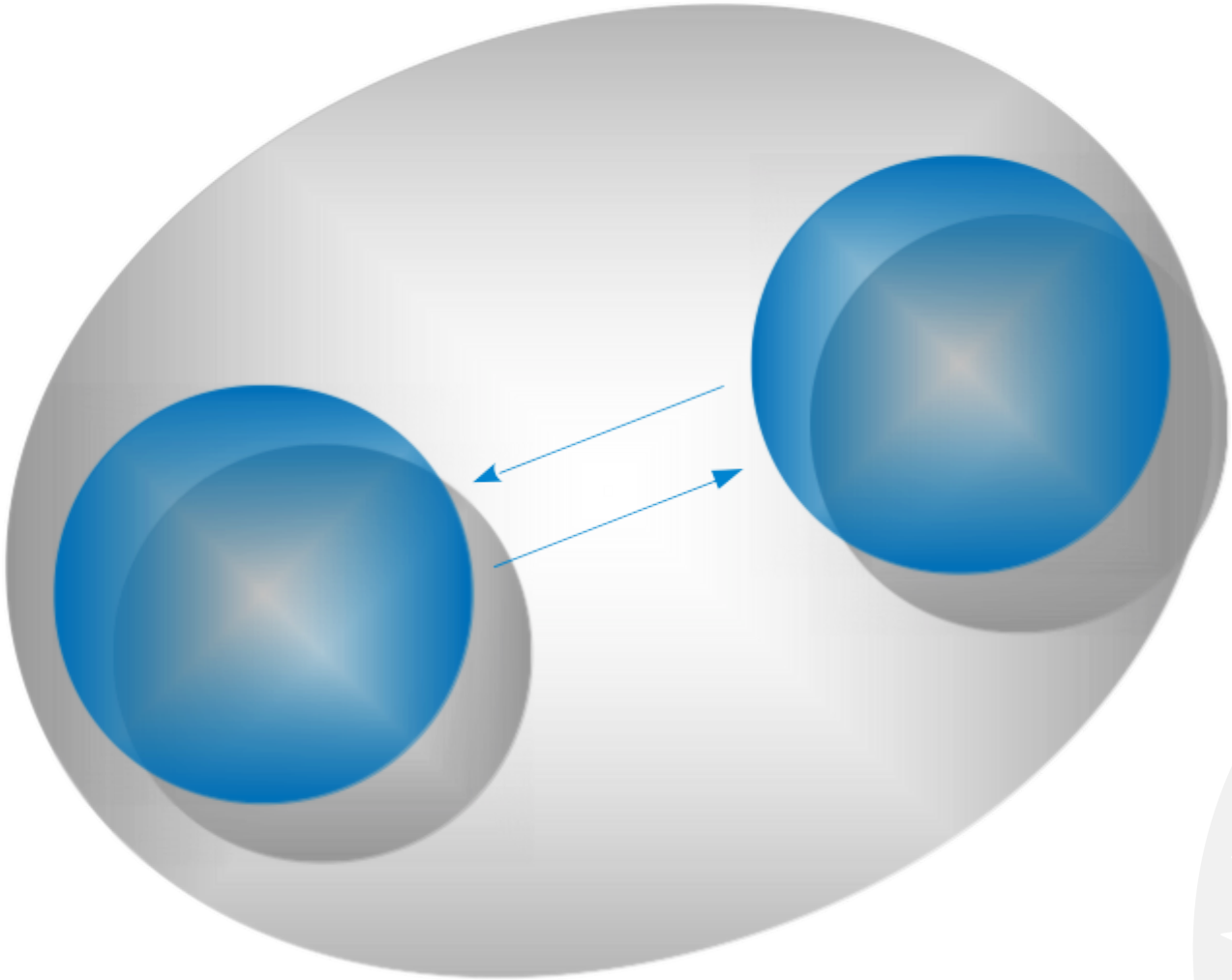
A different context for each process

What is Context for a Computational System? III



One context for all processes

What is Context for a Computational System? IV



A different context for each process plus one context for all processes

What is Context for a Computational System? V

Different contexts, different sorts of systems

The choice of the sort of the context defines the sort of the computational system, such as

- parallel systems
- concurrent systems
- distributed systems

Next in Line...

Parallel vs. Concurrent Systems I

Parallel computing in the literature

- the term is typically used for non-sequential computing processes, where more than one computation can be performed at the same time[Shonkwiler and Lefton, 2006]
- typically requires *multi-core* architectures
- usually exploited to solve *computationally-intensive* scientific / mathematical problems

Parallel vs. Concurrent Systems II

Concurrency in the literature

- the term is typically used with a twofold acceptation[Degano and Montanari, 1987]
  - interleaving, where events occurring in separate concurrent processes could occur in *any* relative *order*—temporal / causal relations between events are not relevant
  - true concurrency, where *partial orderings* are used to explicitly capture temporal / causal relations between events

Parallel vs. Concurrent Systems III

Concurrency vs. parallelism

- relative ordering of events is the main point here
  - in parallel systems, events are *totally* ordered
  - in concurrent systems, events are at most *partially* ordered
- *temporal* relation

→ *temporal* context sets the difference

Distributed Computing & Systems I

Distributed computing in the literature

- the term typically refers to a number of asynchronous computational processes *located* on *different* devices and communicating via message passing (no shared memory)[Kshemkalyani and Singhal, 2011]

  Distributed computing is an activity that is performed on a spatially distributed system[Lamport and Lynch, 1990]

Distributed systems in the literature

- the term typically refer to a collection of devices working together through a network connection

  A distributed system is a collection of independent computers that appears to its users as a single coherent system[Tanenbaum and van Steen, 2017]

Distributed Computing & Systems II

Distribution

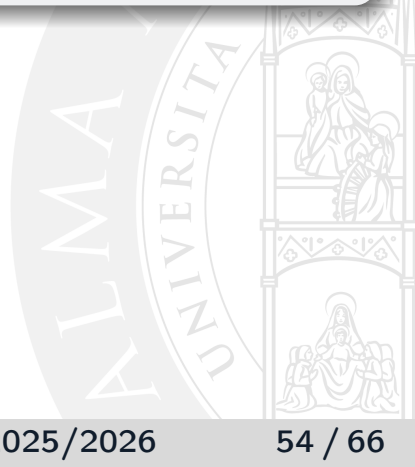
- *physical distribution* of computational processes and computing devices is the main point here
  - in distributed computing, the focus is on the spatial distribution of processes
  - in distributed systems, the focus is on the spatial distribution of devices
- *spatial* relation

→ *spatial* context defines both

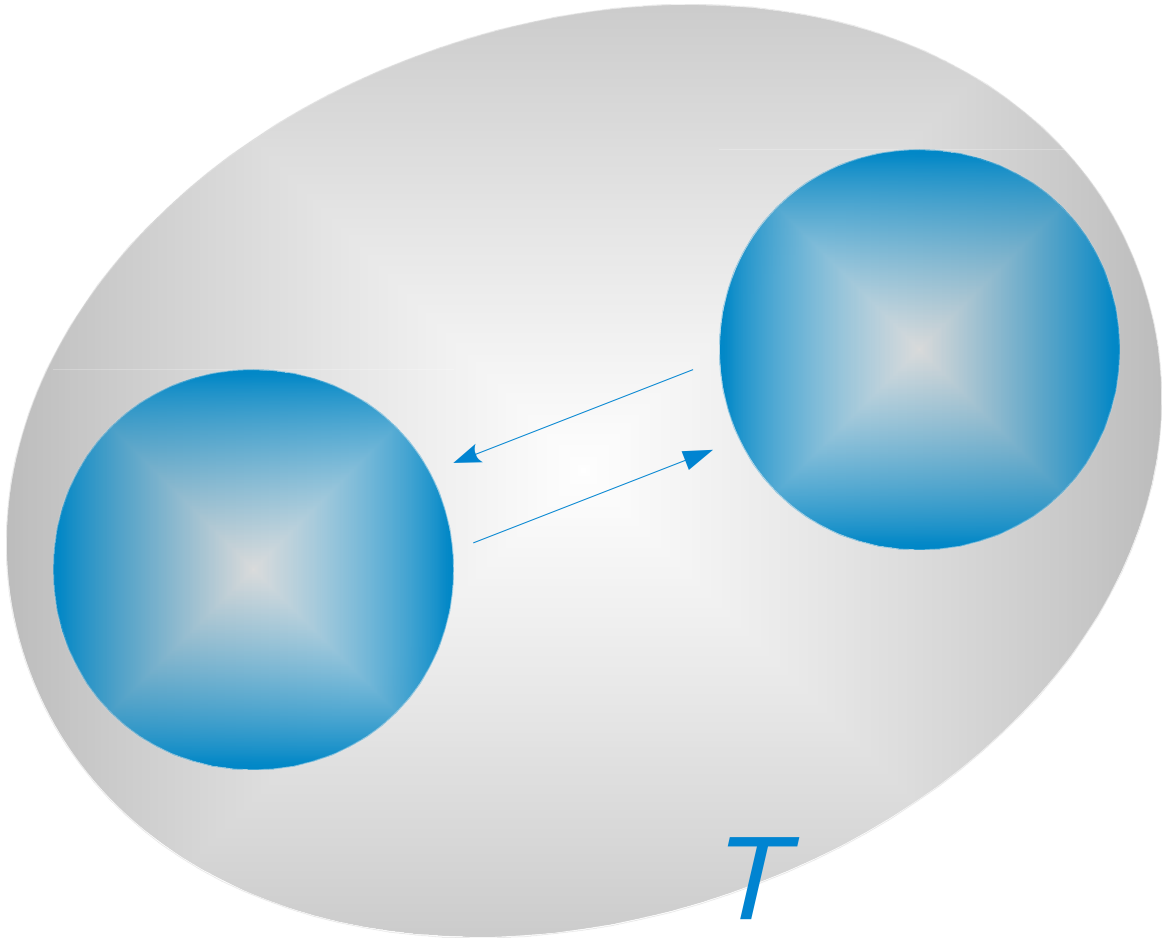
Definitions I

Parallel computing & systems

- given a computational system, we talk of parallel computation whenever the *temporal* context is the same for all computational process
- a parallel system is a computational system performing parallel computations



Definitions II

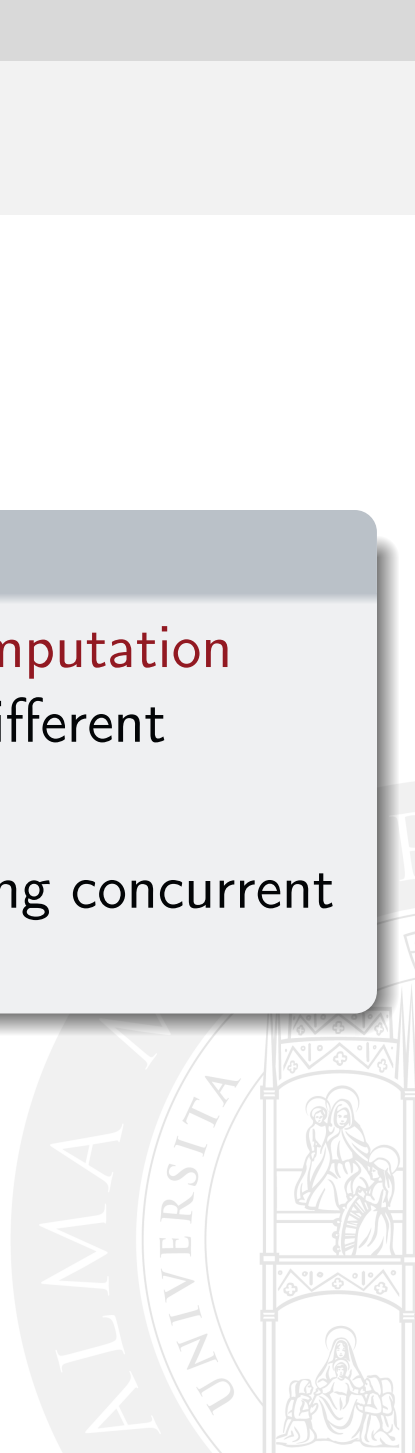


Parallel computing: the same temporal context T for all processes

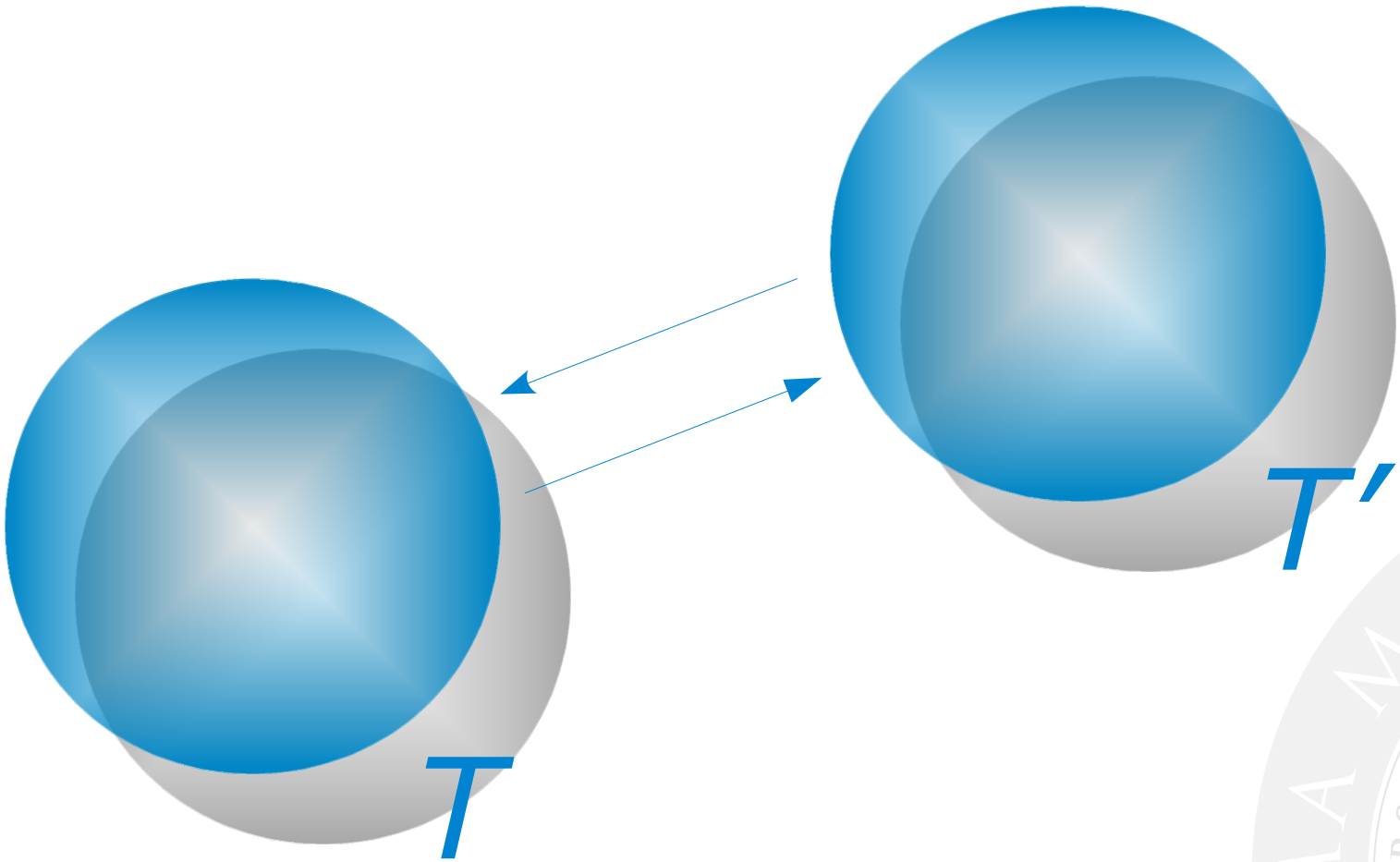
Definitions III

Concurrent computing & systems

- given a computational system, we talk of concurrent computation whenever at least two computational processes have a different *temporal* context
- a concurrent system is a computational system performing concurrent computations



Definitions IV

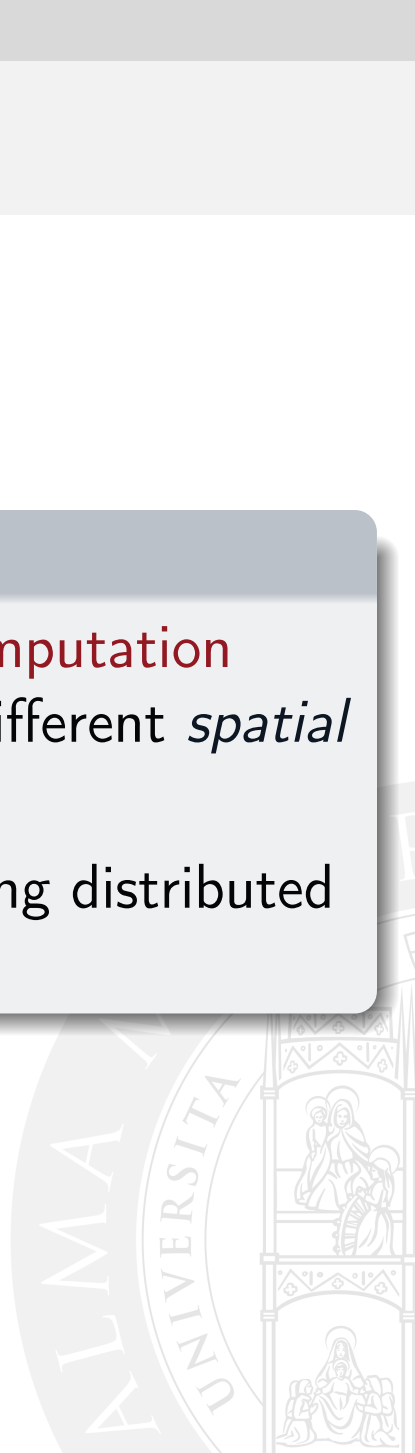


Concurrent computing: different temporal contexts $T \neq T'$ for different processes

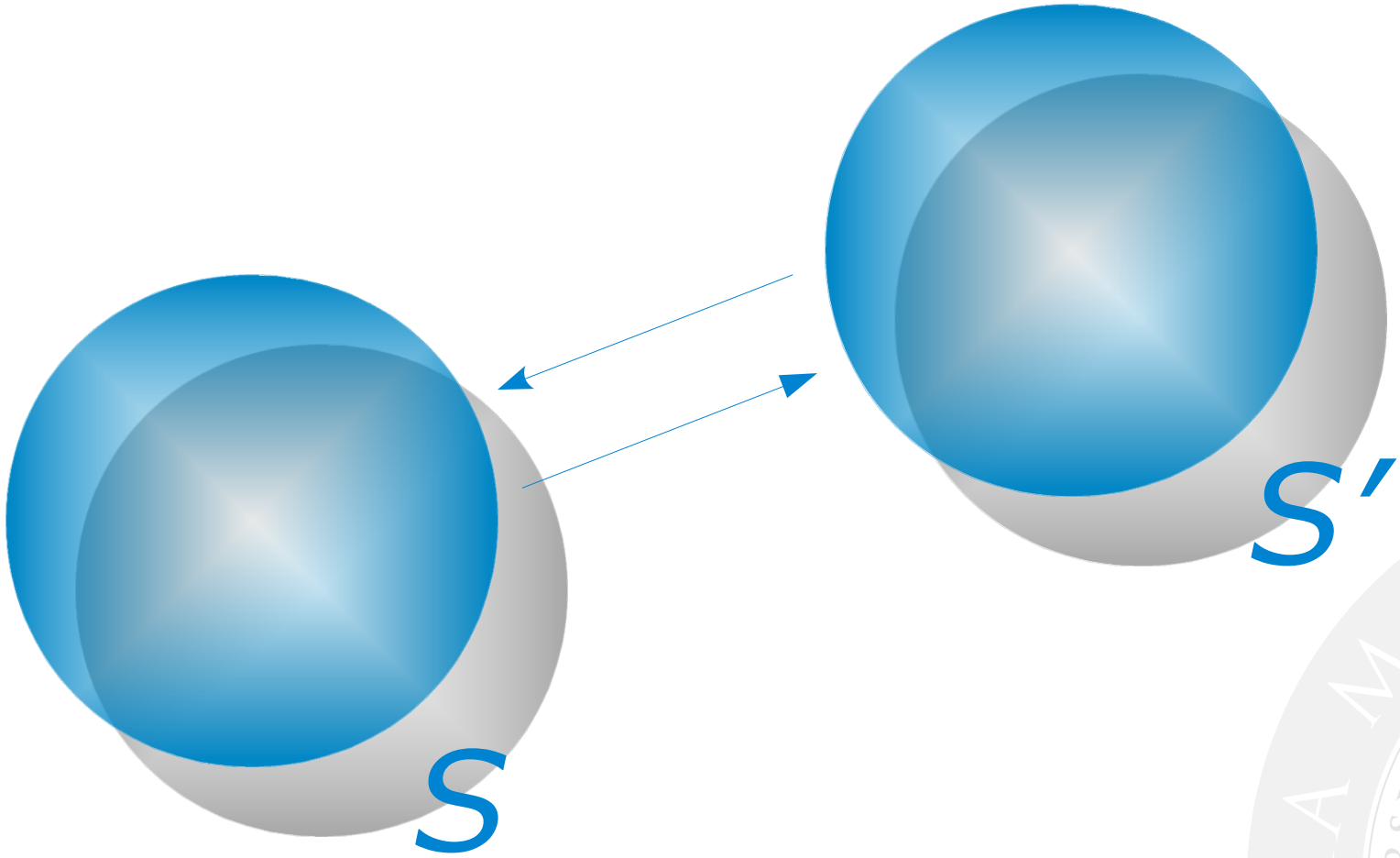
Definitions V

Distributed computing & systems

- given a computational system, we talk of distributed computation whenever at least two computational processes have a different *spatial* context
- a distributed system is a computational system performing distributed computations

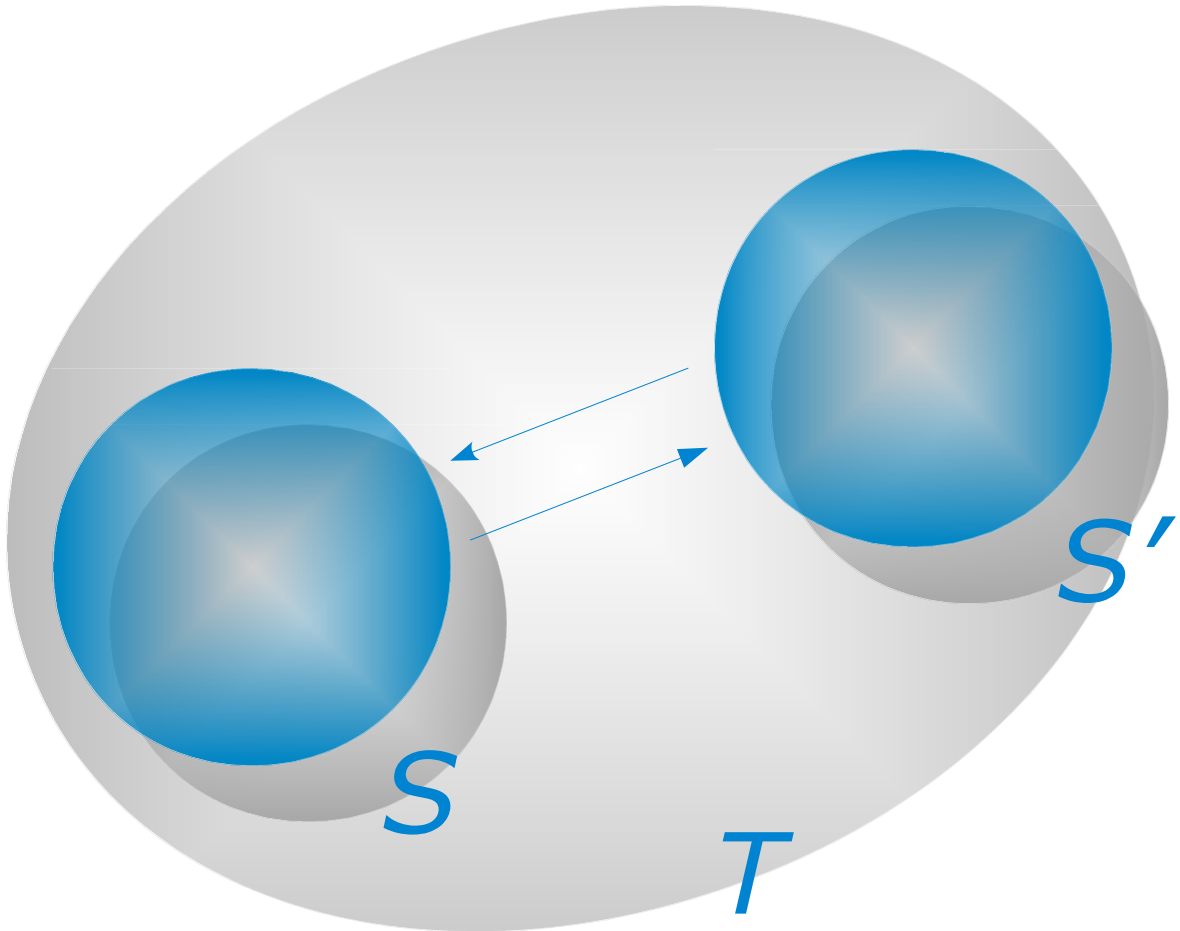


Definitions VI



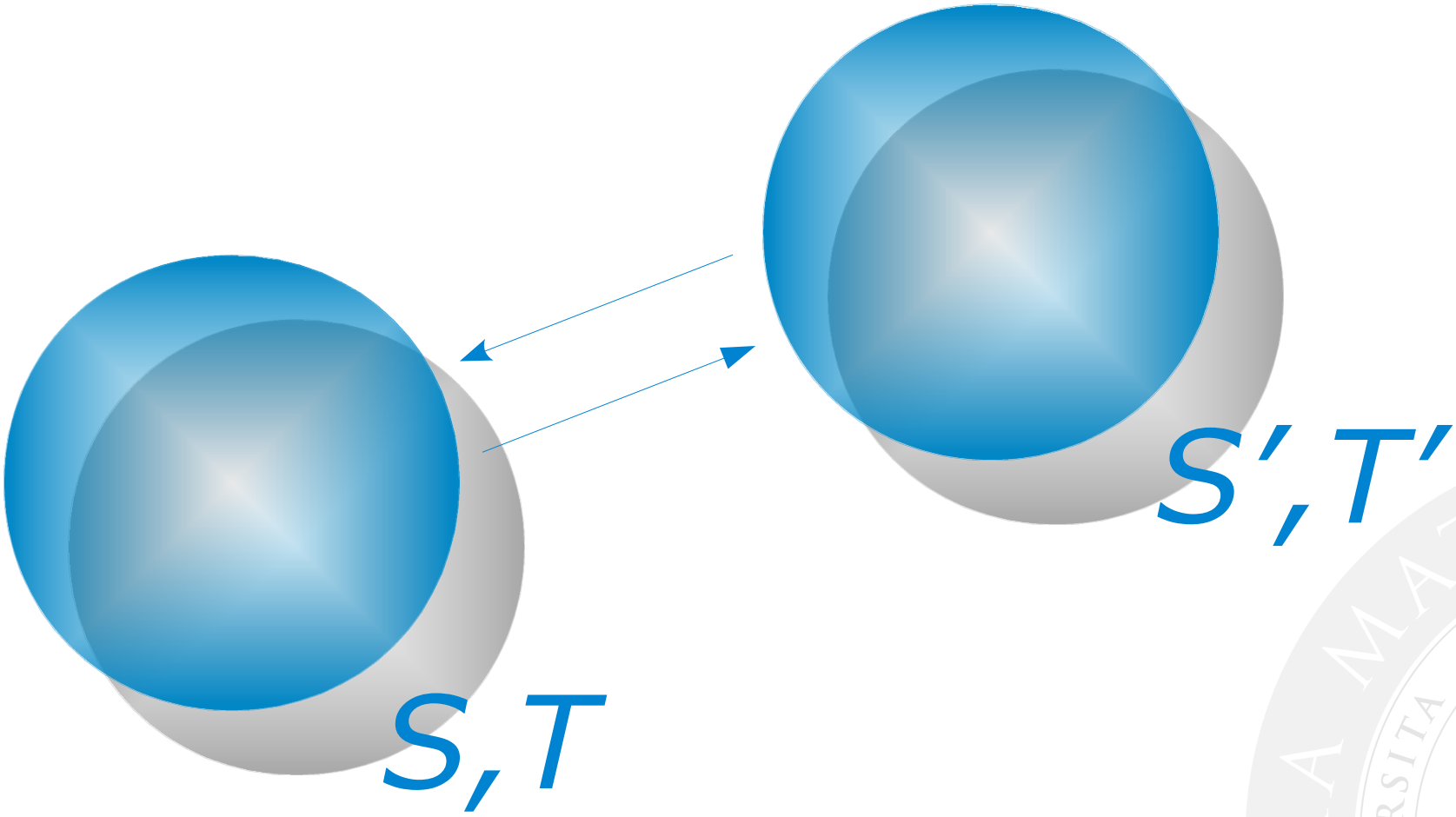
Distributed computing: different spatial contexts $S \neq S'$ for different processes

Spatial vs. Temporal Contexts I



Distributed parallel computing: $S \neq S'$, same T

Spatial vs. Temporal Contexts II



Distributed concurrent computing: $S \neq S'$, $T \neq T'$

Next in Line...

Summing Up I

Foundations

- there is a basic needs to devise out a solid and shared foundation for the field of *computer science*
- first of all, by understanding what science is in general
  - phenomena & noumena
  - explanation & prediction
- then, by understanding and defining the most abstract and comprehensive notion of *computation*

Summing Up II

Basic ontology

We instrument a simple notion of

- computational process & device
- context
- computational system

for our course

Summing Up III

Basic definitions

We also provide a reference definition of

- parallel computation & system
- concurrent computation & system
- distributed computation & system

to navigate literature fruitfully, against all odds

Roots of Distributed Systems
Computation in Space & Time

Distributed Systems

Andrea Omicini
andrea.omicini@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna

Academic Year 2025/2026

References I

[Brewer, 2000] Brewer, E. A. (2000).
Towards robust distributed systems (abstract).
In *19th Annual ACM Symposium on Principles of Distributed Computing (PODC '00)*, page 7, New York, New York, USA. ACM Press
DOI:10.1145/343477.343502

[Burks et al., 1982] Burks, A. W., Goldstine, H. H., and von Neumann, J. (1982).
Preliminary discussion of the logical design of an electronic computing instrument.
In Randell, B., editor, *The Origins of Digital Computers*, Texts and Monographs in Computer Science, pages 399–413. Springer Berlin Heidelberg, Berlin, Heidelberg
DOI:10.1007/978-3-642-61812-3_32

[Conery, 2010a] Conery, J. S. (2010a).
Explorations in Computing: An Introduction to Computer Science.
Chapman & Hall/CRC Textbooks in Computing. CRC Press, 1st edition
https://www.crcpress.com/Explorations-in-Computing-An-Introduction-to-Computer-Science/Conery/p/book/9781439812624

[Conery, 2010b] Conery, J. S. (2010b).
Ubiquity symposium 'What is computation?': Computation is symbol manipulation.
Ubiquity, 2010(November):4:1–4:7
DOI:10.1145/1880066.1889839

[Degano and Montanari, 1987] Degano, P. and Montanari, U. (1987).
Concurrent histories: A basis for observing distributed systems.
Journal of Computer and System Sciences, 34(2-3):422–461
DOI:10.1016/0022-0000(87)90032-8

References II

[Denning, 2008] Denning, P. J. (2008).
The computing field: Structure.
In Wah, B., editor, *Wiley Encyclopedia of Computer Science and Engineering*, pages 615–623. John Wiley & Sons, Inc., Hoboken, NJ, USA
DOI:10.1002/9780470050118.ecse962

[Denning, 2010] Denning, P. J. (2010).
Ubiquity symposium 'What is computation?': Opening statement.
Ubiquity, 2010(November):1:1–1:11
DOI:10.1145/1880066.1880067

[Denning, 2011] Denning, P. J. (2011).
Ubiquity symposium 'What is computation?': What have we said about computation?
Ubiquity, 2011(April):1:1–1:7
DOI:10.1145/1967045.1967046

[Fich and Ruppert, 2003] Fich, F. and Ruppert, E. (2003).
Hundreds of impossibility results for distributed computing.
Distributed Computing, 16(2-3):121–163
DOI:10.1007/s00446-003-0091-y

[Frailey, 2010] Frailey, D. J. (2010).
Ubiquity symposium 'What is computation?': Computation is process.
Ubiquity, 2010(November):5:1–5:6
DOI:10.1145/1880066.1891341

References III

[Freeman, 2011] Freeman, P. A. (2011).
Ubiquity symposium 'What is computation?': Is the symposium question harmful?: Consideration of the question 'What is computation?' considered harmful.
Ubiquity, 2011(March):1:1–1:4
DOI:10.1145/1959016.1959018

[Gilbert and Lynch, 2002] Gilbert, S. and Lynch, N. (2002).
Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services.
ACM SIGACT News, 33(2):51
DOI:10.1145/564585.564601

[Goldin et al., 2006] Goldin, D. Q., Smolka, S. A., and Wegner, P., editors (2006).
Interactive Computation: The New Paradigm.
Springer Berlin Heidelberg
DOI:10.1007/3-540-34874-3

[Kshemkalyani and Singhal, 2011] Kshemkalyani, A. D. and Singhal, M. (2011).
Distributed Computing: Principles, Algorithms, and Systems.
Cambridge University Press
https://www.cambridge.org/us/universitypress/subjects/engineering/communications-and-signal-processing/distributed-computing-principles-algorithms-and-systems

[Lamport and Lynch, 1990] Lamport, L. and Lynch, N. (1990).
Distributed computing: Models and methods.
In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science, Volume B – Formal Models and Semantics*, chapter 18, pages 1157–1199. Elsevier
DOI:10.1016/B978-0-444-88074-1.50023-8

References IV

[Newell et al., 1967] Newell, A., Perlis, A. J., and Simon, H. A. (1967).
Computer science.
Science, 157(3795):1373–1374
DOI:10.1126/science.157.3795.1373-b

[Shonkwiler and Lefton, 2006] Shonkwiler, R. W. and Lefton, L. (2006).
An Introduction to Parallel and Vector Scientific Computation.
Cambridge University Press
http://www.cambridge.org/us/academic/subjects/computer-science/scientific-computing-scientific-software/introduction-parallel-and-vector-scientific-computation

[Suchman, 1987] Suchman, L. A. (1987).
Plans and Situated Actions: The Problem of Human-Machine Communication.
Cambridge University Press, New York, NYU, USA

[Tanenbaum and van Steen, 2017] Tanenbaum, A. S. and van Steen, M. (2017).
Distributed Systems. Principles and Paradigms.
Pearson Prentice Hall, 3rd edition
https://www.distributed-systems.net/index.php/books/ds3/

[Turing, 1937] Turing, A. M. (1937).
On computable numbers, with an application to the entscheidungsproblem.
Proceedings of the London Mathematical Society, s2-42(1):230–265
DOI:10.1112/plms/s2-42.1.230