Extending the Wait-free Hierarchy to Multi-Threaded Systems

Matthieu Perrin Achour Mostefaoui Grégoire Bonin

LS2N, Université de Nantes

PODC 2020

ACM Symposium on Principles of Distributed Computing

Virtual Conference

August 3-6, 2020

The Wait-free Hierarchy

Consensus number

- Consensus number $k \in \mathbb{N}$:
  - Wait-free universel for k processes
  - Not wait-free universel for $k + 1$ processes
- Consensus number ∞:
  - Wait-free universel for k processes, for all k

Significance

- Objects with CN x cannot implement objects with CN $y > x$

The Wait-free Hierarchy in Multi-Threaded Systems

Multi-threaded systems

- Threads can be created dynamically
  - No bound on the number of threads in an execution
- Allocation of unbounded but finite arrays
  - How to allocate one shared register to each thread?

The iterator stack:

- Infinite consensus number
- Not universal in multi-threaded systems

Problem statement

How to compare the synchronization power of shared objects in multi-threaded systems?

The Wait-free Hierarchy in Multi-Threaded Systems

Multi-threaded systems

- Threads can be created dynamically
  - No bound on the number of threads in an execution
- Allocation of unbounded but finite arrays
  - How to allocate one shared register to each thread?

The iterator stack:

- Infinite consensus number
- Not universal in multi-threaded systems

Problem statement

How to compare the synchronization power of shared objects in multi-threaded systems?

The Wait-free Hierarchy in Multi-Threaded Systems

Multi-threaded systems

- Threads can be created dynamically
  - No bound on the number of threads in an execution
- Allocation of unbounded but finite arrays
  - How to allocate one shared register to each thread?

The iterator stack:

- Infinite consensus number
- Not universal in multi-threaded systems

Problem statement

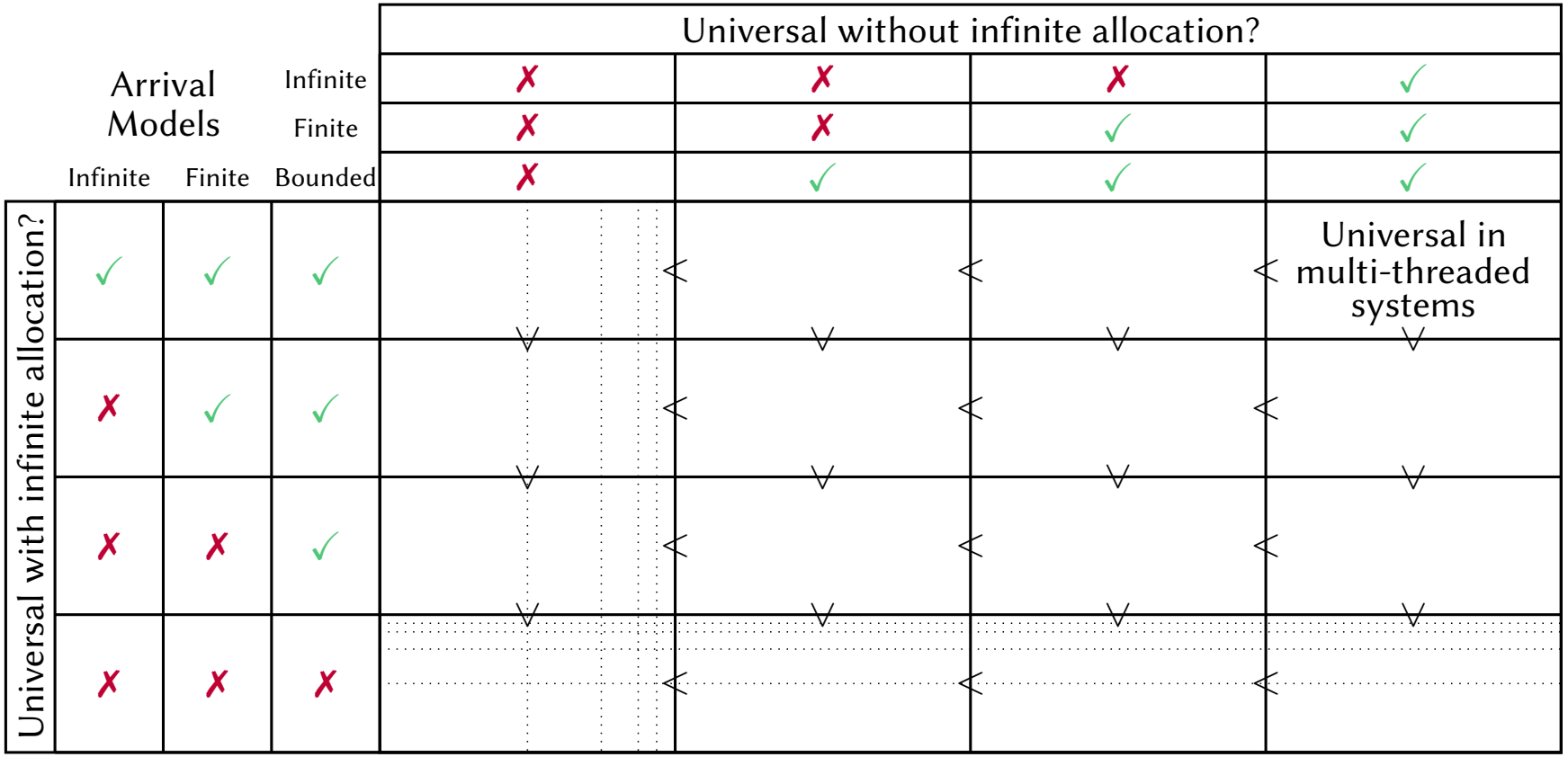
How to compare the synchronization power of shared objects in multi-threaded systems?

Arrival Models

Maximal number of processes in an execution:

- M_1^n Classical model
  - At most n processes (known to the developper)
- M_1 Bounded arrival model
  - The bound is known at initialization
- M_2 Finite arrival model
  - After some time, no new thread is started
- M_3 Infinite arrival model
  - New threads may keep arriving

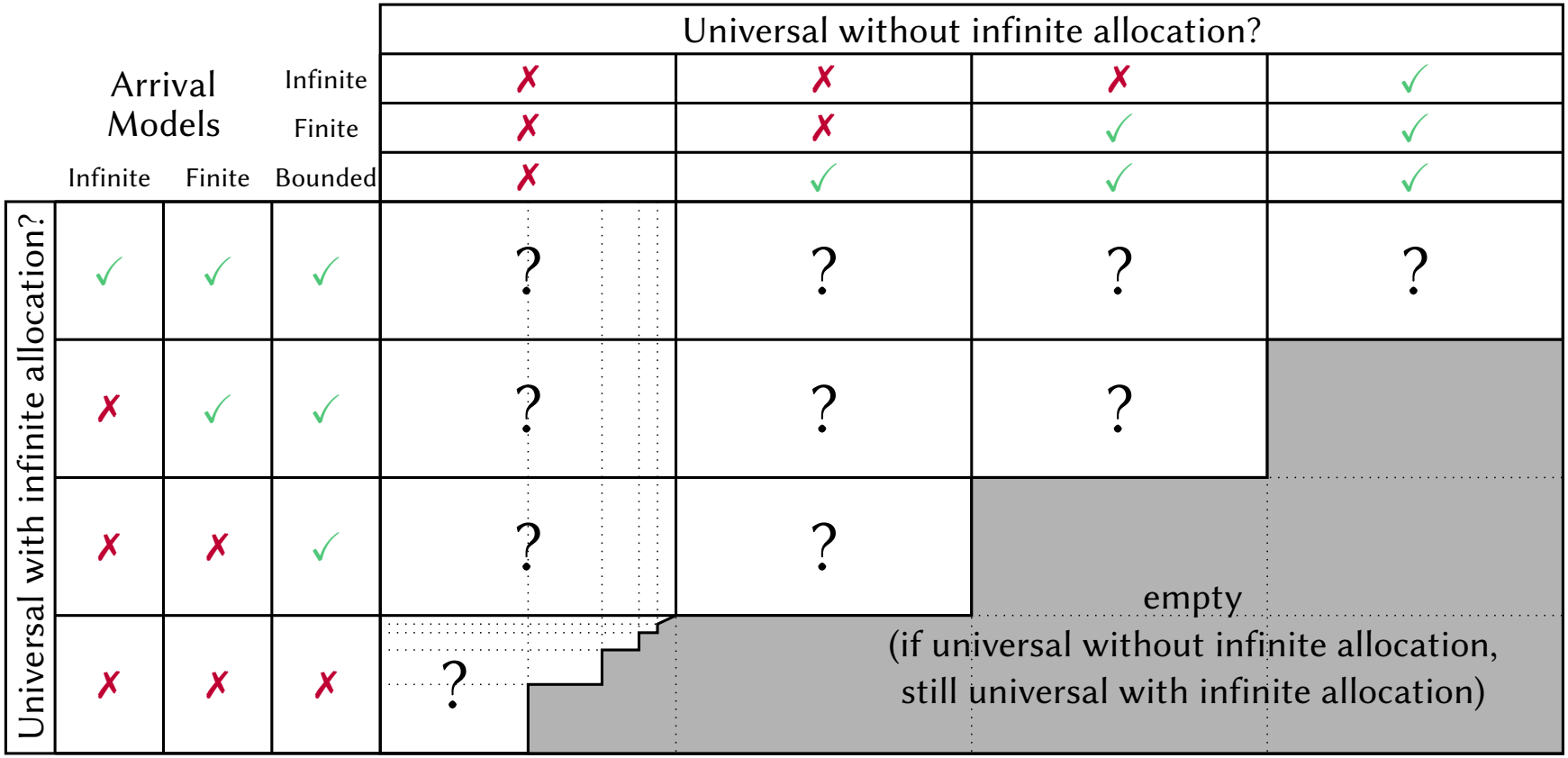
Extended Wait-free Hierarchy



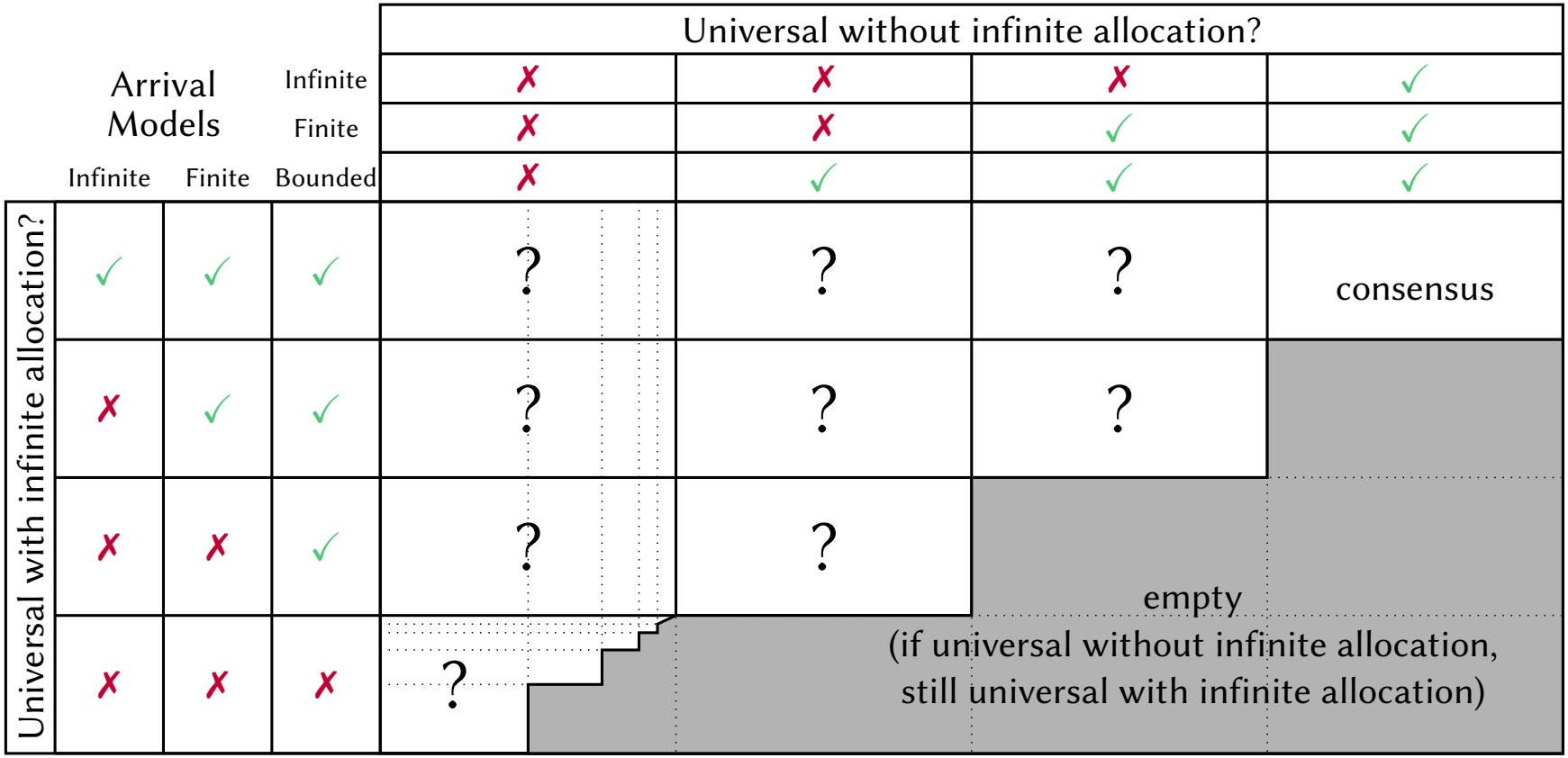
Filling the Hierarchy

Universal with infinite allocation?	Arrival Models: Infinite	Arrival Models: Finite	Arrival Models: Bounded	Universal without infinite allocation?			
			Infinite	✗	✗	✗	✓
			Finite	✗	✗	✓	✓
			Bounded	✗	✓	✓	✓
?	✓	✓	✓	?	?	?	?
✗	✗	✓	✓	?	?	?	?
✗	✗	✗	✓	?	?	?	?
✗	✗	✗	✗	?	?	?	?

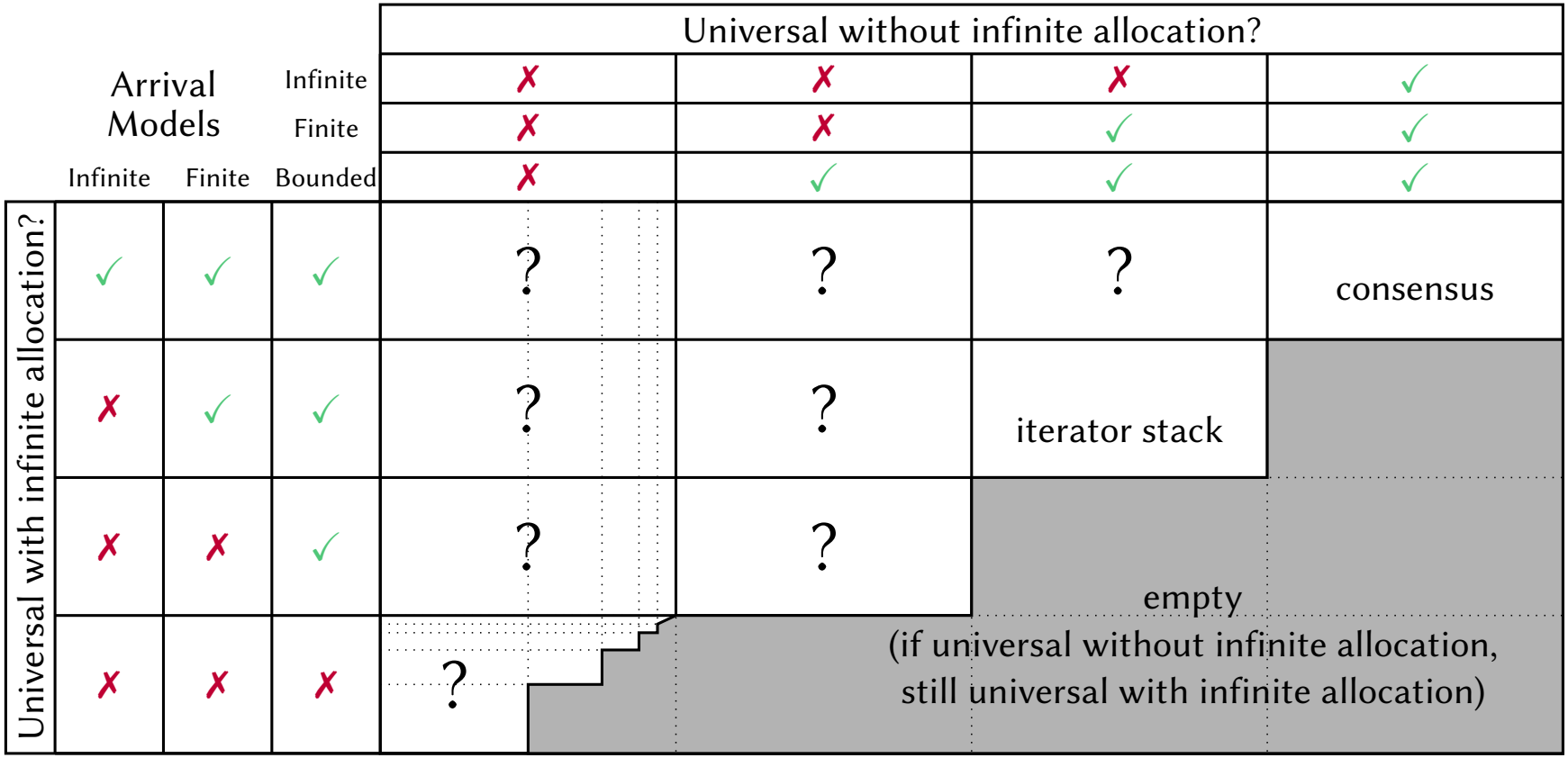
Filling the Hierarchy



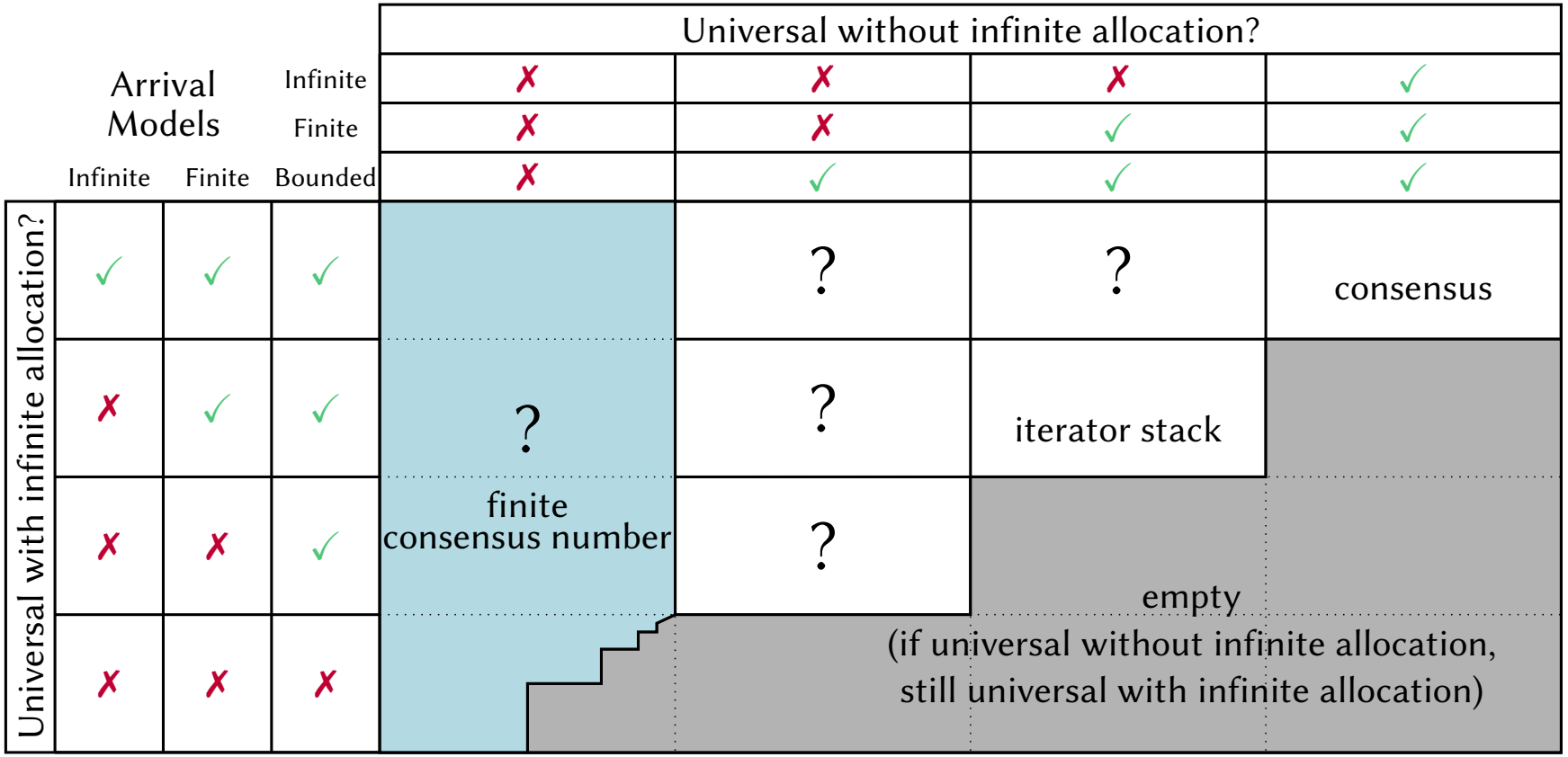
Filling the Hierarchy



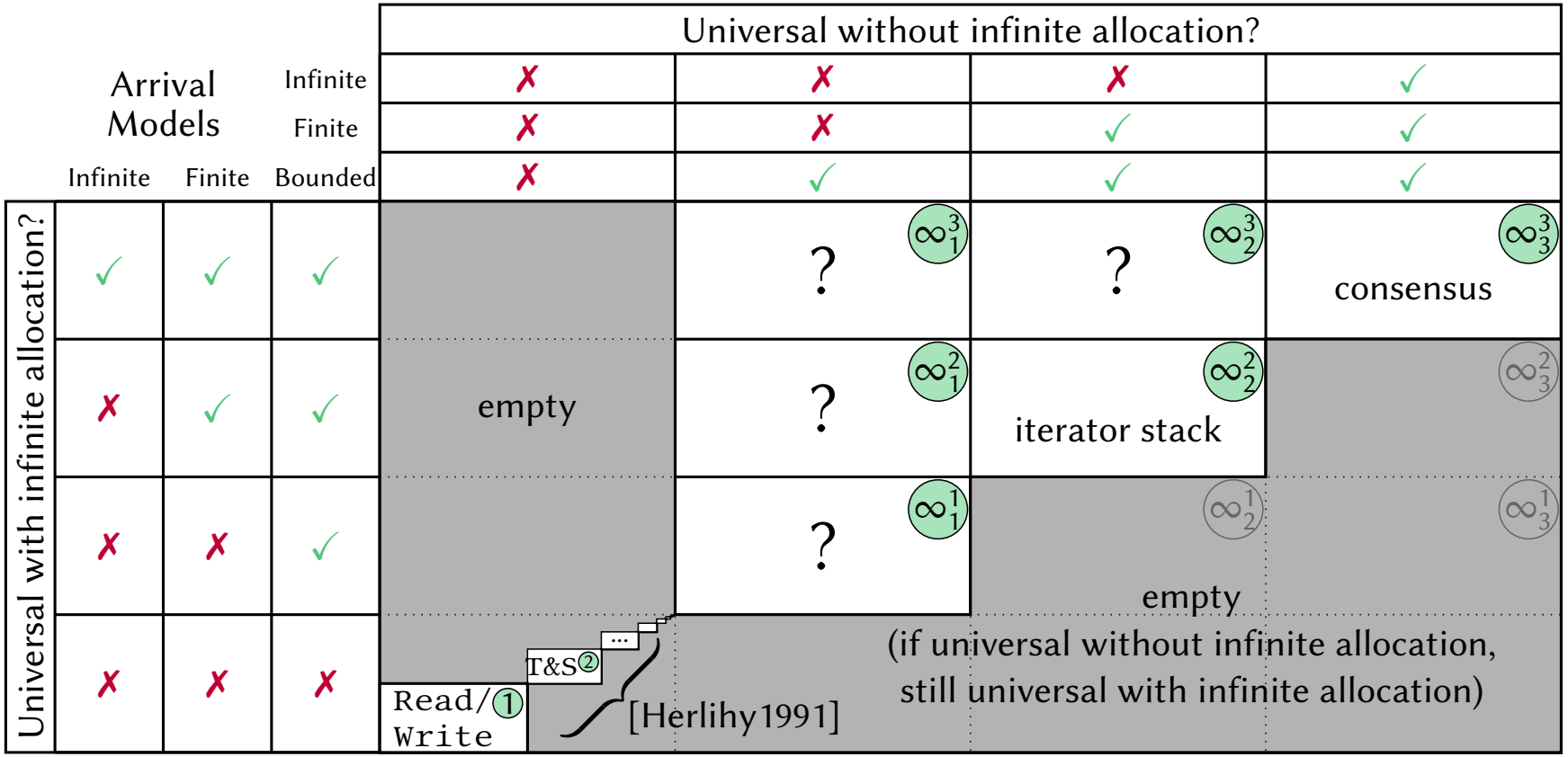
Filling the Hierarchy



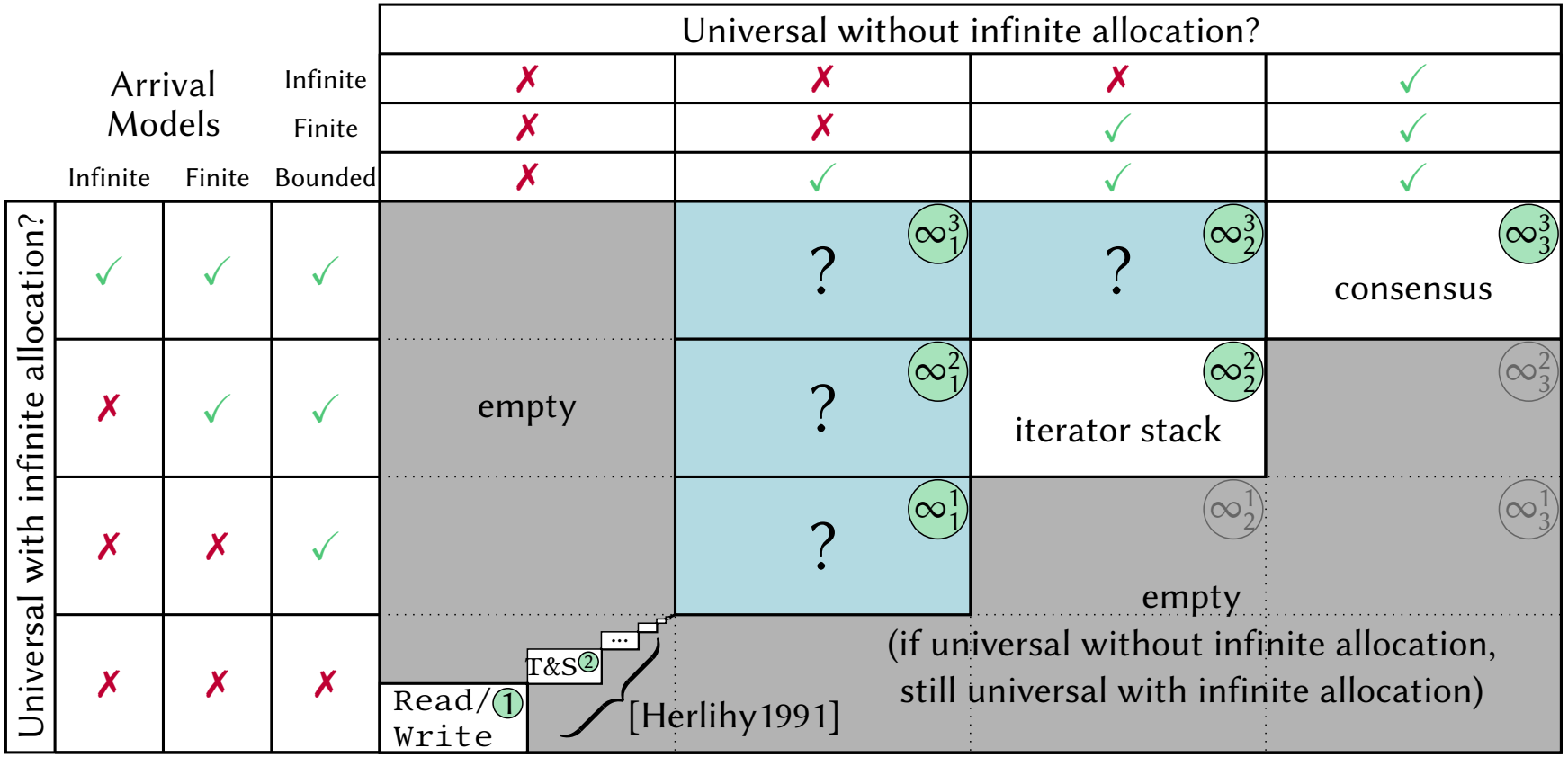
Filling the Hierarchy



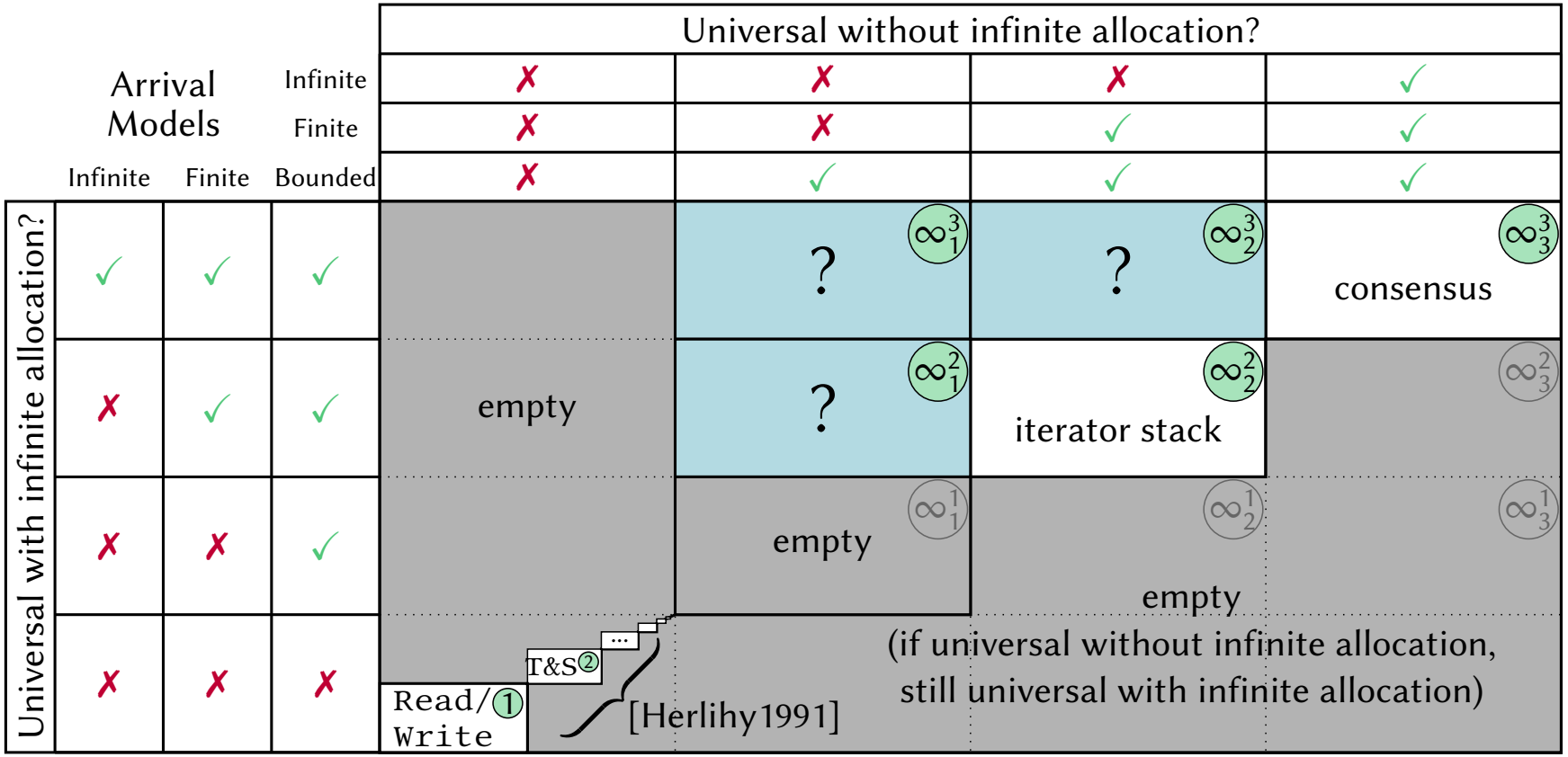
Filling the Hierarchy



Filling the Hierarchy



Filling the Hierarchy



Filling the Hierarchy