

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?

Maximal number of processes in an execution:

M_1^n Classical model

- ▶ At most n processes (known to the developer)

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

			Universal without infinite allocation?			
Arrival Models	Infinite	Infinite	X	X	X	✓
	Finite	Finite	X	X	✓	✓
	Infinite	Finite	X	✓	✓	✓
Universal with infinite allocation?	✓	✓	✓	⋮	⋮	⋮
	X	✓	✓	⋮	⋮	⋮
	X	X	✓	⋮	⋮	⋮
	X	X	X	⋮	⋮	⋮

Universal in multi-threaded systems

Filling the Hierarchy

			Universal without infinite allocation?				
Arrival Models	Infinite	Finite	X	X	X	✓	
	Infinite	Finite	X	X	✓	✓	
	Infinite	Finite	X	✓	✓	✓	
Universal with infinite allocation?	✓	✓	✓	?	?	?	?
	X	✓	✓	?	?	?	?
	X	X	✓	?	?	?	?
	X	X	X	?	?	?	?

Filling the Hierarchy

			Universal without infinite allocation?				
			Infinite	X	X	X	✓
			Finite	X	X	✓	✓
			Bounded	X	✓	✓	✓
Universal with infinite allocation?	✓	✓	✓	?	?	?	?
	X	✓	✓	?	?	?	
	X	X	✓	?	?		empty
	X	X	X	?			(if universal without infinite allocation, still universal with infinite allocation)

Filling the Hierarchy

			Universal without infinite allocation?				
			Infinite	\times	\times	\times	✓
			Finite	\times	\times	✓	✓
			Bounded	\times	✓	✓	✓
Arrival Models							
Universal with infinite allocation?				?	?	?	consensus
				?	?	?	
				?	?	empty	
				?		(if universal without infinite allocation, still universal with infinite allocation)	

Filling the Hierarchy

			Universal without infinite allocation?				
			Infinite	✗	✗	✗	✓
			Finite	✗	✗	✓	✓
			Bounded	✗	✓	✓	✓
Arrival Models							
Universal with infinite allocation?	Infinite	Finite	Bounded	✓	✓	✓	?
	✓	✓	✓	?	?	?	consensus
	✗	✓	✓	?	?		iterator stack
	✗	✗	✓	?	?		empty
	✗	✗	✗	?			(if universal without infinite allocation, still universal with infinite allocation)

Filling the Hierarchy

			Universal without infinite allocation?				
			Infinite	✓	✗	✗	✓
			Finite	✗	✗	✓	✓
			Bounded	✗	✓	✓	✓
Arrival Models							
Universal with infinite allocation?			✓	✓	✓	?	?
			✗	✓	✓	?	iterator stack
			✗	✗	✓	?	empty
			✗	✗	✗	(if universal without infinite allocation, still universal with infinite allocation)	

Filling the Hierarchy

			Universal without infinite allocation?				
			Infinite	✓	✗	✗	✓
			Finite	✗	✗	✓	✓
			Bounded	✗	✓	✓	✓
Arrival Models							
Infinite				✓	✗	✗	✓
Finite				✗	✗	✓	✓
Bounded				✗	✓	✓	✓
Universal with infinite allocation?							
✓	✓	✓		?	∞ ³ ₁	?	∞ ³ ₂
✗	✓	✓	empty	?	∞ ² ₁	∞ ² ₂	∞ ² ₃
✗	✗	✓		?	∞ ¹ ₁	∞ ¹ ₂	∞ ¹ ₃
✗	✗	✗					
Read/ Write			T&S ...	[Herlihy1991]	(if universal without infinite allocation, still universal with infinite allocation)		

Filling the Hierarchy

			Universal without infinite allocation?				
			Infinite	✓	✗	✗	✓
			Finite	✗	✗	✓	✓
			Bounded	✗	✓	✓	✓
Arrival Models							
Infinite				✓	✗	✗	✓
Finite				✗	✗	✓	✓
Bounded				✗	✓	✓	✓
Universal with infinite allocation?				?	∞ ³ ₁	?	∞ ³ ₂
			empty	?	∞ ² ₁	?	∞ ² ₂
				?	∞ ¹ ₁	?	∞ ¹ ₂
						empty	∞ ¹ ₃
							if universal without infinite allocation, still universal with infinite allocation)
Read/ Write			T&S ...	[Herlihy 1991]			

Filling the Hierarchy

			Universal without infinite allocation?				
			Infinite	X	X	X	✓
			Finite	X	X	✓	✓
			Bounded	X	✓	✓	✓
Arrival Models							
Universal with infinite allocation?							
✓	✓	✓		?	∞_1^3	?	∞_3^3
X	✓	✓	empty	?	∞_1^2	∞_2^2	∞_3^2
X	X	✓		empty	∞_1^1	∞_2^1	∞_3^1
X	X	X					(if universal without infinite allocation, still universal with infinite allocation)
Read/ Write			T&S [Herlihy 1991]				

Filling the Hierarchy

			Universal without infinite allocation?				
Arrival Models	Infinite	Infinite	X	X	X	✓	
	Finite	Finite	X	X	✓	✓	
	Infinite	Finite	X	✓	✓	✓	
Universal with infinite allocation?	✓	✓	✓	binary consensus	∞_1^3 binary consensus + iterator stack	∞_3^3 consensus	
	X	✓	✓	empty	∞_1^2 window registers	∞_2^2 iterator stack	
	X	X	✓	empty	∞_1^1	∞_2^1 ∞_3^1	
	X	X	X	(if universal without infinite allocation, still universal with infinite allocation)			
Read/ Write			T&S ...	[Herlihy1991]			