The Semantics of Progress in Lock-Based Transactional Memory

Rachid Guerraoui  Michał Kapałka

EPFL, Switzerland
Transactional Memory
Transactional Memory

thread 1  $T_1$  commit

thread 2  abort
Practice

DSTM, SXM, TL2, RSTM, JVSTM, NZTM, Haskell STM, TinySTM, McRT-STM, BartokSTM, SwissTM, …
Practice  

DSTM, SXM, TL2, RSTM, JVSTM, NZTM, Haskell STM, TinySTM, McRT-STM, BartokSTM, SwissTM, …

semantics, inherent limitations  

Theory
Semantics
of
Transactional Memory
2 aspects
correctness condition

nothing bad happens
correctness condition

opacity

nothing bad happens
progress property

something good happens
progress property:

When can a transaction be aborted?
progress property

something good happens

obstruction-freedom [SPAA’08]
progress property

lock-based TMs?

something good happens

this talk
Lock-Based TM Implementation

$T$

read

write

try-lock A

try-lock B

TM implementation
progress property

lock-based TM = ensures strong progressiveness

something good happens
Strongly progressive TMs
TL2, TinySTM, RSTM, BartokSTM, McRT-STM, ...
Contributions

Lock-based TMs

- progress semantics
  - strong progressiveness
Contributions

Lock-based TMs

- progress semantics
- strong progressiveness
- ≡ strong try-lock
- consensus #2
Contributions

Lock-based TMs

- progress semantics
- ≡ strong try-lock
- reduction

- strong progressiveness
- consensus #2
- proving progress
Reduction

$T_1$

$T_2$

TM implementation

try-lock A

try-lock B

A

B
Contributions

Lock-based TMs

\[\equiv\text{ strong progressiveness}\]
\[\equiv\text{ try-lock}\]
\[\equiv\text{ consensus #2}\]
\[\equiv\text{ proving progress}\]
\[\equiv\text{ exponential space with invisible reads}\]
Inherent Complexity

\[ T \]

\[ \Omega \left( e^k \right) \]

shared memory

TM implementation

A
B

k
Strong Progressiveness
If a group of concurrent transactions conflict on at most 1 object, then 1 of those must commit.
Example 1

$T_1 \quad \text{commit}$
Example 2

$T_1$——commit——$T_2$
Example 3

$T_1$  

A  

B  

write  

$T_2$  

C  

$T_3$  

commit or commit or commit
Strong Progressiveness

If a group of concurrent transactions conflict on at most 1 object, then 1 of those must commit.
If a group of concurrent transactions conflict on at most 1 object, then 1 of those must commit.
In the Paper...

Lock-based TMs

- progress semantics
  - strong progressiveness

- \( \equiv \) strong try-lock
  - consensus #2

- reduction
  - proving progress

- inherent limitation
  - exponential space with invisible reads
How much progress can a TM ensure?