Memory Model-aware Testing
a Unified Complexity Analysis

Florian Furbach, Roland Meyer,
Klaus Schneider, and Maximilian Senftleben

University of Kaiserslautern
Department of Computer Science
Germany

ACSD 2014

June 26, 2014
Introduction
Motivation

- Programmers expect sequential consistency.

Gibbons, Korach 1997
Cantin, Lipasti, Smith 2005
Programmers expect sequential consistency.

Modern architectures lack sequential consistency.

---

Gibbons, Korach 1997
Cantin, Lipasti, Smith 2005
Introduction
Motivation

- Programmers expect sequential consistency.
- Modern architectures lack sequential consistency.
- Modern architectures employ weak memory models.

Gibbons, Korach 1997
Cantin, Lipasti, Smith 2005
Introduction

Motivation

- Programmers expect sequential consistency.
- Modern architectures lack sequential consistency.
- Modern architectures employ weak memory models.
- Weak memory models may introduce undesired states.

Gibbons, Korach 1997
Cantin, Lipasti, Smith 2005
Introduction

Motivation

- Programmers expect sequential consistency.
- Modern architectures lack sequential consistency.
- Modern architectures employ weak memory models.
- Weak memory models may introduce undesired states.
- State explosion for reachability analysis.

Gibbons, Korach 1997
Cantin, Lipasti, Smith 2005
**Introduction**

**Motivation**

- Programmers expect sequential consistency.
- Modern architectures lack sequential consistency.
- Modern architectures employ weak memory models.
- Weak memory models may introduce undesired states.
- State explosion for reachability analysis.

- **Complexity of Testing?**

---

Gibbons, Korach 1997  
Cantin, Lipasti, Smith 2005
Introduction

Notions

- Test: sequences of reads/writes for multiple processes.
- Reads are blocking.
- Memory variables initialized to 0.

**Example:** Test \( \tau \)
Introduction

Notions

- Test: sequences of reads/writes for multiple processes.
- Reads are blocking.
- Memory variables initialized to 0.

**Example:** Test $\mathcal{T}$

```
P1: (w, x, 1)  (w, x, 2)
P2: (r, x, 1)
```

$\downarrow$

$x$: 0
Introduction

Notions

- Test: sequences of reads/writes for multiple processes.
- Reads are blocking.
- Memory variables initialized to 0.

**Example:** Test $\mathcal{T}$

\[ \begin{align*}
\text{P1} & : (w, x, 1) \\
& \quad (w, x, 2) \\
\text{P2} & : (r, x, 1) \\
\end{align*} \]

$\downarrow$ : $(w, x, 1)$

$x$ : 1
Introduction

Notions

- Test: sequences of reads/writes for multiple processes.
- Reads are blocking.
- Memory variables initialized to 0.

**Example:** Test $\mathcal{T}$

\[ (w, x, 1) \]
\[ (w, x, 2) \]
\[ (r, x, 1) \]

$\downarrow: (w, x, 1).(r, x, 1)$

$x: 1$
Introduction

Notions

- Test: sequences of reads/writes for multiple processes.
- Reads are blocking.
- Memory variables initialized to 0.

**Example:** Test $\mathcal{T}$

- $P_1$: $(w, x, 1)$, $(w, x, 2)$
- $P_2$: $(r, x, 1)$

$\downarrow$: $(w, x, 1).(r, x, 1).(w, x, 2)$

$x$: 2
Serial View

- Processes observe operations in different orders (views).
- A **serial view** $\leftarrow = \text{SerialView}(\mathcal{O},<)$ is a sequence of operations from $\mathcal{O}$ that respects some partial order $<.$
- Always read from last write.

Steinke, Nutt 2004
Serial View

- Processes observe operations in different orders (views).
- A **serial view** $\triangleright = \text{SerialView}(\mathcal{O}, <)$ is a sequence of operations from $\mathcal{O}$ that respects some partial order $<$. 
- Always read from last write.

- A Test $\mathcal{T}$ is **executable under sequential consistency** if:

$$\exists \; \triangleright = \text{SerialView}(\mathcal{T}, <_{PO}).$$

Example $\triangleright : \; (w, x, 1).(r, x, 1).(w, x, 2)$

______________________________
Steinke, Nutt 2004
The Testing Problem

Testing Problem of model M:

*Given test $\mathcal{T}$, is it executable under model $\text{M}$?*
The Testing Problem

Testing Problem of model M:

*Given test \( \mathcal{T} \), is it executable under model \( \mathbf{M} \)?*

Steinke, Nutt 2004
The Testing Problem

Testing Problem of model $M$:

Given test $\mathcal{T}$, is it executable under model $M$?

Steinke, Nutt 2004
The Testing Problem

- Testing Problem is in NP for all models
- Testing Problem is NP-hard for most models
- Testing Problem is in P for some models
Testing is in NP

Uniform Reduction to SAT:

- Formula:
  \[ WT(T) \land SV_1 \land .. \land SV_k \]

  \( WT \): Unique Writes-To
  \( SV \): SerialView properties
Testing is in NP

Uniform Reduction to SAT:

- Formula:
  \[
  WT(T) \land SV_1 \land .. \land SV_k
  \]
  \(WT\): Unique Writes-To
  \(SV\): SerialView properties

- Boolean variable:
  \[
  sv_{i,j} \leftrightarrow (op_i \blacktriangleleft op_j)
  \]

- Serial view properties:
  Totality, Asymmetry, Transitivity, Read-Last-Write
The Testing Problem

- **Testing Problem is in NP for all models**
  - Uniform SAT reduction.
  - Optimal solution if NP-hard.

- Testing Problem is NP-hard for most models
- Testing Problem is in P for some models
The Testing Problem

- Testing Problem is in NP for all models
- **Testing Problem is NP-hard for most models**
- Testing Problem is in P for some models
The Testing Problem

- Testing Problem is in NP for all models
- **Testing Problem is NP-hard for most models**
  - Our proofs cover multiple models
- Testing Problem is in P for some models
NP-hard for most models

Range reduction

\[ M_{\text{Strong}} \leq M_{\text{Weak}} \text{-range reduction } f \text{ of SAT to testing:} \]

\[
\begin{align*}
\text{M}_{\text{Strong}} & \downarrow \\
\text{SAT} & \Downarrow \\
\text{M} & \Downarrow \\
\text{M} & \Downarrow \\
\text{M} & \Downarrow \\
\text{M}_{\text{Weak}} & \\
\end{align*}
\]

(i) \( \phi \) is SAT \( \implies \) test \( f(\phi) \) is executable under \( M_{\text{Strong}} \).

(ii) test \( f(\phi) \) is executable under \( M_{\text{Weak}} \) \( \implies \) \( \phi \) is SAT.
NP-hard for most models

Range reduction

\( M_{Strong} \leq M_{Weak} \) - range reduction \( f \) of SAT to testing:

(i) \( \phi \) is SAT \( \implies \) test \( f(\phi) \) is executable under \( M_{Strong} \).

(ii) test \( f(\phi) \) is executable under \( M_{Weak} \) \( \implies \phi \) is SAT.
NP-hard for most models

Range reduction

\[ M_{\text{Strong}} \leq M_{\text{Weak}} \text{-range reduction } f \text{ of SAT to testing:} \]

(i) \( \phi \) is SAT \( \iff \) test \( f(\phi) \) is executable under \( M_{\text{Strong}} \).
(ii) test \( f(\phi) \) is executable under \( M_{\text{Weak}} \) \( \iff \) \( \phi \) is SAT.
NP-hard for most models

Range reduction

\[ M_{Strong} \leq M_{Weak} \text{-range reduction } f \text{ of SAT to testing:} \]

\[ \begin{align*}
(i) & \quad \phi \text{ is SAT } \implies \text{test } f(\phi) \text{ is executable under } M_{Strong}. \\
(ii) & \quad \text{test } f(\phi) \text{ is executable under } M_{Weak} \implies \phi \text{ is SAT.}
\end{align*} \]
NP-hard for most models

\( SC \leq SLOW\text{-Range\text{-}Reduction} \)
NP-hard for most models

Slow Consistency

*Writes from one process to one variable are observed in same order by all processes.*
NP-hard for most models

Slow Consistency

*W*rites from one process to one variable are observed in same order by all processes.

- The **program order** is respected.
- For each process $p$ and variable $x$: there exists a serial view on all writes to $x$ and reads from $x$ of $p$.

\[
\forall x, p \exists \text{ SerialView}(\mathcal{T}|_{w,x} \cup \mathcal{T}|_{p,x}, <_{PO})
\]

\[
\exists \text{ SerialView}(\mathcal{T}, <_{PO}) \quad [SC]
\]

Hutto, Ahamad 1990
NP-hard for most models

Slow Consistency

*Writes from one process to one variable are observed in same order by all processes.*

- The program order is respected.
- For each process $p$ and variable $x$: there exists a serial view on all writes to $x$ and reads from $x$ of $p$.

\[
\forall x, p \exists \ SerialView(\ T_{w,x} \cup T_{p,x}, <PO)\]

\[
\exists \ SerialView(\ T, <PO) \quad [SC]
\]

Hutto, Ahamad 1990
NP-hard for most models

Slow Consistency

```
Writes from one process to one variable are observed in same order by all processes.
```

- The program order is respected.
- For each process $p$ and variable $x$: there exists a serial view on all writes to $x$ and reads from $x$ of $p$.

\[
\forall x, p \exists \downarrow = \text{SerialView}(\: T|_{w,x} \cup T|_{p,x} \: , <PO) \]

\[
\exists \downarrow = \text{SerialView}(\: T \: , <PO) \quad [\text{SC}] 
\]

Hutto, Ahamad 1990
NP-hard for most models
SC ≤ SLOW-Range-Reduction of SAT

Reduction Idea

- Test uses only one variable $\xi$.
- Test has only one process with reads.
- $\Rightarrow$ Test behaves the same from Slow to SC.

$$\forall x, p \exists = \text{SerialView}(\mathcal{T}|_{w,x} \cup \mathcal{T}|_{p,x}, <_{PO}) \quad [\text{Slow}]$$

$$\exists = \text{SerialView}(\mathcal{T}, <_{PO}) \quad [\text{SC}]$$
NP-hard for most models

\[ SC \leq \text{SLOW-Range-Reduction of SAT} \]

Reduction Idea

- Test uses only one variable \( \xi \).
- Test has only one process with reads.
- \( \Rightarrow \) Test behaves the same from Slow to SC.

\[
\forall x, p \exists \bigtriangleup = \text{SerialView}( T|_{w,x} \cup T|_{p,x}, <_{PO}) \quad [\text{Slow}]
\]

\[
\exists \bigtriangleup = \text{SerialView}( T, <_{PO}) \quad [SC]
\]

SAT-Reduction

- We associate clauses and variables with values of \( \xi \).
SC-Slow Reduction - Example

\[
\begin{array}{c}
(a \lor b) \land \neg a \\
\text{cl}_1 \quad \text{cl}_2
\end{array}
\]

\[a = false\]
\[b = true\]
SC-Slow Reduction - Example

\[(a \lor b) \land \neg a\]

\[\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad
SC-Slow Reduction - Example

\[(a \lor b) \land \neg a\]

\[\begin{align*}
  &\quad \text{cl}_1 \\
  &\quad \text{cl}_2
\end{align*}\]

\[a = \text{false}\]
\[b = \text{true}\]

\[\text{True}_a := (w, \xi, \text{cl1}) \cdot (w, \xi, a)\]
\[\text{False}_a := (w, \xi, \text{cl2}) \cdot (w, \xi, a)\]
\[\text{True}_b := (w, \xi, \text{cl1}) \cdot (w, \xi, b)\]
\[\text{False}_b := (w, \xi, b)\]
SC-Slow Reduction - Example

\[(a \lor b) \land \lnot a\]

\(\begin{array}{c}
\text{cl}_1 \\
\text{cl}_2
\end{array}\)

\(a = \text{false}\)

\(b = \text{true}\)

\[\text{True}_a := (w, \xi, \text{cl}1) \cdot (w, \xi, a)\]

\[\text{False}_a := (w, \xi, \text{cl}2) \cdot (w, \xi, a)\]

\[\text{True}_b := (w, \xi, \text{cl}1) \cdot (w, \xi, b)\]

\[\text{False}_b := (w, \xi, b)\]

\[\text{Eval} := (r, \xi, a) \cdot (r, \xi, b) \cdot (r, \xi, \text{cl}1) \cdot (r, \xi, \text{cl}2)\]

\[\triangledown := \]

\[\xi := 0\]
SC-Slow Reduction - Example

\[
\begin{align*}
(a \lor b) \land \neg a
\end{align*}
\]

\[
\begin{align*}
\text{cl}_1 & \quad \text{cl}_2
\end{align*}
\]

\[
\begin{align*}
a = \text{false} \\
b = \text{true}
\end{align*}
\]

\[
\begin{align*}
\text{True}_a := (w, \xi, \text{cl}1) \cdot (w, \xi, a) \\
\text{False}_a := (w, \xi, \text{cl}2) \cdot (w, \xi, a) \\
\text{True}_b := (w, \xi, \text{cl}1) \cdot (w, \xi, b) \\
\text{False}_b := (w, \xi, b) \\
\text{Eval} := (r, \xi, a) \cdot (r, \xi, b) \cdot (r, \xi, \text{cl}1) \cdot (r, \xi, \text{cl}2)
\end{align*}
\]

\[
\begin{align*}
\downarrow & : (w, \xi, \text{cl}1) \\
\xi & : 1
\end{align*}
\]
SC-Slow Reduction - Example

\[
\begin{align*}
(a \lor b) \land \neg a & \quad \text{cl}_1 \\
\neg a & \quad \text{cl}_2
\end{align*}
\]

\[
\begin{align*}
(a \lor b) \land \neg a & \quad a = \text{false} \\
\neg a & \quad b = \text{true}
\end{align*}
\]

\[
\begin{align*}
\text{True}_a & := (w, \xi, \text{cl1}) \cdot (w, \xi, a) \\
\text{False}_a & := (w, \xi, \text{cl2}) \cdot (w, \xi, a) \\
\text{True}_b & := (w, \xi, \text{cl1}) \cdot (w, \xi, b) \\
\text{False}_b & := (w, \xi, b) \\
\text{Eval} & := (r, \xi, a) \cdot (r, \xi, b) \cdot (r, \xi, \text{cl1}) \cdot (r, \xi, \text{cl2})
\end{align*}
\]

\[
\begin{align*}
\downarrow & \quad (w, \xi, \text{cl1}) \cdot (w, \xi, a)
\end{align*}
\]

\[
\xi : \quad a
\]
SC-Slow Reduction - Example

\[
\begin{align*}
(a \lor b) & \land \neg a \\
& \quad \\ \\
& cl_1 \quad cl_2
\end{align*}
\]

\[a = \text{false} \quad b = \text{true}\]

\[\text{True}_a := (w, \xi, cl1) . (w, \xi, a)\]

\[\text{False}_a := (w, \xi, cl2) . (w, \xi, a)\]

\[\text{True}_b := (w, \xi, cl1) . (w, \xi, b)\]

\[\text{False}_b := (w, \xi, b)\]

\[\text{Eval} := (r, \xi, a) . (r, \xi, b) . (r, \xi, cl1) . (r, \xi, cl2)\]

\[\triangleleft := (w, \xi, cl1). (w, \xi, a). (r, \xi, a)\]

\[\xi := a\]
\[
\frac{(a \lor b) \land \neg a}{cl_1} \quad \frac{\neg a}{cl_2}
\]

\begin{align*}
\text{True}_a &:= (w, \xi, cl_1) \cdot (w, \xi, a) \\
\text{False}_a &:= (w, \xi, cl_2) \cdot (w, \xi, a) \\
\text{True}_b &:= (w, \xi, cl_1) \cdot (w, \xi, b) \\
\text{False}_b &:= (w, \xi, b) \\
\text{Eval} &:= (r, \xi, a) \cdot (r, \xi, b) \cdot (r, \xi, cl_1) \cdot (r, \xi, cl_2)
\end{align*}

\[\blacktriangleleft : (w, \xi, cl_1).(w, \xi, a).(r, \xi, a).(w, \xi, b)\]

\[\xi : b\]
SC-Slow Reduction - Example

\[
\frac{(a \lor b) \land \neg a}{cl_1 \quad cl_2} \quad a = \text{false} \\
\quad \quad b = \text{true}
\]

\[
\begin{align*}
\text{True}_a & := (w, \xi, cl1) \cdot (w, \xi, a) \\
\text{False}_a & := (w, \xi, cl2) \cdot (w, \xi, a) \\
\text{True}_b & := (w, \xi, cl1) \cdot (w, \xi, b) \\
\text{False}_b & := (w, \xi, b) \\
\text{Eval} & := (r, \xi, a) \cdot (r, \xi, b) \cdot (r, \xi, cl1) \cdot (r, \xi, cl2)
\end{align*}
\]

\[
\triangleleft : (w, \xi, cl1).(w, \xi, a).(r, \xi, a).(w, \xi, b).(r, \xi, b)
\]

\[
\xi : \quad b
\]
SC-Slow Reduction - Example

\[(a \lor b) \land \neg a\]

\[\begin{aligned}
  \cdot cl_1 & \quad \cdot cl_2 \\
  a &= \text{false} \\
  b &= \text{true}
\end{aligned}\]

\[\begin{aligned}
  True_a & := (w, \xi, cl1) \cdot (w, \xi, a) \\
  False_a & := (w, \xi, cl2) \cdot (w, \xi, a) \\
  True_b & := (w, \xi, cl1) \cdot (w, \xi, b) \\
  False_b & := (w, \xi, b) \\
  Eval & := (r, \xi, a) \cdot (r, \xi, b) \cdot (r, \xi, cl1) \cdot (r, \xi, cl2)
\end{aligned}\]

\[\downarrow : (w, \xi, cl1).(w, \xi, a).(r, \xi, a).(w, \xi, b).(r, \xi, b)\]

\[\cdot (w, \xi, cl1)\]

\[\xi : 1\]
SC-Slow Reduction - Example

\[(a \lor b) \land \neg a\]

\[
\begin{align*}
cl_1 & \quad cl_2 \\
\end{align*}
\]

\[a = false \quad b = true\]

\[
True_a := (w, \xi, cl1) \cdot (w, \xi, a)
\]

\[
False_a := (w, \xi, cl2) \cdot (w, \xi, a)
\]

\[
True_b := (w, \xi, cl1) \cdot (w, \xi, b)
\]

\[
False_b := (w, \xi, b)
\]

\[
Eval := (r, \xi, a) \cdot (r, \xi, b) \cdot (r, \xi, cl1) \cdot (r, \xi, cl2)
\]

\[
\downarrow : \quad (w, \xi, cl1).(w, \xi, a).(r, \xi, a).(w, \xi, b).(r, \xi, b)
\]

\[
\cdot (w, \xi, cl1).(r, \xi, cl1)
\]

\[
\xi : \quad 1
\]
SC-Slow Reduction - Example

\[(a \lor b) \land \neg a\]
\[\text{cl}_1 \quad \text{cl}_2\]
\[a = \text{false}\]
\[b = \text{true}\]

\[True_a := (w, \xi, \text{cl}1) \cdot (w, \xi, a)\]

\[False_a := (w, \xi, \text{cl}2) \cdot (w, \xi, a)\]

\[True_b := (w, \xi, \text{cl}1) \cdot (w, \xi, b)\]

\[False_b := (w, \xi, b)\]

\[Eval := (r, \xi, a) \cdot (r, \xi, b) \cdot (r, \xi, \text{cl}1) \cdot (r, \xi, \text{cl}2)\]

\[\updownarrow: \quad (w, \xi, \text{cl}1).(w, \xi, a).(r, \xi, a).(w, \xi, b).(r, \xi, b)\]
\[.\quad (w, \xi, \text{cl}1).(r, \xi, \text{cl}1). (w, \xi, \text{cl}2)\]

\[\xi: \quad 2\]
SC-Slow Reduction - Example

\[
\frac{(a \lor b) \land \neg a}{cl_1} \quad \frac{\neg a}{cl_2}
\]

\[
a = false
\]
\[
b = true
\]

\[
True_a := (w, \xi, cl1) . (w, \xi, a)
\]
\[
False_a := (w, \xi, cl2) . (w, \xi, a)
\]
\[
True_b := (w, \xi, cl1) . (w, \xi, b)
\]
\[
False_b := (w, \xi, b)
\]
\[
Eval := (r, \xi, a) . (r, \xi, b) . (r, \xi, cl1) . (r, \xi, cl2)
\]

\[
\triangleright: \ (w, \xi, cl1).(w, \xi, a).(r, \xi, a).(w, \xi, b).(r, \xi, b)
\]
\[
\ . (w, \xi, cl1).(r, \xi, cl1).(w, \xi, cl2).(r, \xi, cl2)
\]
\[
\xi: \ 2
\]
SC-Slow Reduction - Example

\[
\begin{align*}
(a \lor b) \land \neg a & \quad a = \text{false} \\
c_{l1} & \quad b = \text{true} \\
c_{l2}
\end{align*}
\]

\[
\begin{align*}
\text{True}_a & := (w, \xi, cl1) . (w, \xi, a) \\
\text{False}_a & := (w, \xi, cl2) . (w, \xi, a) \\
\text{True}_b & := (w, \xi, cl1) . (w, \xi, b) \\
\text{False}_b & := (w, \xi, b) \\
\text{Eval} & := (r, \xi, a) . (r, \xi, b) . (r, \xi, cl1) . (r, \xi, cl2)
\end{align*}
\]

\[
\begin{align*}
\triangledown : & \quad (w, \xi, cl1).(w, \xi, a).(r, \xi, a).(w, \xi, b).(r, \xi, b) \\
& \quad .(w, \xi, cl1).(r, \xi, cl1).(w, \xi, cl2).(r, \xi, cl2).(w, \xi, a) \\
\xi : & \quad a
\end{align*}
\]
SC-Slow Reduction - Example

\[
\frac{(a \lor b) \land \neg a}{c_{l1}} \quad c_{l2} \quad a = false \\
\]

\[
b = true
\]

\[
True_a := (w, \xi, c1). (w, \xi, a)
\]

\[
False_a := (w, \xi, c2). (w, \xi, a)
\]

\[
True_b := (w, \xi, c1). (w, \xi, b)
\]

\[
False_b := (w, \xi, b)
\]

\[
Eval := (r, \xi, a). (r, \xi, b). (r, \xi, c1). (r, \xi, c2)
\]

\[
\blacktriangledown := (w, \xi, c1). (w, \xi, a). (r, \xi, a). (w, \xi, b). (r, \xi, b) \\
\quad . (w, \xi, c1). (r, \xi, c1). (w, \xi, c2). (r, \xi, c2). (w, \xi, a). (w, \xi, b)
\]

\[
\xi := b
\]
SC-Slow Reduction - Example

\[
\frac{(a \lor b) \land \neg a}{cl_1} \quad \frac{b = true}{cl_2} \quad \text{a = false}
\]

\[
True_a := (w, \xi, cl1) \cdot (w, \xi, a)
\]

\[
False_a := (w, \xi, cl2) \cdot (w, \xi, a)
\]

\[
True_b := (w, \xi, cl1) \cdot (w, \xi, b)
\]

\[
False_b := (w, \xi, b)
\]

\[
Eval := (r, \xi, a) \cdot (r, \xi, b) \cdot (r, \xi, cl1) \cdot (r, \xi, cl2)
\]

\[
\downarrow : (w, \xi, cl1). (w, \xi, a). (r, \xi, a). (w, \xi, b). (r, \xi, b).
\]

\[
\cdot (w, \xi, cl1). (r, \xi, cl1). (w, \xi, cl2). (r, \xi, cl2). (w, \xi, a). (w, \xi, b)
\]

\[
\xi : b
\]
## Results

<table>
<thead>
<tr>
<th>Memory Model</th>
<th>Complexity Class of $\text{Test}(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General</td>
</tr>
<tr>
<td>SC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>TSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-G</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-D</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GAO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GPO+GDO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>Causal</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM-M</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GWO</td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>SLOW</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>LOCAL</td>
<td></td>
</tr>
</tbody>
</table>
### Results

<table>
<thead>
<tr>
<th>Memory Model</th>
<th>Complexity Class of $\text{Test}(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General</td>
</tr>
<tr>
<td>SC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>TSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-G</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-D</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GAO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GPO+GDO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>Causal</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM-M</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GWO</td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>SLOW</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>LOCAL</td>
<td>$P_2$</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Memory Model</th>
<th>Complexity Class of Test(M)</th>
<th>General</th>
<th>Process</th>
<th>Length</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC</td>
<td>NPC(by 1)</td>
<td>NPC(by 3)</td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSO</td>
<td>NPC(by 1)</td>
<td></td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSO</td>
<td>NPC(by 1)</td>
<td></td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC-G</td>
<td>NPC(by 1)</td>
<td></td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC-D</td>
<td>NPC(by 1)</td>
<td></td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAO</td>
<td>NPC(by 1)</td>
<td></td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPO+GDO</td>
<td>NPC(by 1)</td>
<td></td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Causal</td>
<td>NPC(by 1)</td>
<td>NPC(by 3)</td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRAM-M</td>
<td>NPC(by 1)</td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GWO</td>
<td>NPC(by 3)</td>
<td>NPC(by 3)</td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>NPC(by 1)</td>
<td></td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRAM</td>
<td>NPC(by 1)</td>
<td></td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLOW</td>
<td>NPC(by 1)</td>
<td></td>
<td>NPC(by 1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL</td>
<td>$P_2$</td>
<td>P(by 2)</td>
<td>P(by 2)</td>
<td>P(by 2)</td>
<td></td>
</tr>
</tbody>
</table>
Results

<table>
<thead>
<tr>
<th>Memory Model</th>
<th>Complexity Class of $\text{Test}(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General</td>
</tr>
<tr>
<td>SC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>TSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-G</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-D</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GAO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GPO+GDO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>Causal</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM-M</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GWO</td>
<td>NPC(by 3)</td>
</tr>
<tr>
<td>CC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>SLOW</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>LOCAL</td>
<td>$P_2$</td>
</tr>
</tbody>
</table>
Results

<table>
<thead>
<tr>
<th>Memory Model</th>
<th>Complexity Class of TEST(M)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General</td>
</tr>
<tr>
<td>SC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>TSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-G</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-D</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GAO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GPO+GDO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>Causal</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM-M</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GWO</td>
<td>NPC(by 3)</td>
</tr>
<tr>
<td>CC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>SLOW</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>LOCAL</td>
<td>P_2</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Memory Model</th>
<th>Complexity Class of Test(M)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General</td>
</tr>
<tr>
<td>SC</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>TSO</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>PSO</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>PC-G</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>PC-D</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>GAO</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>GPO+GDO</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>Causal</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>PRAM-M</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>GWO</td>
<td><strong>NPC</strong>(by 3)</td>
</tr>
<tr>
<td>CC</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>PRAM</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>SLOW</td>
<td><strong>NPC</strong>(by 1)</td>
</tr>
<tr>
<td>LOCAL</td>
<td><strong>P_2</strong></td>
</tr>
</tbody>
</table>
Results

<table>
<thead>
<tr>
<th>Memory Model</th>
<th>Complexity Class of $\text{Test}(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General</td>
</tr>
<tr>
<td>SC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>TSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-G</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-D</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GAO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GPO+GDO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>Causal</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM-M</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GWO</td>
<td>NPC(by 3)</td>
</tr>
<tr>
<td>CC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>SLOW</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>LOCAL</td>
<td>P_2</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Memory Model</th>
<th>Complexity Class of $\text{Test}(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General</td>
</tr>
<tr>
<td>SC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>TSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PSO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-G</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PC-D</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GAO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GPO+GDO</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>Causal</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM-M</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>GWO</td>
<td>NPC(by 3)</td>
</tr>
<tr>
<td>CC</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>PRAM</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>SLOW</td>
<td>NPC(by 1)</td>
</tr>
<tr>
<td>LOCAL</td>
<td>$P_2$</td>
</tr>
</tbody>
</table>