A Theory of Speculative Computation

Gérard Boudol, Gustavo Petri

Projet INDES

January 2010 – Parsec meeting

Speculations: Motivation

- Speculative computation
  - Value prediction
  - Branch prediction
  - Instruction reordering
- Relaxed memory models
  - Write-buffers allow for W $\rightarrow$ R and W $\rightarrow$ W
  - But not for R $\rightarrow$ R and R $\rightarrow$ W

IRIW example

initially $x = y = 0$

$x := 1$	$y := 1$	$!x\,;$ (1)	$!y\,;$ (1)
		$!y$ (0)	$!x$ (0)

Speculations: Motivation

- Speculative computation
  - Value prediction
  - Branch prediction
  - Instruction reordering
- Relaxed memory models
  - Write-buffers allow for W $\rightarrow$ R and W $\rightarrow$ W
  - But not for R $\rightarrow$ R and R $\rightarrow$ W

IRIW example

initially $x = y = 0$

$x := 1$	$y := 1$	$!x\,;$ (1)	$!y\,;$ (1)
		$!y$ (0)	$!x$ (0)

Speculations could explain these behaviors

Valid speculations: an intuition

Intuitively valid

$$(\text{if } \mathbf{!}p \text{ then } () \text{ else } q := tt) \xrightarrow{\mathsf{wr}_{q,tt}}$$
$$(\text{if } \mathbf{!}p \text{ then } () \text{ else } ()) \xrightarrow{\mathsf{rd}_{p,ff}}$$
$$(\text{if } ff \text{ then } () \text{ else } ()) \xrightarrow{\searrow}$$
$$()$$

Valid speculations: an intuition

Intuitively valid

$$(\text{if } !p \text{ then } () \text{ else } q := tt) \xrightarrow{\mathsf{wr}_{q,tt}}$$
$$(\text{if } !p \text{ then } () \text{ else } ()) \xrightarrow{\mathsf{rd}_{p,ff}}$$
$$(\text{if } ff \text{ then } () \text{ else } ()) \xrightarrow{\searrow}$$
$$()$$

Intuitively invalid

$$(\text{if } !p \text{ then } () \text{ else } p := tt) \xrightarrow{\mathsf{wr}_{p,tt}}$$
$$(\text{if } !p \text{ then } () \text{ else } ()) \xrightarrow{\mathsf{rd}_{p,tt}}$$
$$(\text{if } tt \text{ then } () \text{ else } ()) \xrightarrow{\swarrow}$$
$$()$$

Valid speculations: an intuition

Intuitively valid	Intuitively invalid
$(\text{if } !p \text{ then } () \text{ else } q := tt) \xrightarrow{\mathsf{wr}_{q,tt}}$	$(\text{if } !p \text{ then } () \text{ else } p := tt) \xrightarrow{\mathsf{wr}_{p,tt}}$
$(\text{if } !p \text{ then } () \text{ else } ()) \xrightarrow{\mathsf{rd}_{p,ff}}$	$(\text{if } !p \text{ then } () \text{ else } ()) \xrightarrow{\mathsf{rd}_{p,tt}}$
$(\text{if } ff \text{ then } () \text{ else } ()) \xrightarrow{\searrow}$	$(\text{if } tt \text{ then } () \text{ else } ()) \xrightarrow{\swarrow}$
$()$	$()$

Validity

We say that a speculative computation is *valid* when it is equivalent by permutations [Berry&Levy'79] to a normal (sequential) computation.

Concurrent speculations

Programmability is an issue with parallel speculations (as it is in relaxed memory models)

- Programmability compromise in relaxed memory models for high-level languages:

Data Race Freeness (DRF)

Programs free of data races in their interleaving semantics, expose (only) sequentially consistent behaviors in the relaxed semantics.

- Can we find a similar compromise for parallel speculations?

Speculative Data Race Freeness

Outline & Contributions

1. Operational semantics for speculations (with locks):
   - Speculative evaluation contexts:
     out-of-order execution, branch prediction
   - Value prediction
2. Validity of speculations
3. Programmability: SDRF
4. A variation of the language with memory barriers

The language (locks)

$$
\begin{array}{lcll}
v & ::= & x \mid \lambda x e \mid tt \mid ff \mid () & \textit{values} \\
e & ::= & v \mid (e_0\, e_1) & \textit{expressions} \\
 & \mid & (\mathsf{if}\ e\ \mathsf{then}\ e_0\ \mathsf{else}\ e_1) & \\
 & \mid & (\mathsf{ref}\ e) \mid (!\, e) \mid (e_0 := e_1) & \\
 & \mid & (\mathsf{thread}\, e) \mid (\mathsf{with}\ \ell\ \mathsf{do}\ e) &
\end{array}
$$

$e_0\, ;\, e_1$ stands for $(\lambda x e_1\, e_0)$ whenever x is not free in e_1

$$
\begin{array}{lcll}
\mathbf{E} & ::= & [] \mid \mathbf{E}[\mathbf{F}] & \textit{evaluation contexts} \\
\mathbf{F} & = & ([]\, e) \mid (v\, []) & \textit{frames} \\
 & \mid & (\mathsf{if}\ []\ \mathsf{then}\ e_0\ \mathsf{else}\ e_1) & \\
 & \mid & (\mathsf{ref}\, []) \mid (!\, []) \mid ([] := e) \mid (v := []) & \\
 & \mid & (\mathsf{holding}\ \ell\ \mathsf{do}\ []) &
\end{array}
$$

Speculation Contexts

$$
\begin{array}{rcll}
\mathbf{\Sigma} & ::= & [\,] \mid \mathbf{\Sigma}[\mathbf{\Phi}] & \textit{speculation contexts} \\
\mathbf{\Phi} & ::= & \mathbf{F} & \textit{speculation frames} \\
& \mid & (e\,[\,]) \mid (\lambda x[\,]\,e) & \\
& \mid & (\text{if } e \text{ then } [\,] \text{ else } e_1) \mid (\text{if } e \text{ then } e_0 \text{ else } [\,]) & \\
& \mid & (e := [\,]) &
\end{array}
$$

$(\lambda x[\,]e_0)$ can be seen as $e_0\,;[\,]$

No speculation for (with ℓ do e)

$r := (!p)\ ; q := tt$

Speculation Contexts

$$
\begin{array}{rcll}
\mathbf{\Sigma} & ::= & [] \mid \mathbf{\Sigma}[\mathbf{\Phi}] & \textit{speculation contexts} \\
\mathbf{\Phi} & ::= & \mathbf{F} & \textit{speculation frames} \\
& \mid & (e\,[]) \mid (\lambda x[]\,e) & \\
& \mid & (\text{if } e \text{ then } [] \text{ else } e_1) \mid (\text{if } e \text{ then } e_0 \text{ else } []) & \\
& \mid & (e := []) &
\end{array}
$$

$(\lambda x[]e_0)$ can be seen as $e_0\,;[]$

No speculation for (with ℓ do e)

$$
r := \underbrace{(\mathbf{!}p)}_{\mathbf{E}[(\mathbf{!}p)]}\ ;\ q := tt
$$

Speculation Contexts

$$
\begin{array}{rcll}
\mathbf{\Sigma} & ::= & [\,] \mid \mathbf{\Sigma}[\mathbf{\Phi}] & \textit{speculation contexts} \\
\mathbf{\Phi} & ::= & \mathbf{F} & \textit{speculation frames} \\
& \mid & (e\,[\,]) \mid (\lambda x[\,]\,e) & \\
& \mid & (\text{if } e \text{ then } [\,] \text{ else } e_1) \mid (\text{if } e \text{ then } e_0 \text{ else } [\,]) & \\
& \mid & (e := [\,]) &
\end{array}
$$

$(\lambda x[\,]e_0)$ can be seen as $e_0\,;[\,]$

No speculation for (with ℓ do e)

$$
r := \underbrace{(\mathbf{!}p)}_{\mathbf{E}[(\mathbf{!}p)]}\ ;\ \overbrace{q := tt}^{\mathbf{\Sigma}[(q:=tt)]}
$$

Speculation Contexts

$$
\begin{array}{lcll}
\boldsymbol{\Sigma} & ::= & [] \mid \boldsymbol{\Sigma}[\boldsymbol{\Phi}] & \textit{speculation contexts} \\
\boldsymbol{\Phi} & ::= & \mathbf{F} & \textit{speculation frames} \\
& \mid & (e\,[]) \mid (\lambda x[]\,e) & \\
& \mid & (\text{if } e \text{ then } [] \text{ else } e_1) \mid (\text{if } e \text{ then } e_0 \text{ else } []) & \\
& \mid & (e := []) &
\end{array}
$$

$(\lambda x[]e_0)$ can be seen as $e_0\,;[]$

No speculation for (with ℓ do e)

$$r := \underbrace{(!p)}_{\mathbf{E}[(!p)]}\ ;\ \overbrace{q := tt}^{\boldsymbol{\Sigma}[(q:=tt)]}$$

$$(\text{if } \underbrace{(!r)}_{\mathbf{E}[(!r)]} \text{ then } \overbrace{p := tt}^{\boldsymbol{\Sigma}_0[(p:=tt)]} \text{ else } \overbrace{q := tt}^{\boldsymbol{\Sigma}_1[(q:=tt)]}\ )$$

Speculative semantics

$$
\begin{array}{rcll}
\mathbf{\Sigma}[(\lambda x e v)] & \xrightarrow[@\mathbf{\Sigma}]{\beta} & \mathbf{\Sigma}[\{x \mapsto v\}e] & \\
\mathbf{\Sigma}[(\text{if } tt \text{ then } e_0 \text{ else } e_1)] & \xrightarrow[@\mathbf{\Sigma}]{\swarrow} & \mathbf{\Sigma}[e_0] & \\
\mathbf{\Sigma}[(\text{if } ff \text{ then } e_0 \text{ else } e_1)] & \xrightarrow[@\mathbf{\Sigma}]{\searrow} & \mathbf{\Sigma}[e_1] & \\
\mathbf{\Sigma}[(\text{ref } v)] & \xrightarrow[@\mathbf{\Sigma}]{\nu_{p,v}} & \mathbf{\Sigma}[p] & \\
\mathbf{\Sigma}[(\mathbf{!}\, p)] & \xrightarrow[@\mathbf{\Sigma}]{\color{red}{\text{rd}_{p,v}}} & \mathbf{\Sigma}[v] & \\
\mathbf{\Sigma}[(p := v)] & \xrightarrow[@\mathbf{\Sigma}]{\text{wr}_{p,v}} & \mathbf{\Sigma}[()] & \\
\color{red}{\mathbf{E}}[(\text{thread } e)] & \xrightarrow[@\mathbf{E}]{\text{spw}_e} & \color{red}{\mathbf{E}}[()] & \\
\mathbf{\Sigma}[(\text{with } \ell \text{ do } e)] & \xrightarrow[@\mathbf{\Sigma}]{\mu} & \mathbf{\Sigma}[e] & \ell \in \lceil \mathbf{\Sigma} \rceil \\
\mathbf{\Sigma}[(\text{with } \ell \text{ do } e)] & \xrightarrow[@\mathbf{\Sigma}]{\curvearrowright \atop \ell} & \mathbf{\Sigma}[(\text{holding } \ell \text{ do } e)] & \ell \notin \lceil \mathbf{\Sigma} \rceil \\
\color{red}{\mathbf{E}}[(\text{holding } \ell \text{ do } v)] & \xrightarrow[@\mathbf{E}]{\curvearrowleft \atop \ell} & \color{red}{\mathbf{E}}[v] &
\end{array}
$$

Semantics: IRIW example

$$x := 1 \mid y := 1 \mid \mathbf{!}x \,;\, \mathbf{!}y \mid \mathbf{!}y \,;\, \mathbf{!}x$$

IRIW example (revisited)

Semantics: IRIW example

$$
\begin{array}{c|c|c|c}
x := 1 & y := 1 & \mathbf{!}x \,;\, \mathbf{!}y & \mathbf{!}y \,;\, \mathbf{!}x \\
 & & \downarrow \mathsf{rd}_{y,0} & \downarrow \mathsf{rd}_{x,0} \\
x := 1 & y := 1 & \mathbf{!}x \,;\, 0 & \mathbf{!}y \,;\, 0
\end{array}
$$

IRIW example (revisited)

Semantics: IRIW example

$$
\begin{array}{cccc}
x := 1 & \mid \; y := 1 & \mid \; !x\,;\,!y & \mid \; !y\,;\,!x \\
 & & \downarrow \mathsf{rd}_{y,0} & \downarrow \mathsf{rd}_{x,0} \\
x := 1 & \mid \; y := 1 & \mid \; !x\,;0 & \mid \; !y\,;0 \\
\downarrow \mathsf{wr}_{x,1} & \downarrow \mathsf{wr}_{y,1} & & \\
() & \mid \; () & \mid \; !x\,;0 & \mid \; !y\,;0
\end{array}
$$

IRIW example (revisited)

Semantics: IRIW example

$$x := 1 \mid y := 1 \mid \mathbf{!}x \,;\, \mathbf{!}y \mid \mathbf{!}y \,;\, \mathbf{!}x$$

$$\qquad\qquad\qquad\qquad \downarrow \mathsf{rd}_{y,0} \quad \downarrow \mathsf{rd}_{x,0}$$

$$x := 1 \mid y := 1 \mid \mathbf{!}x \,;\, 0 \mid \mathbf{!}y \,;\, 0$$

$$\downarrow \mathsf{wr}_{x,1} \quad \downarrow \mathsf{wr}_{y,1}$$

$$() \mid () \mid \mathbf{!}x \,;\, 0 \mid \mathbf{!}y \,;\, 0$$

$$\qquad\qquad\qquad\qquad \downarrow \mathsf{rd}_{x,1} \quad \downarrow \mathsf{rd}_{y,1}$$

$$() \mid () \mid 1 \,;\, 0 \mid 1 \,;\, 0$$

IRIW example (revisited)

Concurrency

$$\frac{e \xrightarrow[o]{\text{spw}_{e'}} e''}{(S, L, (t, e) \parallel T) \xrightarrow[t,o]{\text{spw}_{e'}} (S, L, (t, e'') \parallel (t', e') \parallel T)} \; t' \notin \text{dom}(T) \cup \{t\}$$

$$\frac{e \xrightarrow[o]{a} e'}{(S, L, (t, e) \parallel T) \xrightarrow[t,o]{a} (S', L', (t, e') \parallel T)} \; a \neq \text{spw}_{e''} \; \& \; (*)$$

$$(*) \begin{cases} a = \text{rd}_{p,v} & \Rightarrow \quad v = S(p) \; \& \; S' = S \; \& \; L' = L \\ a = \text{wr}_{p,v} & \Rightarrow \quad S' = S[p := v] \; \& \; L' = L \\ a = \overset{\curvearrowright}{\ell} & \Rightarrow \quad S' = S \; \& \; \ell \notin L \; \& \; L' = L \cup \{\ell\} \\ a = \overset{\curvearrowleft}{\ell} & \Rightarrow \quad S' = S \; \& \; L' = L - \{\ell\} \\ \quad \ldots & \end{cases}$$

The need for Validity

$$
\begin{array}{lllll}
[& \{p \mapsto \mathit{ff}\}, & \emptyset, & (\text{if } \mathbf{!}p \text{ then } p := \mathit{tt}) &] \\
 & & & \quad \downarrow \mathsf{wr}_{p,\mathit{tt}} & \\
[& \{p \mapsto \mathit{tt}\}, & \emptyset, & (\text{if } \mathbf{!}p \text{ then } ()) &] \\
 & & & \quad \downarrow \mathsf{rd}_{p,\mathit{tt}} & \\
[& \{p \mapsto \mathit{tt}\}, & \emptyset, & (\text{if } \mathit{tt} \text{ then } ()) &]
\end{array}
$$

Causality is not enforced by the semantics

The need for Validity

$$
\begin{array}{lllll}
[& \{p \mapsto \mathit{ff}\}, & \emptyset, & (\text{if } \mathbf{!}p \text{ then } p := \mathit{tt}) &] \\
 & & & \downarrow \mathsf{wr}_{p,\mathit{tt}} & \\
[& \{p \mapsto \mathit{tt}\}, & \emptyset, & (\text{if } \mathbf{!}p \text{ then } ()) &] \\
 & & & \downarrow \mathsf{rd}_{p,\mathit{tt}} & \\
[& \{p \mapsto \mathit{tt}\}, & \emptyset, & (\text{if } \mathit{tt} \text{ then } ()) &]
\end{array}
$$

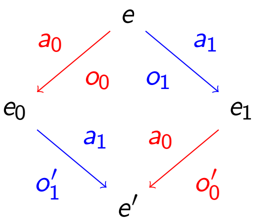
Causality is not enforced by the semantics

Validity

We say that a speculative computation is *valid* when it is equivalent by permutations [Berry&Levy'79] to a normal (sequential) computation.

Permutation equivalence

Diamond Lemma



where $(\star)$ $\begin{cases} o_1/(a_0, o_0) \equiv o_1' \\ o_0/(a_1, o_1) \equiv o_0' \end{cases}$

- $(\star)$ rules out permutations of control dependent events:

$$(\text{if } tt \text{ then } () \text{ else } p := tt) \xrightarrow{\mathsf{wr}_{p,tt}} \xrightarrow{\swarrow} ()$$

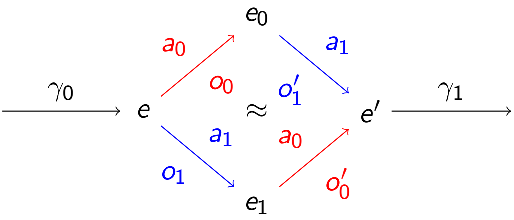
- What about permutation of data dependent events?

$$\# = \bigcup_{p \in \mathcal{R}ef, v, w \in \mathcal{V}al} \{(\mathsf{wr}_{p,v}, \mathsf{wr}_{p,w}), (\mathsf{wr}_{p,v}, \mathsf{rd}_{p,w}), (\mathsf{rd}_{p,v}, \mathsf{wr}_{p,w})\}$$

Validity: definition

Equivalence by Permutations

- Given that $\neg a_0 \# a_1$ we have:



Valid Speculative Computation

A speculation is valid if it is equivalent by permutation to a normal computation. A speculative computation γ is valid if all its thread projections $\gamma|_t$ are valid speculations

Speculatively Data Race Free

- By valid speculations we can explain most of the Java Memory Model litmus tests
- But it fails for DRF programs:

$$(\text{if } !p \text{ then } q := tt) \mid (\text{if } !q \text{ then } p := tt)$$

- We need a stronger property:

DRF Configuration (resp. Speculative DRF Configuration)

A configuration C is DRF (resp. SDRF) iff for any configuration C' reachable from C by normal (resp. speculative) computations, such that $C' \xrightarrow[t_0,o_0]{a_0} C_0$ and $C' \xrightarrow[t_1,o_1]{a_1} C_1$ we have

$$t_0 \neq t_1 \;\Rightarrow\; \neg(a_0 \# a_1)$$

SDRF result

Theorem (Main Result)

Every configuration reachable from a Speculatively Data Race Free closed expressions by a speculative computation is also reachable by a normal computation.

A lower level language (barriers)

- Assuming that we have locks is not necessarily realistic for lower level languages
- The DRF (cf. SDRF) guarantee is not very useful for these languages

$$
\begin{array}{rcll}
v & ::= & x \mid \lambda x e \mid tt \mid ff \mid () & \textit{values} \\
e & ::= & v \mid (e_0 e_1) & \textit{expressions} \\
 & \mid & (\text{if } e \text{ then } e_0 \text{ else } e_1) & \\
 & \mid & (\text{ref } e) \mid (\mathbf{!}\, e) \mid (e_0 := e_1) \mid (\text{spin e}) & \\
 & \mid & (\text{thread } e) \mid \text{rr} \mid \text{rw} \mid \text{wr} \mid \text{ww} &
\end{array}
$$

Validity

- The dependency relation permutations across barriers

$$
\begin{aligned}
\bowtie \quad = \quad & \# \bigcup_{p \in \mathcal{R}ef} \{(\mathsf{spw}, \mathsf{rd}_p), (\mathsf{spw}, \mathsf{rd}_p), (\mathsf{spw}, \mathsf{wr}_p), (\mathsf{wr}_p, \mathsf{spw})\} \\
& \cup \bigcup_{p \in \mathcal{R}ef} \{(\mathsf{rd}_p, \mathsf{rr}), (\mathsf{rr}, \mathsf{rd}_p), (\mathsf{wr}, \mathsf{rd}_p), (\mathsf{rd}_p, \mathsf{rw})\} \\
& \cup \bigcup_{p \in \mathcal{R}ef} \{(\mathsf{wr}_p, \mathsf{ww}), (\mathsf{ww}, \mathsf{wr}_p), (\mathsf{rw}, \mathsf{wr}_p), (\mathsf{wr}_p, \mathsf{wr})\}
\end{aligned}
$$

- Permutation equivalence as before (considering $\bowtie$)
- Valid Speculative computation as before

Preserving Order of Shared Memory Accesses

POSMA

A configuration C Preserves Ordering of Shared Memory Accesses (POSMA) iff for any *valid* speculative computation $\gamma : (C \xrightarrow{*} C')$ with

$$\gamma = \gamma_0 \cdot \xrightarrow[t, o_0]{a_0} \cdot \xrightarrow[t', o'_0]{a'_0} \cdot \gamma_1 \cdot \xrightarrow[t'', o'_1]{a'_1} \cdot \xrightarrow[t, o_1]{a_1} \cdot \gamma_2$$

and where $t' \neq t \neq t''$, $\neg(a_0 \# a_1)$, $a_0 \neq a_1$ and $a_i \# a'_i$ we have

$$[\gamma_0|_t, (a_0, o_0)] \prec_{\gamma|_t} [\gamma_0|_t \cdot \xrightarrow[a_0]{o_0} \cdot \gamma_1, (a_1, o_1)]$$

Theorem (POSMA Main Result)

Every configuration reachable from a POSMA well-formed closed configuration by a valid speculative computation is also reachable by a normal computation.

Some current work

- How do we make SDRF and POSMA useful for programming?
  - Common data-race detection type systems check for SDRF rather than DRF
  - Enforcement of SDRF by compilation [some work that we did]
    - Type-directed compilation
  - Enforcement of POSMA by compilation [work in progress]
- Prove that common synchronization implementations are POSMA (eg. spinlocks in TSO)