

Program verification under weak memory consistency

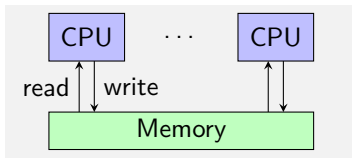
Viktor Vafeiadis

MPI-SWS

Marktobersdorf, August 2019

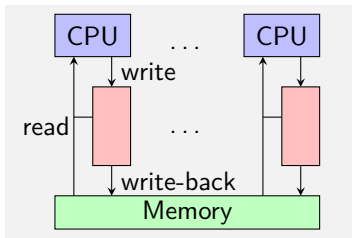
Sequential consistency (SC)

- ▶ Interleaving semantics
- ▶ **Intuitive, isn't it?**



Weak memory consistency

- ▶ Get all the SC behaviours
- ▶ Plus some weak behaviours
- ▶ **More complicated...**



Standard perception: WMC is *complicated*.

- ▶ Memory model definitions are complex.
- ▶ More behaviours to consider.
- ▶ Standard techniques (e.g., Owicki-Gries) are unsound.

Today: WMC is actually *quite easy*.

- ▶ When thinking in terms of sound reasoning principles.
 - ▶ Allows *local* and *causal* reasoning.
 - ▶ Forbids *global* reasoning.
- ▶ *Separation logic* nicely captures those principles.

Weak memory enforces *local* reasoning.

- ▶ Ownership-based reasoning.
- ▶ Proof of a thread mentions only variables accessed by it.
- ▶ Key underlying principle of [separation logic](#).

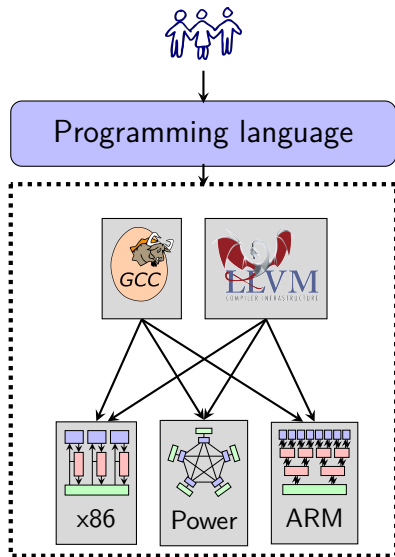
RA allows *causal* reasoning.

- ▶ I have seen an update, so I have seen all previous updates.
- ▶ “Ownership transfer” in [separation logic](#).

SC, in addition, allows *global* reasoning.

- ▶ Proof of a thread can mention local variables of other threads.
- ▶ Global reasoning is complicated.

Which memory model?



Choose a PL model

- ▶ Platform-independence
- ▶ Takes into account the compiler optimisations

C/C++11

- ▶ The main existing model
- ▶ Many interesting features
- ▶ But also partially broken
- ▶ Use fixed version(s)

The C11 Memory Model

- ▶ Introduced in the C/C++ 2011 standards
- ▶ Formalized along with the standard [Batty et al., POPL'11]
- ▶ Many proposed fixes [OOPSLA'13, POPL'15, PLDI'17]

Two types of locations

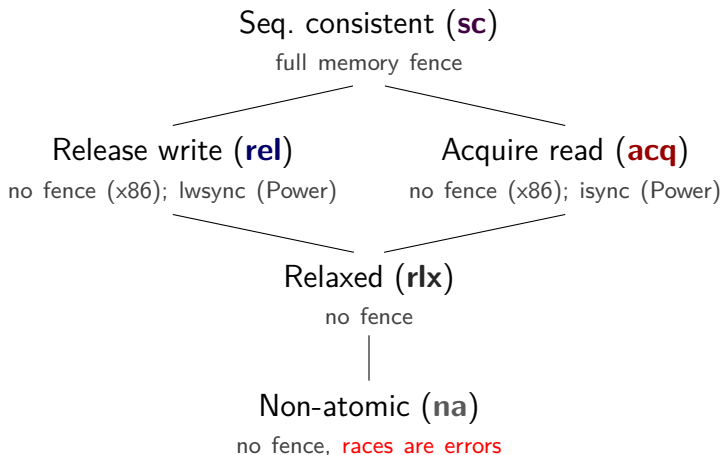
Ordinary
(Non-Atomic)

Races are **errors**

Atomic

Welcome to the
expert mode

A spectrum of accesses



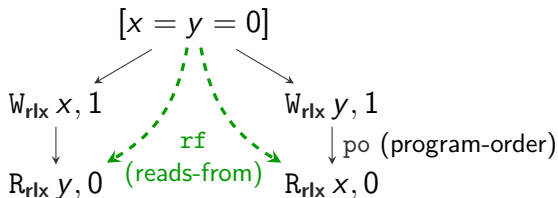
Explicit primitives for fences

Store buffering in C11

Initially $x = y = 0$.

$$\begin{array}{l} x_{rlx} := 1; \\ a := y_{rlx} \quad //0 \end{array} \parallel \begin{array}{l} y_{rlx} := 1; \\ b := x_{rlx} \quad //0 \end{array}$$

Can return $a = b = 0$ with the following execution:

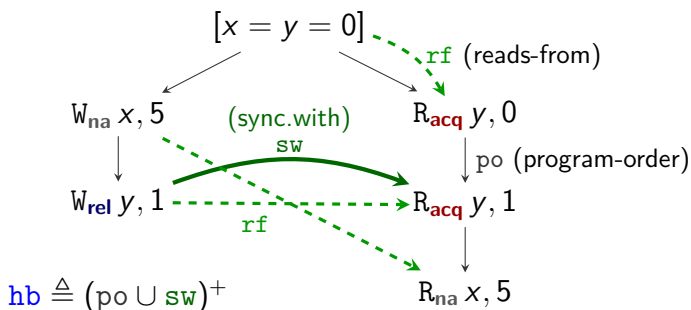


Release-acquire synchronization

Initially $x = y = 0$.

$x_{na} := 5;$		repeat
$y_{rel} := 1$		$a := y_{acq};$
		until $a \neq 0;$
		$b := x_{na}$

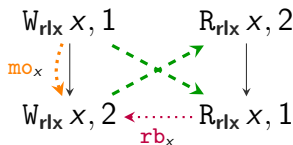
One possible execution:



Programs with a single shared variable behave as under SC.

$$\begin{array}{l} x_{rlx} := 1; \\ x_{rlx} := 2 \end{array} \parallel \begin{array}{l} a := x_{rlx}; // 2 \\ b := x_{rlx}; // 1 \end{array}$$

The outcome $a = 2 \wedge b = 1$ is forbidden.



- ▶ Modification order, mo_x , total order of writes to x .
- ▶ Reads-before : $rb \triangleq (rf^{-1}; mo) \cap (\neq)$
- ▶ Coherence : $hb \cup rf_x \cup mo_x \cup rb_x$ is acyclic for all x .

Relaxed program logics

- ▶ RSL (relaxed separation logic, OOPSLA'13)
- ▶ FSL (fenced separation logic, VMCAI'16)
- ▶ GPS (ghosts & protocols, OOPSLA'14, PLDI'15)

Key concept of *ownership* :

- ▶ Resourceful reading of Hoare triples.

$$\{P\} C \{Q\}$$

- ▶ To access a non-atomic location, you must own it:

$$\begin{aligned} & \{\text{emp}\} a := \mathbf{alloc} \{a \mapsto _ \} \\ & \{x \mapsto v\} a := x_{\text{na}} \quad \{x \mapsto v \wedge a = v\} \\ & \{x \mapsto v\} x_{\text{na}} := v' \quad \{x \mapsto v'\} \end{aligned}$$

- ▶ Disjoint parallelism:

$$\frac{\{P_1\} C_1 \{Q_1\} \quad \{P_2\} C_2 \{Q_2\}}{\{P_1 * P_2\} C_1 \parallel C_2 \{Q_1 * Q_2\}}$$

$$\begin{array}{c}
 \{x \mapsto 0\} \\
 \mathbf{a} := x_{na}; \\
 \{x \mapsto 0 \wedge a = 0\} \\
 x_{na} := a + 1; \\
 \{x \mapsto 1\}
 \end{array}
 \parallel
 \begin{array}{c}
 \{y \mapsto 0\} \\
 \mathbf{b} := y_{na}; \\
 \{y \mapsto 0 \wedge b = 0\} \\
 y_{na} := b + 1; \\
 \{y \mapsto 1\}
 \end{array}$$

$$\{x \mapsto 0 * y \mapsto 0\}$$

$$\{x \mapsto 1 * y \mapsto 1\}$$

Simple programs are easy to verify!

Ownership transfer by release/acquire synchronizations.

- ▶ Initially, pick location invariant Q .

$$x \mapsto v * Q(v) \Rightarrow \mathbf{W}_Q(x) * \mathbf{R}_Q(x)$$

- ▶ Release write \rightsquigarrow give away permissions.

$$\{\mathbf{W}_Q(x) * Q(v)\} x_{\text{rel}} := v \{\mathbf{W}_Q(x)\}$$

- ▶ Acquire read \rightsquigarrow gain permissions.

$$\{\mathbf{R}_Q(x)\} a := x_{\text{acq}} \{\mathbf{R}_{Q[a:=\text{emp}]}(x) * Q(a)\}$$

where $Q[a:=\text{emp}] \triangleq \lambda v. \text{if } v = a \text{ then emp else } Q(v)$

Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.

$$\{x \mapsto 0 * y \mapsto 0\}$$

$x_{\text{na}} := 5;$

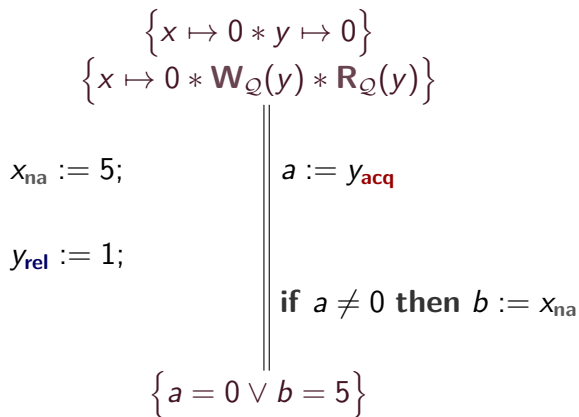
$y_{\text{rel}} := 1;$

$a := y_{\text{acq}}$

if $a \neq 0$ **then** $b := x_{\text{na}}$

$$\{a = 0 \vee b = 5\}$$

Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.



Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.

$$\begin{array}{c}
 \{x \mapsto 0 * y \mapsto 0\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y) * \mathbf{R}_Q(y)\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y)\} \parallel \{ \mathbf{R}_Q(y) \} \\
 x_{\text{na}} := 5; \qquad a := y_{\text{acq}} \\
 \\
 y_{\text{rel}} := 1; \qquad \text{if } a \neq 0 \text{ then } b := x_{\text{na}} \\
 \\
 \{a = 0 \vee b = 5\}
 \end{array}$$

Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.

$$\begin{array}{c}
 \{x \mapsto 0 * y \mapsto 0\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y) * \mathbf{R}_Q(y)\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y)\} \parallel \{ \mathbf{R}_Q(y) \} \\
 x_{\text{na}} := 5; \quad a := y_{\text{acq}} \\
 \{x \mapsto 5 * \mathbf{W}_Q(y)\} \\
 y_{\text{rel}} := 1; \\
 \text{if } a \neq 0 \text{ then } b := x_{\text{na}} \\
 \parallel \\
 \{a = 0 \vee b = 5\}
 \end{array}$$

Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.

$$\begin{array}{c}
 \{x \mapsto 0 * y \mapsto 0\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y) * \mathbf{R}_Q(y)\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y)\} \parallel \{ \mathbf{R}_Q(y) \} \\
 x_{\text{na}} := 5; \quad a := y_{\text{acq}} \\
 \{x \mapsto 5 * \mathbf{W}_Q(y)\} \\
 y_{\text{rel}} := 1; \\
 \{ \mathbf{W}_Q(y) \} \quad \text{if } a \neq 0 \text{ then } b := x_{\text{na}} \\
 \parallel \\
 \{a = 0 \vee b = 5\}
 \end{array}$$

Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.

$$\begin{array}{c}
 \{x \mapsto 0 * y \mapsto 0\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y) * \mathbf{R}_Q(y)\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y)\} \\
 x_{na} := 5; \\
 \{x \mapsto 5 * \mathbf{W}_Q(y)\} \\
 y_{rel} := 1; \\
 \{\mathbf{W}_Q(y)\} \\
 \{\top\} \\
 \{a = 0 \vee b = 5\}
 \end{array}
 \parallel
 \begin{array}{c}
 \{\mathbf{R}_Q(y)\} \\
 a := y_{acq} \\
 \text{if } a \neq 0 \text{ then } b := x_{na}
 \end{array}$$

Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.

$$\begin{array}{c}
 \{x \mapsto 0 * y \mapsto 0\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y) * \mathbf{R}_Q(y)\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y)\} \parallel \{\mathbf{R}_Q(y)\} \\
 x_{na} := 5; \quad a := y_{\mathbf{acq}} \\
 \{x \mapsto 5 * \mathbf{W}_Q(y)\} \parallel \{(a = 0 \vee x \mapsto 5) * \mathbf{R}_{Q[a:=\text{emp}]}(y)\} \\
 y_{rel} := 1; \\
 \{\mathbf{W}_Q(y)\} \quad \text{if } a \neq 0 \text{ then } b := x_{na} \\
 \{\top\} \\
 \{a = 0 \vee b = 5\}
 \end{array}$$

Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.

$$\begin{array}{c}
 \{x \mapsto 0 * y \mapsto 0\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y) * \mathbf{R}_Q(y)\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y)\} \parallel \{ \mathbf{R}_Q(y) \} \\
 x_{na} := 5; \quad a := y_{\mathbf{acq}} \\
 \{x \mapsto 5 * \mathbf{W}_Q(y)\} \parallel \{(a = 0 \vee x \mapsto 5) * \mathbf{R}_{Q[a:=\text{emp}]}(y)\} \\
 y_{rel} := 1; \quad \{a = 0 \vee x \mapsto 5\} \\
 \{ \mathbf{W}_Q(y) \} \quad \text{if } a \neq 0 \text{ then } b := x_{na} \\
 \{ \top \} \\
 \{a = 0 \vee b = 5\}
 \end{array}$$

Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.

$$\begin{array}{c}
 \{x \mapsto 0 * y \mapsto 0\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y) * \mathbf{R}_Q(y)\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y)\} \parallel \{ \mathbf{R}_Q(y) \} \\
 x_{\text{na}} := 5; \quad a := y_{\text{acq}} \\
 \{x \mapsto 5 * \mathbf{W}_Q(y)\} \parallel \{ (a = 0 \vee x \mapsto 5) * \mathbf{R}_{Q[a:=\text{emp}]}(y) \} \\
 y_{\text{rel}} := 1; \quad \{ a = 0 \vee x \mapsto 5 \} \\
 \{ \mathbf{W}_Q(y) \} \quad \text{if } a \neq 0 \text{ then } b := x_{\text{na}} \\
 \{ \top \} \quad \{ a = 0 \vee (x \mapsto 5 \wedge b = 5) \} \\
 \{ a = 0 \vee b = 5 \}
 \end{array}$$

Let $Q(v) \triangleq (v = 0 \vee x \mapsto 5)$.

$$\begin{array}{l}
 \{x \mapsto 0 * y \mapsto 0\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y) * \mathbf{R}_Q(y)\} \\
 \{x \mapsto 0 * \mathbf{W}_Q(y)\} \parallel \{\mathbf{R}_Q(y)\} \\
 x_{na} := 5; \quad a := y_{acq} \\
 \{x \mapsto 5 * \mathbf{W}_Q(y)\} \parallel \{(a = 0 \vee x \mapsto 5) * \mathbf{R}_{Q[a:=emp]}(y)\} \\
 y_{rel} := 1; \quad \{a = 0 \vee x \mapsto 5\} \\
 \{\mathbf{W}_Q(y)\} \quad \text{if } a \neq 0 \text{ then } b := x_{na} \\
 \{\top\} \quad \{a = 0 \vee (x \mapsto 5 \wedge b = 5)\} \\
 \{a = 0 \vee b = 5\}
 \end{array}$$

Ownership transfer works!

Basically, disallow ownership transfer.

- ▶ Relaxed reads:

$$\{\mathbf{R}_Q(x)\} a := x_{\text{rlx}} \{\mathbf{R}_Q(x) \wedge (Q(a) \neq \text{false})\}$$

- ▶ Relaxed writes:

$$\frac{Q(v) = \text{emp}}{\{\mathbf{W}_Q(x)\} x_{\text{rlx}} := v \{\mathbf{W}_Q(x)\}}$$

Basically, disallow ownership transfer.

- ▶ Relaxed reads:

$$\{\mathbf{R}_Q(x)\} a := x_{\text{rlx}} \{\mathbf{R}_Q(x) \wedge (Q(a) \neq \text{false})\}$$

- ▶ Relaxed writes:

$$\frac{Q(v) = \text{emp}}{\{\mathbf{W}_Q(x)\} x_{\text{rlx}} := v \{\mathbf{W}_Q(x)\}}$$

Unsound because of dependency cycles!

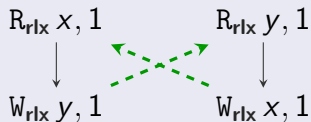
Dependency cycles

Initially $x = y = 0$.

```
a := xrlx;           b := yrlx;  
if a ≠ 0 then       if b ≠ 0 then  
    yrlx := 1      xrlx := 1
```

C11 allows the outcome $x = y = 1$.

Justification



Relaxed accesses
don't synchronize

Initially $x = y = 0$.

$$\begin{array}{l} a := x_{rlx}; \\ \text{if } a \neq 0 \text{ then} \\ \quad y_{rlx} := 1 \end{array} \parallel \begin{array}{l} b := y_{rlx}; \\ \text{if } b \neq 0 \text{ then} \\ \quad x_{rlx} := 1 \end{array}$$

C11 allows the outcome $x = y = 1$.

What goes wrong:

Non-relational invariants are unsound.

$$x = 0 \wedge y = 0$$

The DRF-property does not hold.

Dependency cycles

Initially $x = y = 0$.

$$\begin{array}{l} a := x_{rlx}; \\ \text{if } a \neq 0 \text{ then} \\ \quad y_{rlx} := 1 \end{array} \parallel \parallel \begin{array}{l} b := y_{rlx}; \\ \text{if } b \neq 0 \text{ then} \\ \quad x_{rlx} := 1 \end{array}$$

C11 allows the outcome $x = y = 1$.

A simple fix:

Strengthen the model to forbid $po \cup rf$ cycles.

A better fix /future work:

Use the “promising” model [Kang et al., POPL’17]

Incorrect message passing

Initially $x = y = 0$.

$x_{na} := 5;$

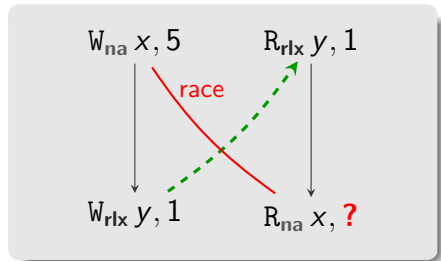
$y_{rlx} := 1$

repeat

$a := y_{rlx}$

until $a \neq 0;$

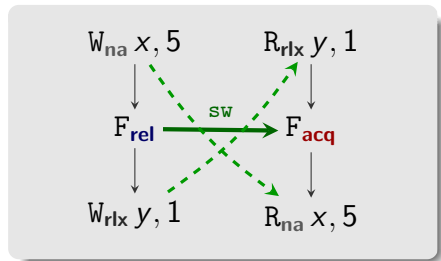
$b := x_{na}$



Message passing with C11 memory fences

Initially $x = y = 0$.

$x_{na} := 5;$		repeat
fence(rel);		$a := y_{rlx}$
$y_{rlx} := 1$		until $a \neq 0;$
		fence(acq);
		$b := x_{na}$



Introduce two 'modalities' in the logic:

- ▶ ΔP : state ready to be transferred away.
- ▶ ∇P : state that will be acquired after a **fence(acq)**.

Proof rules:

$$\{P\} \text{ fence(rel)} \{\Delta P\}$$

$$\{\mathbf{W}_Q(x) * \Delta Q(v)\} \quad x_{\text{rlx}} := v \quad \{\mathbf{W}_Q(x)\}$$

$$\{\mathbf{R}_Q(x)\} \quad t := x_{\text{rlx}} \quad \{\mathbf{R}_{Q[t:=\text{emp}]}(x) * \nabla Q(t)\}$$


$$\{\nabla P\} \text{ fence(acq)} \{P\}$$

Message passing with C11 memory fences

Let $Q(v) \triangleq v = 0 \vee x \mapsto 5$.

	$\{x \mapsto 0 * y \mapsto 0\}$	
		$\{R_Q(y)\}$
$\{x \mapsto 0 * W_Q(y)\}$		$a := y_{rlx}$
$x_{na} := 1;$		$\{\nabla(a = 0 \vee x \mapsto 5)\}$
$\{x \mapsto 5 * W_Q(y)\}$		if $a \neq 0$ then
fence (rel);		$\{\nabla(x \mapsto 5)\}$
$\{\Delta(x \mapsto 5) * W_Q(y)\}$		fence (acq)
$y_{rlx} := 1;$		$\{x \mapsto 5\}$
$\{W_Q(y)\}$		$b := x_{na}$
		$\{x \mapsto 5 \wedge b = 5\}$
		$\{a = 0 \vee (x \mapsto 5 \wedge b = 5)\}$
		$\{a = 0 \vee b = 5\}$

Three key features:

- ▶ Location protocols
- ▶ Ghost state/tokens 
- ▶ Escrows for ownership transfer

Example (Racy message passing)

Initially, $x = y = 0$.

$$\begin{array}{l}
 x_{rlx} := 1; \quad \parallel \quad x_{rlx} := 1; \quad \parallel \quad a := y_{acq}; \\
 y_{rel} := 1 \quad \parallel \quad y_{rel} := 1 \quad \parallel \quad b := x_{rlx}
 \end{array}$$

Cannot get $a = 1 \wedge b = 0$.

Racy message passing in GPS

Protocol for x : **A**: $x = 0$ \longrightarrow **B**: $x = 1$

Protocol for y : **C**: $y = 0$ \longrightarrow **D**: $y = 1 \wedge x.st \geq \mathbf{B}$

Acquire reads gain knowledge, not ownership.

$$\left\{ \begin{array}{l} \{x.st \geq \mathbf{A} \wedge y.st \geq \mathbf{C}\} \\ x_{rlx} := 1; \\ \{x.st \geq \mathbf{B} \wedge y.st \geq \mathbf{C}\} \\ y_{rel} := 1 \\ \{x.st \geq \mathbf{B} \wedge y.st \geq \mathbf{D}\} \end{array} \right\} \parallel \left\{ \begin{array}{l} \{x.st \geq \mathbf{A} \wedge y.st \geq \mathbf{C}\} \\ a := y_{acq}; \\ \left\{ \begin{array}{l} a = 0 \wedge x.st \geq \mathbf{A} \\ \vee a = 1 \wedge x.st \geq \mathbf{B} \end{array} \right\} \\ b := x_{rlx}; \\ \{a = 0 \vee (a = 1 \wedge b = 1)\} \end{array} \right\}$$