An axiomatic basis for computer programming …on the relaxed Arm-A architecture: the AxSL logic

Angus Hammond*, Zongyuan Liu†,
Thibaut Pérami*, Peter Sewell*,
Lars Birkedal†, Jean Pichon-Pharabod†

*University of Cambridge, †Aarhus University

January 18, 2024

Motivation

- We want to reason about programs in weak memory settings, like Arm-A

Motivation

- We want to reason about programs in weak memory settings, like Arm-A
- We have an authoritative model for user-mode Arm-A

Motivation

- We want to reason about programs in weak memory settings, like Arm-A
- We have an authoritative model for user-mode Arm-A
- But we want to reason about programs in a compositional way

Motivation

- We want to reason about programs in weak memory settings, like Arm-A
- We have an authoritative model for user-mode Arm-A
- But we want to reason about programs in a compositional way
- And the Arm-A model is very global

Why it is hard: Load Buffering

$$
\begin{tikzcd}
\text{a: } r_1 := \mathrm{ldr}\ [x] \arrow[d, "\mathrm{po}"'] & \text{c: } r_2 := \mathrm{ldr}\ [y] \arrow[d, "\mathrm{po}"'] \\
\text{b: } \mathrm{str}\ [y]\ 1 \arrow[ur, "\mathrm{rf}"'] & \text{d: } \mathrm{str}\ [x]\ 1 \arrow[ul, "\mathrm{rf}"]
\end{tikzcd}
$$

From initial state $x = y = 0$, final state $r_1 = r_2 = 1$ is allowed.

Incompatability of simple logics and LB

Expect both:

Incompatability of simple logics and LB

Expect both:

$$\{P * r_1 \overset{\mathrm{r}}{\mapsto} v\}$$
$$r_1 := \mathsf{ldr}\ [x]$$
$$\{P * r_1 \overset{\mathrm{r}}{\mapsto} v'\}$$

Resources passed between program points

Incompatability of simple logics and LB

Expect both:

$\{P * r_1 \overset{\mathrm{r}}{\mapsto} v\}$
$r_1 := \mathsf{ldr}\ [x]$
$\{P * r_1 \overset{\mathrm{r}}{\mapsto} v'\}$

Resources passed between program points

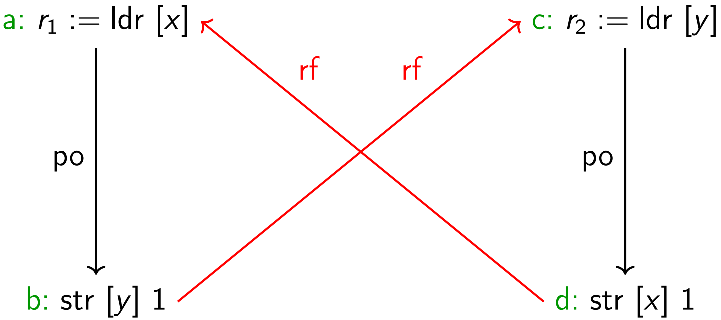
$\{P\}$
$\mathsf{str}\ [x]\ 1$
$\{\top\}$

rf

$\{\top\}$
$r_1 := \mathsf{ldr}\ [x]$
$\{P\}$

Resources passed from writes to reads

Incompatability of simple logics and LB



Logics for RC11 don't suffer this issue as RC11 has $(\mathsf{po} \cup \mathsf{rf})$ acyclic.

Existing WM models

- Operational models
  - Naturally operational
  - Explicit speculation and instruction rewinding

Existing WM models

- Operational models
  - Naturally operational
  - Explicit speculation and instruction rewinding
- Promising models
  - Fairly operational
  - LB requires tricky to reason about certification step

Existing WM models

- Operational models
  - Naturally operational
  - Explicit speculation and instruction rewinding
- Promising models
  - Fairly operational
  - LB requires tricky to reason about certification step
- Axiomatic models
  - Succinct and straightforward to formalise
  - Not at all operational

Global nature of axiomatic semantics

- Axiomatic models are so global because the consistency check is done after program execution completes

Global nature of axiomatic semantics

- Axiomatic models are so global because the consistency check is done after program execution completes

- We would like to use the information from the consistency check incrementally as the program executes

Global nature of axiomatic semantics

- Axiomatic models are so global because the consistency check is done after program execution completes

- We would like to use the information from the consistency check incrementally as the program executes

- But we cannot easily check consistency of partial executions, because an execution could be made inconsistent by later events

Opax Semantics

- In the Opax semantics, we instead guess a consistent whole program memory event graph before beginning execution

Opax Semantics

- In the Opax semantics, we instead guess a consistent whole program memory event graph before beginning execution

- Then we incrementally check the guessed graph matches program behaviour

Opax Semantics

- Initialisation
  $e \longrightarrow \langle G, \emptyset, e\rangle$ where G is a memory event graph consistent with the axiomatic model

Opax Semantics

- Initialisation
  $e \longrightarrow \langle G, \emptyset, e\rangle$ where G is a memory event graph consistent with the axiomatic model

- Matching an event
  $\langle G, F, r := \mathrm{ldr}\ [a] :: e\rangle \longrightarrow \langle G, F \cup \{\mathrm{R}\ a\ v\}, e\rangle$ where $\mathrm{R}\ a\ v \in G \setminus F$

Opax Semantics

- Initialisation
  $e \longrightarrow \langle G, \emptyset, e \rangle$ where G is a memory event graph consistent with the axiomatic model

- Matching an event
  $\langle G, F, r := \mathsf{ldr}\ [a] :: e \rangle \longrightarrow \langle G, F \cup \{\mathsf{R}\ a\ v\}, e \rangle$ where $\mathsf{R}\ a\ v \in G \setminus F$

Opax Semantics

- Initialisation
  $e \longrightarrow \langle G, \emptyset, e\rangle$ where G is a memory event graph consistent with the axiomatic model

- Matching an event
  $\langle G, F, r := \mathrm{ldr}\ [a] :: e\rangle \longrightarrow \langle G, F \cup \{\mathrm{R}\ a\ v\}, e\rangle$ where $\mathrm{R}\ a\ v \in G \setminus F$

Overview of AxSL

- Iris program logic built above Opax semantics
- thread-modular reasoning about memory event graphs

Overview of AxSL

- Iris program logic built above Opax semantics
- thread-modular reasoning about memory event graphs

$$P, Q \in iProp ::= \text{(Iris connectives)} \;\cdots \quad | \quad r \overset{\mathrm{r}}{\mapsto} v @ a \quad | \quad a \looparrowright P \quad | \quad \{P\}\, e\, \{Q\}_{\Phi} \quad | \quad \cdots$$

Register value v comes from event a

Overview of AxSL

- Iris program logic built above Opax semantics
- thread-modular reasoning about memory event graphs

$$P, Q \in iProp ::= \text{(Iris connectives)} \;\cdots \quad | \quad r \overset{\mathrm{r}}{\mapsto} v@a \quad | \quad a \looparrowright P \quad | \quad \{P\}e\{Q\}_{\Phi} \quad | \quad \cdots$$

Register value v comes from event a

P is tied to event a

Overview of AxSL

- Iris program logic built above Opax semantics
- thread-modular reasoning about memory event graphs

$$P, Q \in iProp ::= (\text{Iris connectives}) \;\cdots \quad |$$

$$\underbrace{r \overset{\mathrm{r}}{\mapsto} v @ a \quad | \quad a \hookrightarrow P}_{\text{sound resource passing along po}} \quad | \quad \{P\}\, e\, \{Q\}_{\Phi} \quad | \quad \cdots$$

Overview of AxSL

- Iris program logic built above Opax semantics
- thread-modular reasoning about memory event graphs

$$
\begin{aligned}
P, Q \in iProp ::= {} & \text{(Iris connectives)} \;\cdots \quad | \\
& r \overset{\mathrm{r}}{\mapsto} v @ a \quad | \quad a \hookrightarrow P \quad | \quad \underbrace{\{P\}\, e\, \{Q\}_{\Phi}}_{\text{modular reasoning}} \quad | \quad \cdots
\end{aligned}
$$

Overview of AxSL

- Iris program logic built above Opax semantics
- thread-modular reasoning about memory event graphs

$$
\begin{aligned}
P, Q \in iProp ::= {} & \text{(Iris connectives)} \;\cdots \quad | \\
& r \overset{\mathrm{r}}{\mapsto} v@a \quad | \quad a \hookrightarrow P \quad | \quad \{P\}\, e\, \{Q\}_{\Phi} \quad | \quad \cdots
\end{aligned}
$$

Hoare triple with
per-location protocol
$\Phi \in \mathsf{addr} \to \mathsf{val} \to \mathsf{eid} \to iProp$

Proving MP in AxSL

a: str $[data]$ 42

b: str_{rel} $[flag]$ 1

c: r_1 := ldr $[flag]$

d: r_2 := ldr $[data + r_1 - r_1]$

Proving MP in AxSL

$$\begin{array}{l}
\textcolor{green}{\mathsf{a}}\text{: } \mathsf{str}\ [\mathit{data}]\ 42 \\
\qquad\quad \big\downarrow \ \mathsf{po}; [\mathsf{Rel}] \subseteq \mathsf{ob} \\
\textcolor{green}{\mathsf{b}}\text{: } \mathsf{str}_{\mathsf{rel}}\ [\mathit{flag}]\ 1
\end{array}
\qquad\qquad
\begin{array}{l}
\textcolor{green}{\mathsf{c}}\text{: } r_1 := \mathsf{ldr}\ [\mathit{flag}] \\
\qquad \mathsf{addr} \subseteq \mathsf{ob}\ \big\downarrow \\
\textcolor{green}{\mathsf{d}}\text{: } r_2 := \mathsf{ldr}\ [\mathit{data} + r_1 - r_1]
\end{array}$$

Proving MP in AxSL

$\{\ \top\ \}$
a: str [*data*] 42

b: $\mathrm{str}_{\mathrm{rel}}$ [*flag*] 1

$\{\ \cdots\ \}$

$\{r_1 \overset{\mathrm{r}}{\mapsto} \_ * r_2 \overset{\mathrm{r}}{\mapsto} \_\ \}$
c: $r_1 := \mathrm{ldr}\ [\mathit{flag}]$

d: $r_2 := \mathrm{ldr}\ [\mathit{data} + r_1 - r_1]$

$$\left\{ \begin{array}{l} r_1 \overset{\mathrm{r}}{\mapsto} v_{\mathit{flag}} @ \_ * r_2 \overset{\mathrm{r}}{\mapsto} v_{\mathit{data}} @ \_ * \\ \quad (v_{\mathit{flag}} = 1 \Rightarrow v_{\mathit{data}} = 42) \end{array} \right\}$$

Proving MP in AxSL

$\{\top\}$

a:

The protocol Φ:

$$\Phi(\mathit{data}, v, e) \triangleq \mathrm{Initial}(e) \lor v = 42$$

$$\Phi(\mathit{flag}, v, e) \triangleq \mathrm{Initial}(e) \lor \left(v = 1 * \exists e'.\, e'{:}\mathrm{W}\ \mathit{data}\ 42 \xrightarrow{\mathrm{po}} e{:}\mathrm{W}_{\mathrm{rel}}\ \mathit{flag}\ 1\right)$$

b:

$\{$

$(v_{flag} = 1 \Rightarrow v_{data} = 42)\}$

Proving MP in AxSL

$$\Phi(data, v, e) \triangleq \mathrm{Initial}(e) \lor v = 42$$

$$\Phi(flag, v, e) \triangleq \mathrm{Initial}(e) \lor \left(v = 1 * \exists e'.\ e'{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\mathrm{po}} e{:}\mathrm{W_{rel}}\ \ flag\ 1\right)$$

$\{\ \top\ \}$

a: str $[data]$ 42

b: $\mathrm{str_{rel}}$ $[flag]$ 1

$\{\ \cdots\ \}$

$\{ r_1 \overset{\mathrm{r}}{\mapsto} \_ * r_2 \overset{\mathrm{r}}{\mapsto} \_ \}$

c: $r_1 :=$ ldr $[flag]$

d: $r_2 :=$ ldr $[data + r_1 - r_1]$

$$\left\{ \begin{array}{c} r_1 \overset{\mathrm{r}}{\mapsto} v_{flag} @ \_ * r_2 \overset{\mathrm{r}}{\mapsto} v_{data} @ \_ * \\ (v_{flag} = 1 \Rightarrow v_{data} = 42) \end{array} \right\}$$

Proving MP in AxSL

$$\Phi(data, v, e) \triangleq \mathrm{Initial}(e) \vee v = 42$$

$$\Phi(flag, v, e) \triangleq \mathrm{Initial}(e) \vee \left(v = 1 * \exists e'.\ e'{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} e{:}\mathrm{W}_{\text{rel}}\ \ flag\ 1\right)$$

$\{\ \top\ \}$

a: str $[data]$ 42

$\{\ a{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} \cdot\}$

b: $\mathrm{str}_{\text{rel}}$ $[flag]$ 1

Proof obligation $\Phi(data, 42, a)$

$\{\ \cdots\ \}$

$\{r_1 \overset{\text{r}}{\mapsto} \_ * r_2 \overset{\text{r}}{\mapsto} \_\ \}$

c: $r_1 :=$ ldr $[flag]$

d: $r_2 :=$ ldr $[data + r_1 - r_1]$

$$\left\{\begin{array}{c} r_1 \overset{\text{r}}{\mapsto} v_{flag}@\_ * r_2 \overset{\text{r}}{\mapsto} v_{data}@\_ * \\ (v_{flag} = 1 \Rightarrow v_{data} = 42) \end{array}\right\}$$

Proving MP in AxSL

$$\Phi(data, v, e) \triangleq \mathrm{Initial}(e) \vee v = 42$$

$$\Phi(flag, v, e) \triangleq \mathrm{Initial}(e) \vee \left(v = 1 * \exists e'.\ e'{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} e{:}\mathrm{W}_{\text{rel}}\ \ flag\ 1\right)$$

$\{\ \top\ \}$

a: str $[data]$ 42

$\{\ a{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} \cdot\}$

b: str_{rel} $[flag]$ 1

$\left\{\begin{array}{l} a{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} b{:}\mathrm{W}_{\text{rel}}\ \ flag\ 1* \\ \dots \end{array}\right\}$

$\{\ \dots\ \}$

Proof obligation

$\Phi(flag, 1, b)$

$\{r_1 \overset{\text{r}}{\mapsto} \_ * r_2 \overset{\text{r}}{\mapsto} \_\ \}$

c: $r_1 :=$ ldr $[flag]$

d: $r_2 :=$ ldr $[data + r_1 - r_1]$

$\left\{\begin{array}{l} r_1 \overset{\text{r}}{\mapsto} v_{flag} @ \_ * r_2 \overset{\text{r}}{\mapsto} v_{data} @ \_ * \\ \qquad (v_{flag} = 1 \Rightarrow v_{data} = 42) \end{array}\right\}$

Proving MP in AxSL

$$\Phi(data, v, e) \triangleq \mathrm{Initial}(e) \lor v = 42$$

$$\Phi(flag, v, e) \triangleq \mathrm{Initial}(e) \lor \left(v = 1 * \exists e'.\ e'{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} e{:}\mathrm{W_{rel}}\ \ flag\ 1\right)$$

$\{\ \top\ \}$

a: str $[data]$ 42

$\{\ a{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} \cdot\}$

b: $\mathrm{str_{rel}}$ $[flag]$ 1

$\left\{\begin{array}{l} a{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} b{:}\mathrm{W_{rel}}\ \ flag\ 1 * \\ \cdots \end{array}\right\}$

$\{\ \cdots\ \}$

$\{r_1 \overset{\mathrm{r}}{\mapsto} \_ * r_2 \overset{\mathrm{r}}{\mapsto} \_\ \}$

c: r_1 := ldr $[flag]$

$\left\{\begin{array}{l} c{:}\mathrm{R}\ \ flag\ v_{flag} \xrightarrow{\text{po}} \cdot * \\ r_1 \overset{\mathrm{r}}{\mapsto} v_{flag} @ c * c \looparrowright \Phi(flag, v_{flag}, c) * \cdots \end{array}\right\}$

d: r_2 := ldr $[data + r_1 - r_1]$

$\left\{\begin{array}{c} r_1 \overset{\mathrm{r}}{\mapsto} v_{flag} @ \_ * r_2 \overset{\mathrm{r}}{\mapsto} v_{data} @ \_ * \\ (v_{flag} = 1 \Rightarrow v_{data} = 42) \end{array}\right\}$

Proving MP in AxSL

$$\Phi(data, v, e) \triangleq \mathrm{Initial}(e) \vee v = 42$$

$$\Phi(flag, v, e) \triangleq \mathrm{Initial}(e) \vee \left(v = 1 * \exists e'.\ e'{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\mathrm{po}} e{:}\mathrm{W_{rel}}\ \ flag\ 1\right)$$

$\{\ \top\ \}$

a: str $[data]$ 42

$\{\ a{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\mathrm{po}} \cdot\}$

b: $\mathrm{str_{rel}}$ $[flag]$ 1

$\left\{\begin{array}{l} a{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\mathrm{po}} b{:}\mathrm{W_{rel}}\ \ flag\ 1 * \\ \cdots \end{array}\right\}$

$\{\ \cdots\ \}$

$\{r_1 \overset{\mathrm{r}}{\mapsto} \_ * r_2 \overset{\mathrm{r}}{\mapsto} \_\ \}$

c: $r_1 :=$ ldr $[flag]$

$\left\{\begin{array}{l} c{:}\mathrm{R}\ \ flag\ v_{flag} \xrightarrow{\mathrm{po}} \cdot * \\ r_1 \overset{\mathrm{r}}{\mapsto} v_{flag} @ c * c \hookrightarrow \Phi(flag, v_{flag}, c) * \cdots \end{array}\right\}$

d: $r_2 :=$ ldr $[data + r_1 - r_1]$

$\left\{\begin{array}{l} r_1 \overset{\mathrm{r}}{\mapsto} v_{flag} @ \_ * r_2 \overset{\mathrm{r}}{\mapsto} v_{data} @ \_ * \\ \qquad (v_{flag} = 1 \Rightarrow v_{data} = 42) \end{array}\right\}$

Proving MP in AxSL

$$\Phi(data, v, e) \triangleq \mathsf{Initial}(e) \vee v = 42$$

$$\Phi(flag, v, e) \triangleq \mathsf{Initial}(e) \vee \big(v = 1 * \exists e'.\ e'{:}\mathsf{W}\ \ data\ 42 \xrightarrow{\text{po}} e{:}\mathsf{W}_{\mathsf{rel}}\ \ flag\ \ 1\big)$$

$\{\ \top\ \}$

a: str $[data]$ 42

$\{\ a{:}\mathsf{W}\ \ data\ 42 \xrightarrow{\text{po}} \cdot\}$

b: $\mathsf{str}_{\mathsf{rel}}$ $[flag]$ 1

$\left\{\begin{array}{l} a{:}\mathsf{W}\ \ data\ 42 \xrightarrow{\text{po}} b{:}\mathsf{W}_{\mathsf{rel}}\ \ flag\ \ 1 * \\ \cdots \end{array}\right\}$

$\{\ \cdots\ \}$

$\{r_1 \overset{\text{r}}{\mapsto} \_ * r_2 \overset{\text{r}}{\mapsto} \_\ \}$

c: $r_1 :=$ ldr $[flag]$

$\left\{\begin{array}{l} c{:}\mathsf{R}\ \ flag\ \ v_{flag} \xrightarrow{\text{po}} \cdot * \\ r_1 \overset{\text{r}}{\mapsto} v_{flag}@c * c \hookrightarrow \Phi(flag, v_{flag}, c) * \cdots \end{array}\right\}$

d: $r_2 :=$ ldr $[data + r_1 - r_1]$

$\left\{\begin{array}{l} c{:}\mathsf{R}\ \ flag\ \ v_{flag} \xrightarrow{\text{addr}} d{:}\mathsf{R}\ \ data\ \ v_{data} * \\ r_1 \overset{\text{r}}{\mapsto} v_{flag}@c * r_2 \overset{\text{r}}{\mapsto} v_{data}@d * c \hookrightarrow \top * \\ d \hookrightarrow (\Phi(data, v_{data}, d) * \Phi(flag, v_{flag}, c)) \end{array}\right\}$

$\left\{\begin{array}{l} r_1 \overset{\text{r}}{\mapsto} v_{flag}@\_ * r_2 \overset{\text{r}}{\mapsto} v_{data}@\_ * \\ \qquad (v_{flag} = 1 \Rightarrow v_{data} = 42) \end{array}\right\}$

Proving MP in AxSL

$$\Phi(data, v, e) \triangleq \mathrm{Initial}(e) \vee v = 42$$

$$\Phi(flag, v, e) \triangleq \mathrm{Initial}(e) \vee \left(v = 1 * \exists e'.\ e'{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} e{:}\mathrm{W_{rel}}\ \ flag\ 1\right)$$

$\{\ \top\ \}$

a: str [*data*] 42

$\{\ a{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} \cdot\}$

b: $\mathrm{str_{rel}}$ [*flag*] 1

$\left\{\begin{array}{l} a{:}\mathrm{W}\ \ data\ 42 \xrightarrow{\text{po}} b{:}\mathrm{W_{rel}}\ \ flag\ 1 * \\ \cdots \end{array}\right\}$

$\{\ \cdots\ \}$

$\{r_1 \overset{\text{r}}{\mapsto} \_ * r_2 \overset{\text{r}}{\mapsto} \_\ \}$

c: $r_1 :=$ ldr [*flag*]

$\left\{\begin{array}{l} c{:}\mathrm{R}\ \ flag\ v_{flag} \xrightarrow{\text{po}} \cdot * \\ r_1 \overset{\text{r}}{\mapsto} v_{flag} @ c * c \hookrightarrow \Phi(flag, v_{flag}, c) * \cdots \end{array}\right\}$

d: $r_2 :=$ ldr [$data + r_1 - r_1$]

$\left\{\begin{array}{l} c{:}\mathrm{R}\ \ flag\ v_{flag} \xrightarrow{\text{addr}} d{:}\mathrm{R}\ \ data\ v_{data} * \\ r_1 \overset{\text{r}}{\mapsto} v_{flag} @ c * r_2 \overset{\text{r}}{\mapsto} v_{data} @ d * c \hookrightarrow \top * \\ d \hookrightarrow (\Phi(data, v_{data}, d) * \Phi(flag, v_{flag}, c)) \end{array}\right\}$

$\left\{\begin{array}{l} r_1 \overset{\text{r}}{\mapsto} v_{flag} @ \_ * r_2 \overset{\text{r}}{\mapsto} v_{data} @ \_ * \\ d \hookrightarrow (v_{flag} = 1 \Rightarrow v_{data} = 42) \end{array}\right\}$

Soundness (Adequacy theorem) of AxSL

- Soundness proof is challenging
  - tension between reasoning along program order and induction along ob

Soundness (Adequacy theorem) of AxSL

- Soundness proof is challenging
  - tension between reasoning along program order and induction along ob
- AxSL has an adequacy theorem
  - results proven in AxSL also hold at the meta level w.r.t. the (axiomatic-model-based) Opax semantics

Soundness (Adequacy theorem) of AxSL

- Soundness proof is challenging
  - tension between reasoning along program order and induction along ob
- AxSL has an adequacy theorem
  - results proven in AxSL also hold at the meta level w.r.t. the (axiomatic-model-based) Opax semantics
- The statement is similar to stardard Iris adequacy, but the proof is novel
  - by **stratification**: two traversals over program executions

Conclusion

- AxSL is an expressive program logic for (user-mode) Arm-A memory model, that
  - supports thread-local reasoning and many advanced CSL features
  - is proven sound w.r.t. the axiomatic-model-based Opax semantics (first in Iris)
  - is fully mechanised in Coq

Conclusion

- AxSL is an expressive program logic for (user-mode) Arm-A memory model, that
  - supports thread-local reasoning and many advanced CSL features
  - is proven sound w.r.t. the axiomatic-model-based Opax semantics (first in Iris)
  - is fully mechanised in Coq
- Main limitations
  - Lacking support for coherence
  - Missing many abstractions

Conclusion

- AxSL is an expressive program logic for (user-mode) Arm-A memory model, that
  - supports thread-local reasoning and many advanced CSL features
  - is proven sound w.r.t. the axiomatic-model-based Opax semantics (first in Iris)
  - is fully mechanised in Coq
- Main limitations
  - Lacking support for coherence
  - Missing many abstractions
- Our approach will generalise
  - The Opax semantics can be adapted for other axiomatic memory models
  - The resource-tied-to assertions will allow sound reasoning above other very relaxed MMs, e.g. RISC-V

AD: If you like beautiful interactive robots...

Check out **Glowbot Garden** @ St Mary le Strand Church (3 min away! 12noon-8pm)