Concurrent separation logic and operational semantics

Viktor Vafeiadis

MPI-SWS

What is the paper about?

Soundness proof for CSL

— Simple

— Extensible
  Permissions
  RGSep
  Storable locks [Buisse, Birkedal, Støvring, MFPS 2011]
  Concurrent abstract predicates [Dinsdale-Young et al., ECOOP 2010]

— Explains precision & conjunction rule

— Fully mechanized in Isabelle/HOL

Hoare triples (partial correctness)

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

$$\forall s\ h\ s'\ h'.\ \ s,h \models P \land (C, s, h) \rightarrow^* (\mathbf{skip}, s', h') \implies s',h' \models Q$$

Standard operational semantics

Judgment form: $(C, s, h) \rightarrow (C', s', h')$

Rules for seq. composition:

$$(\mathbf{skip};C, s, h) \rightarrow (C, s, h)$$

$$\frac{(C_1, s, h) \rightarrow (C_1', s', h')}{(C_1;C_2, s, h) \rightarrow (C_1';C_2, s', h')}$$

Or equivalently...

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

$$\forall s\ h.\ s,h \models P \implies \underbrace{\forall s'\ h'.\ (C, s, h) \to^{*} (\textbf{skip}, s', h') \implies s',h' \models Q}_{\text{safe}(C,s,h,Q)}$$

$$\Updownarrow$$

$$\forall s'\ h'.\ \forall m.\quad (C, s, h) \to^{m} (\textbf{skip}, s', h') \implies s',h' \models Q$$

$$\Updownarrow$$

$$\forall n.\ \underbrace{\forall s'\ h'.\ \forall m < n.\ (C, s, h) \to^{m} (\textbf{skip}, s', h') \implies s',h' \models Q}_{\text{safe}_n(C,s,h,Q)}$$

As an inductive definition...

$\vDash \{P\}\ C\ \{Q\}$ iff $\forall s\ h\ n.\ \ s,h \vDash P \Longrightarrow \mathrm{safe}_n\ (C, s, h, Q)$

$$\mathrm{safe}_0(C, s, h, Q) = \ \mathrm{true}$$
$$\mathrm{safe}_{n+1}(C, s, h, Q) =$$
$$(C = \textbf{skip} \ \Longrightarrow s,h \vDash Q)$$
$$\wedge\ (\forall C'\ s'\ h'.\ (C, s, h) \rightarrow (C', s', h') \Longrightarrow \mathrm{safe}_n(C', s', h', Q))$$

$$\underbrace{\forall s'\ h'.\ \forall m < n.\ \ (C, s, h) \rightarrow^m (\textbf{skip}, s', h') \ \Longrightarrow\ s',h' \vDash Q}_{\mathrm{safe}_n(C,s,h,Q)}$$

Fault-avoidance

$\vDash \{P\}\ C\ \{Q\}$ iff $\forall s\ h\ n.\ \ s,h \vDash P \Longrightarrow \mathit{safe}_n\ (C, s, h, Q)$

$\mathit{safe}_0(C, s, h, Q) = \text{true}$

$\mathit{safe}_{n+1}(C, s, h, Q) =$

$(C = \textbf{skip} \Longrightarrow s,h \vDash Q)$

$\wedge\ (\neg\ (C, s, h) \to \textbf{abort})$

$\wedge\ (\forall C'\ s'\ h'.\ (C, s, h) \to (C', s', h')$

$\Longrightarrow \mathit{safe}_n(C', s', h', Q))$

— “Well-specified programs don’t go wrong”

"Bake in" the frame rule

$\models \{P\}\ C\ \{Q\}$ iff $\forall s\ h\ n.\ \ s,h \models P \Longrightarrow \text{safe}_n\ (C, s, h, Q)$

$\text{safe}_0(C, s, h, Q) = \text{true}$

$\text{safe}_{n+1}(C, s, h, Q) =$

$(C = \textbf{skip} \Longrightarrow s,h \models Q)$

$\land\ (\forall h_F.\ \neg\ (C, s, h{+}h_F) \rightarrow \textbf{abort})$

$\land\ (\forall h_F\ C'\ s'\ h'.\ (C, s, h{+}h_F) \rightarrow (C', s', h')$

$\Longrightarrow \exists h''.\ h' = h''{+}h_F \land \text{safe}_n(C', s', h'', Q))$

— No safety monotonicity & frame property

— Same definition works for permissions
(every permission-heap can be extended to a normal heap)

Atomic blocks

C ::= ... | **atomic** C

Semantics:

$$\frac{(C, s, h) \rightarrow^* (\textbf{skip}, s', h')}{(\textbf{atomic}\ C, s, h) \rightarrow (\textbf{skip}, s', h')}$$

$$\frac{(C, s, h) \rightarrow^* \textbf{abort}}{(\textbf{atomic}\ C, s, h) \rightarrow \textbf{abort}}$$

$$\frac{\vdash \{ P * J \}\ C\ \{ Q * J \}}{J \vdash \{ P \}\ \textbf{atomic}\ C\ \{ Q \}}$$

$$\frac{J * R \vdash \{ P \}\ C\ \{ Q \}}{J \vdash \{ P * R \}\ C\ \{ Q * R \}}$$

$$\frac{J \vdash \{ P_1 \}\ C_1\ \{ Q_1 \} \qquad J \vdash \{ P_2 \}\ C_2\ \{ Q_2 \}}{J \vdash \{ P_1 * P_2 \}\ C_1 \parallel C_2\ \{ Q_1 * Q_2 \}}$$

Atomic blocks

$J \vDash \{P\}\ C\ \{Q\}$ iff $\forall s\ h\ n.\ \ s,h \vDash P \Longrightarrow \mathrm{safe}_n\ (C,\ s,\ h,\ J,\ Q)$

$\mathrm{safe}_0(C,\ s,\ h,\ J,\ Q) =$ true

$\mathrm{safe}_{n+1}(C,\ s,\ h,\ J,\ Q) =$

$(C = \textbf{skip} \Longrightarrow s,h \vDash Q)$

$\wedge\ (\forall h_J\ h_F.\ s,h_J \vDash J \Longrightarrow \neg\ (C,\ s,\ h{+}h_J{+}h_F) \rightarrow \textbf{abort})$

$\wedge\ (\forall h_J\ h_F\ C'\ s'\ h'.\ (C,\ s,\ h{+}h_J{+}h_F) \rightarrow (C',\ s',\ h') \wedge s,h_J \vDash J$

$\Longrightarrow \exists h''\ h_J'.\ h' = h''{+}h_J'{+}h_F \wedge s',h_J' \vDash J \wedge \mathrm{safe}_n(C',\ s',\ h'',\ J,\ Q))$

No races

$J \models \{P\}\ C\ \{Q\}$ iff $\forall s\ h\ n.\ \ s,h \models P \Longrightarrow \mathrm{safe}_n\ (C, s, h, J, Q)$

$\mathrm{safe}_0(C, s, h, J, Q) = \mathrm{true}$

$\mathrm{safe}_{n+1}(C, s, h, J, Q) =$

$(C = \mathbf{skip} \Longrightarrow s,h \models Q)$

$\wedge\ (\forall h_J\ h_F.\ s,h_J \models J \Longrightarrow \neg\ (C, s, h+h_J+h_F) \to \mathbf{abort})$

$\wedge\ \mathrm{accesses}(C, s) \subseteq \mathrm{dom}(h)$

$\wedge\ (\forall h_J\ h_F\ C'\ s'\ h'.\ (C, s, h+h_J+h_F) \to (C', s', h') \wedge s,h_J \models J$

$\Longrightarrow \exists h''\ h_J'.\ h' = h''+h_J'+h_F \wedge s',h_J' \models J \wedge \mathrm{safe}_n(C', s', h'', J, Q))$

Multiple resources

C ::= ... | **resource** r **in** C | **with** r **when** B **do** C | **within** r **do** C

Semantics (Extract)

$$\frac{B(s)}{(\textbf{with}\ r\ \textbf{when}\ B\ \textbf{do}\ C, s, h) \to (\textbf{within}\ r\ \textbf{do}\ C, s, h)}$$

$$\frac{(C, s, h) \to (C', s', h') \qquad r \notin L(C)}{(\textbf{within}\ r\ \textbf{do}\ C, s, h) \to (\textbf{within}\ r\ \textbf{do}\ C, s', h')}$$

$$\frac{}{(\textbf{within}\ r\ \textbf{do}\ \textbf{skip}, s, h) \to (\textbf{skip}, s, h)}$$

L(C) : set of locks currently acquired by C

$$\frac{\Gamma \vdash \{ (P * J) \wedge B \}\ C\ \{ Q * J \}}{\Gamma, r : J \vdash \{ P \}\ \textbf{with}\ r\ \textbf{when}\ B\ \textbf{do}\ C\ \{ Q \}}$$

$$\frac{\Gamma, r : J \vdash \{ P \}\ C\ \{ Q \}}{\Gamma \vdash \{ P * J \}\ \textbf{resource}\ r\ \textbf{in}\ C\ \{ Q * J \}}$$

Multiple resources

$\Gamma \models \{P\}\ C\ \{Q\}$ iff $\forall s\ h\ n.\ \ s,h \models P \Longrightarrow \text{safe}_n\ (C, s, h, \Gamma, Q)$

$\text{safe}_0(C, s, h, \Gamma, Q) = \text{true}$

$\text{safe}_{n+1}(C, s, h, \Gamma, Q) =$

$(C = \textbf{skip} \Longrightarrow s,h \models Q)$

$\wedge\ (\forall h_F.\ \neg\ (C, s, h+h_F) \rightarrow \textbf{abort})$

$\wedge\ \text{accesses}(C, s) \subseteq \text{dom}(h)$

locks acquired

$\wedge\ (\forall h_\Gamma\ h_F\ C'\ s'\ h'.\ (C, s, h+h_\Gamma+h_F) \rightarrow (C', s', h') \wedge s,h_\Gamma \models \circledast_{r \in L(C') \setminus L(C)} \Gamma(r)$

$\Longrightarrow \exists h''\ h_\Gamma'.\ h'=h''+h_\Gamma'+h_F \wedge s',h_\Gamma' \models \circledast_{r \in L(C) \setminus L(C')} \Gamma(r)$

$\wedge\ \text{safe}_n(C', s', h'', \Gamma, Q))$

locks released

L(C) : set of locks currently acquired by C

What is the paper about?

Soundness proof for CSL

— Simple

— Extensible
  Permissions
  RGSep
  Storable locks [Buisse, Birkedal, Støvring, MFPS 2011]
  Concurrent abstract predicates [Dinsdale-Young et al., ECOOP 2010]

— Explains precision & conjunction rule

— Fully mechanized in Isabelle/HOL

Precision & the conjunction rule

Prove: $\mathrm{safe}_n(C,s,h,\Gamma,Q_1) \wedge \mathrm{safe}_n(C,s,h,\Gamma,Q_2) \Longrightarrow \mathrm{safe}_n(C,s,h,\Gamma,Q_1 \wedge Q_2)$

$\mathrm{safe}_{n+1}(C, s, h, \Gamma, Q) =$ [...]
$\wedge\ (\forall h_\Gamma\ h_F\ C'\ s'\ h'.$ [...]
$\Longrightarrow \exists h''\ h_\Gamma'.\ h'=h''+h_\Gamma'+h_F \wedge s',h_\Gamma' \vDash \circledast_{r \in L(C) \setminus L(C')} \Gamma(r)$
$\wedge\ \mathrm{safe}_n(C, s', h'', \Gamma, Q))$

$\exists h''_1\ h_\Gamma'_1.\ \ h'=h''_1+h_\Gamma'_1+h_F \wedge s',h_\Gamma'_1 \vDash \circledast_{r \in L(C) \setminus L(C')} \Gamma(r) \wedge \mathrm{safe}_n(C,s',h''_1,\Gamma,Q_1)$

$\exists h''_2\ h_\Gamma'_2.\ \ h'=h''_2+h_\Gamma'_2+h_F \wedge s',h_\Gamma'_2 \vDash \circledast_{r \in L(C) \setminus L(C')} \Gamma(r) \wedge \mathrm{safe}_n(C,s',h''_2,\Gamma,Q_2)$

Definition. P precise iff $\forall s\ h_1\ h_2\ h'_1\ h'_2.$
$h_1+h'_1= h_2+h'_2 \wedge s,h_1 \vDash P \wedge s,h_1 \vDash P \Longrightarrow h_1=h_2 \wedge h'_1=h'_2$