Introduction to Separation Logic

Lectures at MGS'18

Georg Struth

on action short of strike at University of Sheffield, UK

Lecture 4: Verification Conditions

This Lecture

- verification conditions
- Reynold's local and global laws for basic commands
- extended Hoare logic with frame rule
- extended refinement calculus with framing law
- from a relational fault model to weakest liberal preconditions

Overview

- so far: we derived PHL/PRC for separation logic
- including frame rule/framing law
- it remains to derive Hore-style laws/rules for basic commands

Verification Conditions

Hoare triple

in any quantale Q and for $\varphi : Q \to Q$

$$H\, x\, \varphi\, y \Leftrightarrow x \leq \varphi\ y$$

remarks

- compatible with backward conjunctive transformers $[-]$
- dually, specification $\varphi\, x \leq y$ compatible with forward disjunctive transformers $\langle - \rangle$
- isotone transformers cover both cases

Verification Conditions

remember

rules of PHL with frame rule derivable

$$\{p\}\,[\![\text{skip}]\!]\,\{p\}$$

$$p \le p' \wedge q' \le q \ \wedge \{p'\}\,[\![C]\!]\,\{q'\} \Rightarrow \{p\}\,[\![C]\!]\,\{q\}$$

$$\{p\}\,[\![C_1]\!]\,\{r\} \ \wedge \{r\}\,[\![C_2]\!]\,\{q\} \Rightarrow \{p\}\,[\![C_1; C_2]\!]\,\{q\}$$

$$\{p \sqcap b\}\,[\![C_1]\!]\,\{q\} \wedge \{p \sqcap -b\}\,[\![C_2]\!]\,\{q\} \Rightarrow \{p\}\,[\![\text{if } b \text{ then } C_1 \text{ else } C_2 \text{ fi}]\!]\,\{q\}$$

$$\{p \sqcap b\}\,[\![C]\!]\,\{p\} \Rightarrow \{p\}\,[\![\text{while } b \text{ do } C \text{ od}]\!]\,\{-b \sqcap p\}$$

$$[\![C]\!]\ p\text{-local} \wedge \{q\}\,[\![C]\!]\,\{r\} \Rightarrow \{p * q\}\,[\![C]\!]\,\{p * r\}$$

one inference rule per programming construct!

Verification Conditions

while-rule with invariant

$$p \subseteq i \wedge i \sqcup -t \subseteq q \wedge i \sqcap t \subseteq [\![C]\!] i \Rightarrow p \subseteq [\![\text{while } t \text{ inv } i \text{ do } C \text{ od}]\!] q$$

Verification Conditions

useful predicates

- empty heaplet: $\mathsf{emp}\, s \Leftrightarrow \eta_s = \varepsilon$
- singleton heaplet: $(l \mapsto e)\, s \Leftrightarrow dom\, \eta_s = \{l\, \sigma_s\} \wedge \eta_s\, (l\, \sigma_s) = e\, \sigma_s$
- $l \mapsto -$ indicates any value at location l: $l \mapsto - \Leftrightarrow \bigsqcup_{e \in E} l \mapsto e$

example

- $[\![x := y]\!]$ is not local with respect to $x \mapsto 2$
  - $\{y \mapsto 1\}\ [\![x := y]\!]\ \{x \mapsto 1\}$ is obviously true
  - frame rule implies $\{(y \mapsto 1) * (x \mapsto 2)\}\ [\![x := y]\!]\ \{(x \mapsto 1) * (x \mapsto 2)\}$
  - this reduces to $\{(y \mapsto 1) * (x \mapsto 2)\}\ [\![x := y]\!]\ \{0\}$, which is false
- non-parametric locality excludes using $[\![x := y]\!]$ in frame rule
- yet command is p-local for $p = z \mapsto 1$ if $z \neq x$

Local Assignment Axioms

lemma

Reynolds' local assignment axioms are derivable in p-local isotone predicate transformer semantics over $\mathcal{P}\, S_S$

- store assignment: $q[e/x] = [\![x := e]\!]\, q$
- heap lookup: for x and m distinct

$$(l \mapsto e) \sqcap (x = m) \leq [\![x := *l]\!]\, ((l[m/x] \mapsto e) \sqcap (x = e))$$

- heap mutation: $(x \mapsto -) \leq [\![*x := e]\!]\, (x \mapsto e)$
- heap allocation: for x not free in e

$$\mathsf{emp} \leq [\![x := \mathsf{alloc}\, e]\!]\, (x \mapsto e)$$

- heap deallocation: $(e \mapsto -) = [\![\mathsf{dispose}\, e]\!]\, \mathsf{emp}$

these laws translate immediately into Hoare-style assignment axioms

Global Assignment Axioms

lemma

Reynolds' global assignment axioms are derivable in the p-local isotone predicate transformer semantics over $\mathcal{P}\, S_S$

- store assignment: for x not free in p

$$\{q[e/x] * p\}\ [\![x := e]\!]\ \{q * p\}$$

- heap lookup: for x not free in p

$$\{(l \mapsto e) * (x = m) * p\}\ [\![x := *l]\!]\ \{(l[m/x] \mapsto e) * (x = e)\rangle * p\}$$

- heap mutation: $\{(x \mapsto -) * p)\}\ [\![*x := e]\!]\ \{(x \mapsto e) * p\}$
- heap allocation: for x not free in e and p

$$\{p\}\ [\![x := \mathsf{alloc}\, e]\!]\ \{(x \mapsto e) * p\}$$

- heap deallocation: $\{(l \mapsto -) * p\}\ [\![\mathsf{dispose}\, l]\!]\ \{p\}$

Reynolds' lookup rule looks slightly different

Backward Assignment Axioms

lemma

Reynolds' backward allocation axioms are derivable in the p-local isotone predicate transformer semantics over $\mathcal{P}\, S_S$

- heap lookup: if y neither free in l nor in p unless it is x

$$\exists y.\ (l \mapsto y) * ((l \mapsto y) \mathrel{-\!\!*} p[y/x]) \leq [\![x := *l]\!] p$$

- heap mutation: $(l \mapsto -) * ((l \mapsto e) \mathrel{-\!\!*} p) \leq [\![*l := e]\!] p$
- heap allocation: if $y \neq x$ and not free in e, p

$$\forall y.\ (y \mapsto e) \mathrel{-\!\!*} p[y/x] \leq [\![v := \mathsf{alloc}\, e]\!] p$$

remark

existential quantifier is defined by poitwise lifting:

$$\exists x.\ p\ x = \lambda s.\ \exists x.\ p\ x\ s = \lambda s.\ \bigsqcup_x p\, x\, s$$

Backward Assignment Axioms

- backward laws work for any postcondition p
- only lookup, mutation and allocation need to be derived
- local store assignment and global heap disposal law are backward

Verification Condition Generation

- assignment laws and structural rules of PHL allows programming tactics for vcg
- in Isabelle one can use Eisbach language
- tactic computes weakest liberal precondition for given program
- it then remains to check whether it contains precondition of program
- resulting proof obligations are at data level
- frame rule is used like a weakening law to insert appropriate contexts
- more concretely in terms of predicates that hold on separate heaplets

Refinement Laws

Morgan's specification statement

$$[x, y]_p = \bigsqcap \{\varphi \mid x \leq \varphi\, y \wedge \varphi \text{ } p\text{-local isotone}\}$$

lemma

if φ is p-local and isotone, then

1. $[x, y]_p \leq \varphi \Leftrightarrow x \leq \varphi\, y$
2. $\{x\}\, [x, y]_p\, \{y\}$
3. $\{x\}\, \varphi\, \{y\} \Rightarrow [x, y]_p \leq \varphi$

lemma

$[x, y]_p$ is p-local isotone

Refinement Laws

lemma

rules of PRC with framing law derivable in space of p-local isotone transformers

$$x \leq x' \wedge y \leq y' \Rightarrow [x, y]_p \leq [x', y']_p$$
$$[x, y]_p \leq [x, z]_p \circ [z, y]_p$$
$$[\bigsqcap X, y]_p \leq \bigsqcap \{[x, y]_p \mid x \in X\}$$
$$[x, x]_p \leq [x, x]_p^\omega$$
$$[x, y]_p \leq [p] \circ [p \sqcap x, y]_p \sqcap [q] \circ [q \sqcap x, y]_p$$
$$[x, q \sqcap x]_p \leq ([p] \circ [x \sqcap p, , x])^\omega \circ [q]$$
$$[p * x, p * y]_p \leq [x, y]_p$$

Refinement Laws for Store Assignment

lemma

the following refinement laws for store assignment are derivable in the predicate transformer semantics over $\mathcal{P}\, S_S$, assuming x not free in p

$$q \leq r[e/x] \Rightarrow [q, r]_p \leq [\![x := e]\!]$$
$$r' \leq r[e/x] \Rightarrow [q, r]_p \leq [q, r']_p \circ [\![x := e]\!]$$
$$q' \leq q[e/x] \Rightarrow [q.r]_p \leq [\![x := e]\!] \circ [q', r]_p$$

Refinement Laws for Heap Assignments

lemma

the following refinement laws are derivable in the predicate transformer semantics over $\mathcal{P}\, S_S$, assuming x, y and z are distinct

$$[(x = y) \wedge (e \mapsto z), (x = z) \wedge (e[y/x] \mapsto z)]_p \leq [\![x := *e]\!] \quad \text{if } x \text{ free in } p$$

$$[e \mapsto -, e \mapsto e']_p \leq [\![*e := e']\!]$$

$$[\mathsf{emp}, x \mapsto e]_p \leq [\![\mathsf{alloc}\, e]\!] \quad \text{if } x \text{ not free in } e,\ r$$

$$[e \mapsto -, \mathsf{emp}]_p \leq [\![\mathsf{dispose}\, e]\!]$$

Example

Justification of Approach

- predicate transformer semantics is simpler than usual state transformer semantics
  - backward analysis starts from non-faulting state
  - no faults are generated during analysis
  - faults are due to partiality of heaplet operations
- we now construct the predicate transformer semantics from a relational semantics with faults

Relational Fault Semantics

idea

- consider RAM model as semigroup $S_\perp$ with zero adjoined to PAM S
- programs that may fault are relations $S \times S_\perp$ on semigroup $S_\perp$
- pairs $(s, s') \in S \times S_\perp$ model
  - successful program execution if $s' \neq \perp$
  - faulting program execution if $s' = \perp$
- tests and assertions in partial correctness specs range over S
  - assertions prescribe correctness conditions, but don't observe faults
  - tests observe statelets, but do not modify heaps

Relational Fault Semantics

redefining composition

- partition $R \subseteq S \times S_\perp$ into
  - proper part $R_p \subseteq S \times S$
  - faulting part $R_f \subseteq S \times \{\perp\}$
- represent R as (R_p, R_f)
- redefine $R; S = R_f \cup R_p; S$
- that is $(R_p, R_f); (S_p, S_f) = (R_p; S_p, R_f \cup R_p; S_f)$
- faults thus override compositions

Algebraic Justification

definition

a quantale module of quantale Q and complete lattice L is action $\circ : Q \to L \to L$ that satisfies

$$(p \cdot q) \circ x = p \circ (q \circ x)$$

$$(\bigsqcup P) \circ x = \bigsqcup_{p \in P} p \circ x$$

$$p \circ \bigsqcup X = \bigsqcup_{x \in X} p \circ X$$

$$1 \circ x = x$$

Algebraic Justification

definition

semidirect product $Q \ltimes L$ on $Q \times L$ is defined as

$$(p, x) \ltimes (q, y) = (p \cdot q, x \sqcup p \circ y)$$

remarks

- semidirect products are standard tools in (semi)group theory
- relational definition of composition above is simple instance
- $(1, 0)$ is unit of $\ltimes$

Algebraic Justification

theorem

if Q is a quantale and L a complete lattice, then $Q \ltimes L$ forms weak quantale with left distributivity law weakened to

$$X \neq \emptyset \Rightarrow p \ltimes \bigsqcup X = \bigsqcup_{x \in X} p \ltimes x$$

and lattice order/operations defined as for $Q \times L$

Relational Fault Semantics

corollary

$\mathcal{P}(S \times S_\perp)$ is weak quantale with composition $R; S = R_f \cup R_p; S$ and unit Id.

proof

- proper relations form unital quantales with relational unit Id
- faulting parts of relations form complete lattices □

remarks

- predicates in this weak quantale are just relational subidentities: relations below $(Id, \emptyset)$ in $\mathcal{P}(S \times S_\perp)$
- all predicates are proper by this definition
- they form a boolean subalgebra
- and a boolean quantale with respect to separating conjunction

The Star in Weak Quantales

definition

for $(p, x)^0 = (1, 0)$ and $(p, x)^{i+1} = (p, x) \ltimes (p, x)^i$ we define on $Q \ltimes L$

$$(p, x)^* = \bigsqcup_{i \in \mathbb{N}} (p, x)^i$$

lemma

if Q is quantale and L complete lattice, then

$$(p, x)^* = (p^*, p^* \circ x)$$

one can then define Hoare logic on weak quantales

Hoare Triples

definition

$$\{p\}\, R\, \{q\} \Leftrightarrow \lceil p \rceil ; R \subseteq (R; \lceil q \rceil)_p$$

remarks

- partial correctness specification $\{p\}\, R\, \{q\}$ holds if every execution of R from p is contained in executions of R after which q holds
- projection on proper part guarantees absence of faults
- if program R faults from precondition p, then Hoare triple is false

Locality for Relations

definition

for relation $R \subseteq S \times S_\perp$ and predicate $p \subseteq S$

- R is p-safe if

$$(s, \perp) \notin R \wedge p\, s' \Rightarrow (s \oplus s', \perp) \notin R$$

- R satisfy the p-frame property if

$$(s, s') \in R \wedge s = s_1 \oplus s_2 \wedge (s_1, \perp) \notin R \wedge p\, s_2 \Rightarrow \\ \exists s_1'.\ s' = s_1' \oplus s_2 \wedge (s_1, s_1') \in R$$

Locality for Relations

lemma

$\cup$, $\cap$, $;$ and rt-closure * preserve p-safety and p-frame property

lemma

relations in $\mathcal{P}(S \times S_\perp)$ that are p-safe and satisfy the p-frame property for a given $p \in \mathcal{P}\, S$ form weak sub-quantale of $\mathcal{P}(S \times S_\perp)$

State Transformers

definition

for $R \subseteq S \times S_\perp$ we define state transformer $f_R : S \to (\mathcal{P}\ S_\perp)^\top$ defined by

$$f_R\, s = \begin{cases} \{s' \mid (s, s') \in R\} & \text{if } (s, \perp) \notin R \\ \top & \text{otherwise} \end{cases}$$

remarks

- in range of f_R, $\top$ is adjoined to $\mathcal{P}\, S_\perp$
- hence $p \subseteq_\top \top$ for all $p \in \mathcal{P}\,(S_\perp)^\top$
- separating conjunction is extended to $*_\top$ by $p *_\top \top = \top = \top *_\top p$
- composing state transformers by Kleisli composition requires explaining what $\langle f_R \rangle \top$ or $\langle R \rangle \top$ should be
- $\langle f_R \rangle \top = \top$ according to the standard semantics

State Transformers

lemma

if $R \subseteq S \times S_\perp$ and $p, q \in \mathcal{P}\, S$, then $\langle R \rangle p \subseteq_\top q \Leftrightarrow \lceil p \rceil ; R \subseteq (R ; \lceil q \rceil)_p$

remarks

- if fault occurs in state s, then $\langle f_R \rangle (p\, s) = \top$
- this is greater than any postcondition

this explains why faults must be propagated to $\top$ in assertion quantale

Locality for State Transformers

definition

state transformer f is p-local with respect to predicate $p \in \mathcal{P}\, S$ if for all states s and s'

$$p\, s' \Rightarrow f\ (s \oplus s') \subseteq_\top f\ s *_\top p.$$

lemma

if $R \subseteq S \times S_\perp$ and $p \subseteq S$, then f_R is p-local if and only if R is p-safe and satisfies the p-frame property.

Predicate Transformer Semantics

definition

for all $p, q \in \mathcal{P}\, S$ and $f_R : S \to (\mathcal{P}\, S_\perp)^\top$

$$[f_R]q = \{x \mid f_R\, x \subseteq_\top q\},$$

lemma

Let $R \subseteq R \times R_\perp$, $f_R : S \to (\mathcal{P}\, S_\perp)^\top$ and $p \in \mathcal{P}\, S$. Then $[R] = [R_p]$.

remarks

- It follows that $p \subseteq [R]q \Leftrightarrow p \subseteq [R_p]q$
- this justifies using predicate transformers of type $\mathcal{P}\, S \to \mathcal{P}\, S$
  - if f_R generates failure states from each input
  - then $[f_R]q = \emptyset$ for any q
  - falsifies the $p \subseteq [f_R]q$ for any $p \neq \emptyset$, as expected

Predicate Transformer Semantics

lemma
if $R \subseteq S \times S_\perp$ and $p, q \in \mathcal{P}\, S$, then $p \subseteq [R_p]q \Leftrightarrow \lceil p \rceil ; R \subseteq (R ; \lceil q \rceil)_p$

lemma
if $R \subseteq S \times S_\perp$ and $p \subseteq S$, then f_R is p-local iff $[R_p]$ is

Predicate Transformer Semantics

theorem
if $R \subseteq S \times S_\perp$ and $p, q \subseteq S$, then

$$\langle R \rangle p \subseteq_\top q \iff \{p\}\, R\, \{q\} \iff p \subseteq [R_p]q$$
$$\Updownarrow$$
$$\lceil p \rceil ; R \subseteq (R ; \lceil q \rceil)_p$$

theorem
if $R \subseteq S \times S_\perp$ and $p \subseteq S$, then the following are equivalent

1. R is p-safe and satisfies the p-frame property
2. f_R is p-local
3. $[R_p]$ is p-local

Semantics in Context

- backward predicate transformer is simplest semantics
- mathematically and with respect to verification condition generation
- does not require explicit reasoning with fault elements
- backward reasoning automatically excludes all possible dangling pointers and faulty states

Exercises

?

Further Reading

- Calcagno et al, *Local Action and Abstract Separation Logic*
- Reynolds, *Separation Logic: A Logic for Shared Mutable Data Structures*
- Dongol, Gomes, Struth, *A Program Construction and Verification Tool for Separation Logic*
- Dongol, Hayes, Struth, *Relational Convolution, Generalised Modalities and Incidence Algebras*
- Abramsky, Vickers, *Quantales, Observational Logics and Process Semantics*
- Clifford, Preston, *The Algebraic Theory of Semigroups*