RGSep/SAGL

Local Heaps

(Joint work with Viktor Vafeiadis)

(SAGL: Xinyu Feng, Zhong Shao, …)

Shared and Local assertions

We extend our assertion language:

$$p, q \ ::= p * q \mid \exists x. p \mid p \vee q \mid P \mid \boxed{P}$$

Global heap

Local heap

Assertions

Global heap

$$s, l, h \vDash p \ * q$$

Local heap

$$\Leftrightarrow \exists l_1, l_2.\ s, l_1, h \vDash p$$
$$\wedge\, s, l_2, h \vDash q \ \wedge \ l_1 * l_2 = l$$

What do the following mean?

$$x \mapsto 5 \ * \ y \mapsto 5$$

$$x \mapsto 5 \ * \ x \mapsto 5$$

$$x \mapsto 5 \ * \ x \mapsto 5$$

$$x \mapsto 5 \vee emp \quad * \ x \mapsto 5$$

Axioms

$$P * P \Leftrightarrow P \wedge P$$

$$\frac{P * P' \Longrightarrow \text{false}}{P * P' \Longrightarrow \text{false}}$$

Relations

We use

$$P \to Q$$

to express heap changes.

Semantically means:

$$\left\{ \begin{matrix} (s, h * h''), \\ (s, h' * h'') \end{matrix} \middle| \begin{matrix} (s, h) \vDash P \wedge \\ (s, h') \vDash Q \end{matrix} \right\}$$

Anything not mentioned is unchanged

Stable

p $stable\ wrt\ R$

$$\Leftrightarrow \forall s, l, h, s', h'.$$

$$s, l, h \vDash p \wedge \ (s, h), (s', h') \in R$$

$$\Longrightarrow s', l, h' \vDash p$$

Ownership transfer

What is the meaning of:

A) $x \mapsto 0 \ * y \mapsto \_ \ \rightarrow \ \ x \mapsto 1$

B) $x \mapsto 0 \ * y \mapsto \_ \ \rightarrow \ \ x \mapsto 0$

C) $x \mapsto 0 \ \ \rightarrow \ \ x \mapsto 1 \ * y \mapsto 0$

D) $x \mapsto i \ \ \rightarrow \ \ x \mapsto i + 1$

Examples

Given interference

A) $x \mapsto 0 \ * y \mapsto \_ \rightarrow \ x \mapsto 1$

B) $x \mapsto 0 \ * y \mapsto \_ \rightarrow \ x \mapsto 0$

C) $x \mapsto 0 \rightarrow \ x \mapsto 1 \ * y \mapsto 0$

are the following assertions stable

1) $x \mapsto 0 \ * y \mapsto \_$

2) $x \mapsto 0 \ * true \ * y \mapsto \_$

3) $x \mapsto 0 \ * true$

Proof rule for parallel

$$\frac{\begin{array}{c} R_1, G_1 \vdash \{p_1\}\, C_1\, \{q_1\} \\ R_2, G_2 \vdash \{p_2\}\, C_2\, \{q_2\} \\ mods(C_1) \cap mods(C_2) = \emptyset \\ G_1 \subseteq R_2 \\ G_2 \subseteq R_1 \end{array}}{R_1 \cup R_2, G_1 \cap G_2 \vdash \{p_1 * p_2\} C_1 \,||\, C_2 \{q_1 * q_2\}}$$

Proof rule for atomic commands

$$\frac{\begin{array}{c}(P \to \exists \bar{x}.\, Q) \subseteq G \\ \vdash_{SL} \{P * P'\}\, C\, \{Q * Q'\} \\ mods(C) = \bar{x}\end{array}}{R, G \vdash \{\boxed{P} * P'\} < C > \{\boxed{Q} * Q'\}}$$

Proof rule for atomic commands

$$\frac{\begin{array}{c}(P \to \exists \bar{x}.\, Q) \subseteq G \\ \vdash_{SL} \{P * P'\}\, C\, \{Q * Q'\} \\ mods(C) = \bar{x}\end{array}}{R, G \vdash \{P * F * P'\} < C > \{Q * F * Q'\}}$$

Proof rule for atomic commands

$$\frac{\begin{array}{c}(P \to \exists \bar{x}.\, Q) \subseteq G \\ \vdash_{SL} \{P * P'\}\, C\, \{Q * Q'\} \\ mods(C) = \bar{x} \\ P'' \Rightarrow P * F \\ Q * F \Rightarrow Q''\end{array}}{R, G \vdash \{\boxed{P''} * P'\} < C > \{\boxed{Q''} * Q'\}}$$

Examples

```
    while(b) { <b := 0==[x]>; };  [y] := 1
||
    [y] := 1; < [x] := 1 >
```

Examples

```
lock() {
        b = true
        while(b) { b := CAS(x,0,1); }
}
unlock() {
        [x] := 0
}
```

Exercise

Append only list

```
n := new();
n.tl := null;
x := head;
while(!CAS(x.tl, null, n))
    while(x.tl != null)
        x := x.tl
```

Verify this keeps the shared state looking like a list.

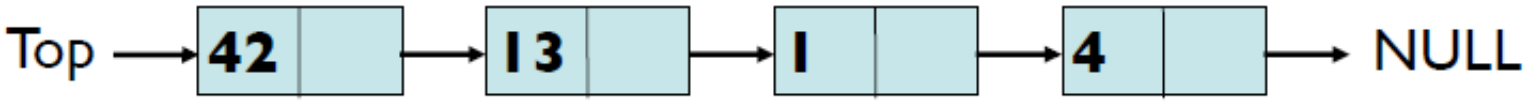
Non-blocking stack (Treiber)

```
struct Node {
  Node *next;
  data_t val;
} *Top;
```

```
void push(data_t v) {
  Node *t, *x;
  x = new Node;
  x->val = v;
  do {
    t = Top;
    x->next = t;
  } while(!CAS(&Top,t,x));
}
```

```
data_t pop() {
  Node *t, *x;
  do {
    t = Top;
    if (t == NULL)
      return EMPTY;
    x = t->next;
  } while(!CAS(&Top,t,x));
  return t->val;
}
```

Treat Top as
in the heap.



Proof

Push:

$$Top \mapsto x \;\; \rightarrow \;\; \mathrm{Top} \mapsto y \;\; * \, y \mapsto x$$

Pop:

$$Top \mapsto y \;\; * \, y \mapsto x \;\; \rightarrow \;\; \mathrm{Top} \mapsto x \;\; * \, y \mapsto x$$

Stable assertion

$$\exists x. Top \mapsto x \;\; *$$
$$\exists \, y. ls(\mathrm{t}, y) * ls(x, y) * ls(y, 0) * true$$
$$\vee \, ls(x, \mathrm{t}) * ls(\mathrm{t}, 0)$$

Proof

Push:

$$Top \mapsto x \quad \rightarrow \quad \text{Top} \mapsto y \ * y \mapsto x$$

Pop:

$$Top \mapsto y \ * y \mapsto x \quad \rightarrow \quad \text{Top} \mapsto x \ * y \mapsto x$$

Stable assertion

$$\exists x. Top \mapsto x \ * ls(x, 0)$$