A9.1 Gelle - $\vdash_{SL} \{A\}\ com\ \{B\}$ (a)

- $modifies(com) \cap free(R,G) = \emptyset$ (b)

Zeige: $com : (A, R, G, B)$

1) $modifies(com) \cap free(R,G) = \emptyset$ ✓ nach (a)

2) $\neg locked(com)$ ✓ nach (a), da SL keine Abstraktion für variablen hat

3) Seien s, h_L, h_S, n bel. mit $[\![A]\!](s, h_L, h_S)$ $\hookrightarrow [\![A]\!](s\ h_L) = 1 \wedge h_L \perp h_S$ (c)

Zeige: $sake_n(com, s, h_L, h_S, R, G, B)$

Falls $n = 0$, nichts zu zeigen ✓

Also $n > 0$.

3i) Falls $com = skip$, so liefert (a)+(c): $[\![B]\!](s\ h_L) = 1$

Nach Def RGSep dann: $[\![B]\!](s, h_L, h_S) = 1$ ✓

3ii) Sei h_F bel. mit $h_L \uplus h_S \uplus h_F$ definiert.

Es gilt nach (a)+(c): $(com, s, h_L) \not\to$ abort.

Lemma Yang & O'Hearn (W) liefert: $(com, s, h_L \uplus h_S \uplus h_F) \not\to$ abort ✓

3 iii) Seien c', s', h_L', h_F bel. mit $(com, s, h_L \uplus h_S \uplus h_F) \to (c', s', h')$.

Nach (a)+(c): $(com, s, h_L) \not\to$ abort.

Dann liefert Lemma Yang & O'Hearn (2) ein h_L' mit

$(com, s, h_L) \to (c', s', h_L')$ und $h' = h_L' \uplus h_S \uplus h_F$ ✓

Ferner gilt $(s, h_S, h_S) \in [\![G]\!]$, da $[\![G]\!]$ nach Definition die reflexive Hülle enthält.

Wissen auch: $c' = skip$. Also:

$(com, s, h_L) \to (skip, s', h_L') \to (s', h_L')$

Zusammen mit (a) erhalten wir $[\![B]\!](s', h_L')$.

Dann gilt $sat_{n-1}(skip, s', h_L', h_S, R, G, B)$ nach

> Hilfslemma: Falls $[\![B]\!](s', h_L') = 1$, so gilt
> für alle n, h_S, R, G, B : $sat_n(skip, s', h_L', h_S, R, G, B)$

↳ Beweis analog

3 iv) Sei h_S' bel. mit $(s, h_S, h_S') \in [\![R]\!]$ und $h_L \perp h_S'$.

Es gilt $sat_{n-1}(com, s, h_L, h_S', R, G, B)$,

daraus wiederhole Argument 3) ⇝ Technisch: Induktion über n mit h_S allquantifiziert.

∎

A9.2

```
void add(int e) {
  Node* x,y,z; int t;
```

$\{ \exists A.\ ls(g\_head, A, nil) * sort(A) \wedge -\infty < e \}$

```
  (x,z) = locate(e);
```

$\{ \exists u,v.\ \exists A,B,q.\ ls(g\_head,A,x) * L(x,u,z) * N_{-}(z,v,q) * ls(q,B,nil) * sort(A.u.v.B) \wedge u<e \wedge e \le v \wedge -\infty < e \}$

(z→value = v)

```
  atomic { t := z->value; } if (t != e) {
```

$\{ \exists u,v.\ \exists A,B,q.\ ls(g\_head,A,x) * L(x,u,z) * N_{-}(z,v,q) * ls(q,B,nil) * sort(A.u.v.B) \wedge u<e \wedge e<v \wedge -\infty < e \}$

```
    y := cons(e, c_unlock, z);
```

$\{ \exists u,v.\ \exists A,B,q.\ ls(g\_head,A,x) * L(x,u,z) * N_{-}(z,v,q) * ls(q,B,nil) * sort(A.u.v.B) * U(y,e,z) \wedge u<e \wedge e<v \wedge -\infty < e \}$

~~x->next = y;~~ atomic { x->next := y }

$\{ \exists u,v.\ \exists A,B,q.\ ls(g\_head,A,x) * L(x,u,y) * N_{-}(z,v,q) * ls(q,B,nil) * sort(A.u.e.v.B) * U(y,e,z) \wedge u<e \wedge e<v \wedge -\infty < e \}$

$\{ \exists u.v.\ \exists A,B,q.\ ls(g\_head,A,x) * L(x,u,y) * N_{-}(y,e,q) * ls(q,B,nil) * sort(A.u.e.B) \wedge -\infty < e \}$

} // endif

$$\left\{ \left[\exists u.v. \left[\exists A,B,q.\ ls(g\_head, A, x) * L(x,u,z) * \mathcal{N}\_(z,v,q) * ls(q,B,nil) * sort(A.u.v.B) \right] \wedge u < e \wedge e \leq u \wedge -\infty < e \right] \right.$$

$$\left. \vee \left[\exists u,v. \left[\exists A,B,q.\ ls(g\_head, A, x) * L(x,u,y) * \mathcal{N}\_(y,e,q) * ls(q,B,nil) * sort(A.u.e.B) \right] \wedge -\infty < e \right] \right\}$$

$$\left\{ \exists u. \left[\exists A,B,q.\ ls(g\_head, A, x) * L(x,u,q) * ls(q,B,nil) * sort(A.u.B) \right] \wedge -\infty < e \right\}$$

unlock(x);

$$\left\{ \exists u. \left[\exists A,B,q.\ ls(g\_head, A, x) * \mathcal{N}\_(x,u,q) * ls(q,B,nil) * sort(A.u.B) \right] \wedge -\infty < e \right\}$$

$$\left\{ \left[\exists A.\ ls(g\_head, A, nil) * sort(A) \right] \wedge -\infty < e \right\}$$

A9.3

Betrachte History H:

H = ⟨A, s.push(1)⟩ ⟨A, s:-⟩

⟨A, s.push(2)⟩ ⟨A, s:-⟩ ①

⟨B, s.pop()⟩ ②

⟨A, s.pop()⟩ ⟨A, s:1,2⟩

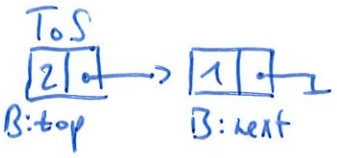
⟨A, s.push(3)⟩ ⟨A, s:-⟩

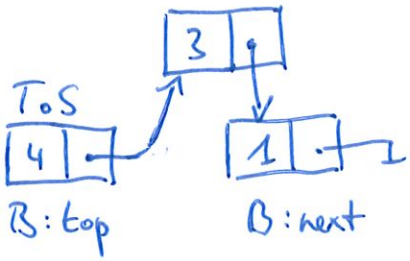
⟨A, s.push(4)⟩ ⟨A, s:-⟩ ③

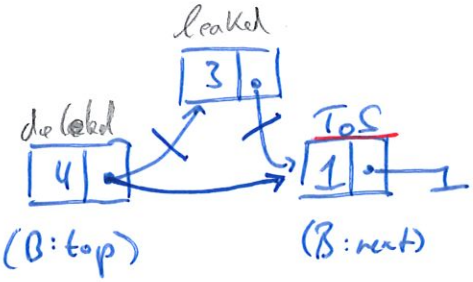
⟨B, s:1, 4⟩ ④ // Return-Wert i.O.

⟨A, s.pop()⟩ ⟨A, s:1, 1⟩

① 

② 

③ 

④ 

⇝ Stack-Inhalt wird auf Grund des ABA-Problems verloren.