8.5 Semantik des CSL-Judgments

Ziel: Definiere die Semantik von $J \vdash \{A\}\, c\, \{B\}$ über Prädikate rasen.

Idee: Wenn der initiale Zustand A erfüllt, gilt für bis zu n Schritte folgende Aussage (0)

- das Programm abortet nicht, (1)
- erhält J und (2)
- erreicht B, sollte es terminieren. (3)

Vorteil: Mit der Definition der Semantik über rasen können wir zum Beweis der Soundness Induktion nutzen

Definition (Hilfsprädikat):

$$\text{sat}^{\mathcal{U}}(s, h, c, J) := \underbrace{(\text{locked}(c) \wedge h = \emptyset)}_{\text{leer, wenn gelocked}} \vee \underbrace{(\neg \text{locked}(c) \wedge [\![J]\!]\, s\, h)}_{\text{erfüllt } J \text{ sonst.}}$$

Definition (rasen):

$$\text{safe}_0(c, s, h, J, B) := \text{true}$$

$$\text{safe}_{n+1}(c, s, h, J, B) :=$$

$$c = \text{skip} \Rightarrow [\![B]\!]\, s\, h \quad \text{// (3)}$$

$$\wedge\ \forall h_J\, h_F.\ \text{sat}^{\mathcal{U}}(s, h_J, c, J) \wedge (c, s, h \uplus h_J \uplus h_F) \longrightarrow \text{abort} \quad \text{// (1), egal welche Konfig}$$

$$\wedge\ \forall h_J, h_F, c', s', h'.\ \text{sat}^{\mathcal{U}}(s, h_J, c, J) \wedge (c, s, h \uplus h_J \uplus h_F) \to (c', s', h')$$
$$\Rightarrow \exists h'', h_J'.\ h' = h'' \uplus h_J' \uplus h_F \wedge \text{sat}^{\mathcal{U}}(s', h_J', c', J) \quad \text{// (2)}$$
$$\wedge\ \text{safe}_n(c', s', h'', J, B). \quad \text{// (0)}$$

Zu den Heaps:

h = Teil des Heaps, der vom Thread geowned wird.

h_J = Teil des Heaps, der zwischen den Threads geshared wird und daher J erfüllen muss.

h_F = Übriger Teil des Heaps, der dem Rest des Systems gehört.

Definition (Semantik der CSL-Judgments):

$J \vdash \{P\}\, c\, \{Q\}$ gilt, falls $\forall n, s, h.\ [\![R]\!]\, s\, h \Rightarrow \text{safe}(n, s, h, J, R)$.

8.6 Eigenschaften der Semantik

Ziel: Ähnlich zum Soundness-Beweis der Frame-Rule in SL benötigen wir auch für den Soundness-Beweis von CSL Eigenschaften der operationellen Semantik von Programmen.

Lemma:

Falls $(c, s, h) \to (c', s', h')$, dann

- $\text{free}(c') \subseteq \text{free}(c)$
- $\text{modifies}(c') \subseteq \text{modifies}(c)$
- $s \sim_X s'$ mit $X = \overline{\text{modifies}(c)}$.

Dabei bedeutet $s \sim_X s'$, dass $s(x) = s'(x)$ f.a. $x \in X$.

Das folgende Lemma wird über einfache Induktionen nach dem Aufbau von arithmetischen / Booleschen Ausdrücken, Prädikaten und Programmen gezeigt.

Lemma:

(1) Falls $s \sim_X s'$ mit $X = free(a)$, dann $[\![a]\!]s = [\![a]\!]s'$.

(2) Falls $s \sim_X s'$ mit $X = free(b)$, dann $[\![b]\!]s = [\![b]\!]s'$.

(3) Falls $s \sim_X s'$ mit $X = free(A)$, dann $[\![A]\!]sh = [\![A]\!]s'h$.

(4) Falls $s \sim_X s'$ mit $X = free(c)$ und $(c, s, h) \to$ abort,
dann $(c, s', h) \to$ abort.

(5) Falls $s \sim_X s'$ mit $free(c) \subseteq X$ und $(c, s, h) \to (c_1, s_1, h_1)$,
dann $(c, s', h) \to (c_1, s_1', h_1)$ mit $s_1 \sim_X s_1'$.

Als Folgerung aus diesem Lemma erhalten wir,
dass safe nur von den Variablen abhängt,
die in c, $\mathcal{J}$ und $\mathcal{B}$ genannt sind.

Lemma:

Falls $safe(c, s, h, \mathcal{J}, \mathcal{B})$ und $s \sim_X s'$ mit $X = free(c, \mathcal{J}, \mathcal{B})$,
dann $safe(c, s', h, \mathcal{J}, \mathcal{B})$.

Eine weitere Beobachtung ist, dass Safety für n-Schritte
Safety für n-Schritte impliziert, für alle $m \leq n$.

Lemma:

Falls $safe(c, s, h, \mathcal{J}, \mathcal{B})$ und $m \leq n$, dann $safe(c, n, h, \mathcal{J}, \mathcal{B})$.

8.7 Beweis der Soundness

Satz (Soundness von CSL):

Die Beweisregeln von CSL sind sound.
Für jede Regel $\frac{P_1, \ldots, P_n}{C}$ gilt also

wenn $P_1, \dots, P_n$ gültige CL-Judgments sind,
dann auch C.

Beweis:

Betrachte alle Regeln.

(SKIP) $\dfrac{}{\Gamma \vdash \{A\}\ \text{skip}\ \{A\}}$

Betrachte s, h mit $\llbracket A \rrbracket_{s,h}$.

Betrachte $n \geq 0$, für $n = 0$ ist $\text{safe}_0(\text{skip}, s, h, \Gamma, B) = \text{true}$.

Um $\text{safe}_n(\text{skip}, s, h, \Gamma, B)$ zu zeigen,

betrachte die drei Konjunkte der Definition:

↳ Da $c = \text{skip}$, muss $\llbracket A \rrbracket_{s,h}$ gelten. ✓

↳ Betrachte h_f und h_r mit $\text{sat}_U(s, h_f, c, \Gamma)$.

↳ gilt

$(\text{skip}, s, h \circ h_f \circ h_r) \not\to$ abort,

da es von skip aus gar keine Transitionen gibt.

↳ Analog ist die letzte Bedingung erfüllt,

da es keine Transitionen von skip aus gibt.

(ATOM) $\dfrac{\text{emp} \vdash \{A\} * J \subset \{B\} * J}{\Gamma \vdash \{A\}\ \text{atomic}(c)\ \{B\}}$

Wir benötigen ein Hilfslemma.

Lemma:

$\text{safe}_n(c, s, h, \text{emp}, J * B)$ impliziert $\text{safe}_n(\underline{\text{atom}}\, c, s, h, J, B)$.

Beweis:

Induktion nach n.

$n=0$: Nichts zu tun.

$n+1$: Wir nehmen die Induktionsvoraussetzung

$$safe_n(c, s, h, emp, J * B) \Rightarrow safe_n(\underline{inatom}\ c, s, h, J, B)$$

und annehmen

$$safe_{n+1}(c, s, h, emp, J * B) \qquad (*)$$

an.

Wir müssen zeigen

$$safe_{n+1}(\underline{inatom}\ c, s, h, J, B).$$

Per Definition von $safe_{n+1}$ bedeutet das:

(i) $c = skip \Rightarrow [\![B]\!]\, s\, h.$

Gilt, da $\underline{inatom}\ c \neq skip$.

(ii) $\forall h_J, h_F.\ sat_U(s, h_J, \underline{inatom}\ c, J)$

$$\wedge\ (\underline{inatom}\ c, s, h \cup h_J \cup h_F) \to \underline{abort}.$$

Angenommen das gilt nicht.

Dann gibt es h_J und h_F mit

$sat_U(s, h_J, \underline{inatom}\ c, J)$ // Also $h_J = \emptyset$

und $(\underline{inatom}\ c, s, h \cup h_J \cup h_F) \to \underline{abort}$.

Mit Blick auf die operationelle Semantik

ist das die einzige Möglichkeit einer Transition für $\underline{inatom}\ c$,

dass diese Transition eben für c gilt,

also $(c, s, h \cup h_F \cup h_J) \to \underline{abort}$. $\lightning$ zu $(*)$.

(iii) Betrachte h_J, h_F, c', s', h' mit

$$\mathrm{sat}_{\mathcal{U}}(s, h_J, \underline{\mathrm{inatom}}\, c, J) \qquad \text{// Also } h_J = \emptyset$$

und $(\underline{\mathrm{matom}}\, c, s, h \uplus h_J \uplus h_F) \to (c', s', h')$.

Wir müssen zeigen:

$$\exists h'', h_J' : \quad h' = h'' \uplus h_J' \uplus h_F$$
$$\wedge\ \mathrm{sat}_{\mathcal{U}}(s', h_J', c', J)$$
$$\wedge\ \mathrm{safe}(c', s', h'', J, \mathcal{B}).$$

Es gibt zwei mögliche Transitionen für $\underline{\mathrm{inatom}}\, c$.

(INATOMSTEP)

$(\underline{\mathrm{inatom}}\, c, s, h \uplus h_F \uplus h_J) \to (c', s', h')$

mit $c' = \underline{\mathrm{inatom}}\, c''$.

Die Transition existiert, da

$$(c, s, h \uplus h_F \uplus h_J) \to (c'', s', h').$$

Außerdem gilt

$$\mathrm{sat}_{\mathcal{U}}(s, h_J, c, \mathrm{emp}),$$

da $h_J = \emptyset$.

Für c gilt aber safety mit (*).

Also gibt es h'' und h_J' mit:

$h' = h'' \uplus h_J' \uplus h_F$ und $\mathrm{sat}_{\mathcal{U}}(s, h_J', c'', \mathrm{emp})$

und $\mathrm{safe}(c'', s, h'', \mathrm{emp}, J * \mathcal{B})$.

Mit der Induktionsvoraussetzung folgt

$$\mathrm{safe}(\underline{\mathrm{inatom}}\, c'', s, h'', J, \mathcal{B}).$$

(INATOMEND)

$(\underline{\text{inatom}}\ \text{skip}, s, h \circledcirc h_1 \circledcirc h_F) \to (\text{skip}, s, \underbrace{h \circledcirc h_1 \circledcirc h_F}_{= h \circledcirc h_F})$

Es gilt $\text{safe}_{n+2}(\text{skip}, s, h_1, \text{emp}, J * B)$.

Also gilt insbesondere

$$[\![J * B]\!]\, s\, h.$$

Damit gilt $h = h_1 \circledcirc h_2$ mit

$$[\![J]\!]\, s\, h_2 \quad \text{und} \quad [\![B]\!]\, s\, h_2.$$

Wir wählen

$h'' := h_2$ und $h_J' := h_2$.

Damit haben wir:

- $h \circledcirc h_F = h_1 \circledcirc h_2 \circledcirc h_F$
- $\text{sadd}(s, h_J', \text{skip}, J)$
- $\text{safe}_n(\text{skip}, s, h'', J, B)$, sogar für alle n, mit der Argumentation für (SKIP) oben. □

Mit diesem Lemma folgt das Resultat.

Lemma:
Falls $\text{emp} \vdash \{A * J\} \subseteq \{B * J\}$ gilt,
dann gilt auch $J \vdash \{A\}\ \underline{\text{atomic}}\ \{B\}$.

Beweis:
Wir nehmen an, $\text{emp} \vdash \{A * J\} \subseteq \{B * J\}$ gilt.
Wir betrachten s, h mit

$$[\![A]\!]\, s\, h.$$

Außerdem betrachten wir noch:

Wir müssen zeigen:

safe (atomic c, s, h, J, B)

Für n=0 ist nichts zu tun.

Betrachte n+1.

Es sind drei Bedingungen zu prüfen.

(i) Trivial, da atomic c ≠ skip.

(ii) Trivial, da (atomic c, s, h) ↛ abort.

(iii) Betrachte h_J, h_F, c', s', h' mit

sat U (s, h_J, atomic c, J) // Also $[\![J]\!]_{s,h_J}$.

und (atomic c, s, h ∪ h_J ∪ h_F) → (c', s', h').

Es gibt nun eine mögliche Transition für atomic c,

also folgt:

$c' = \text{inatom}\ c$

$s' = s$

$h' = h \cup h_J \cup h_F$.

Wähle

$h'' := h \cup h_J$ $\qquad$ $h_J' := \emptyset$.

Dann gilt:

$h' = h'' \cup h_J' \cup h_F$.

Ferner

sat U (s, h_J', inatom c, J), da $h_J' = \emptyset$.

Ferner haben wir

safe (inatom c, s, h'', J, B)

mit obigem Lemma und der Annahme

emp + ST ∗ TRS ⊆ SJ ∗ RS.

□

$$(SEQ)\quad \frac{\Gamma \vdash \{A\}\, c_1\, \{B\} \qquad \Gamma \vdash \{B\}\, c_2\, \{C\}}{\Gamma \vdash \{A\}\, c_1 ; c_2\, \{C\}}$$

Zeige folgendes Lemma.

Lemma:

Wenn $\Gamma \vdash \{B\}\, c_2\, \{C\}$ gilt und safe(c_1, s, h, Γ, B) wahr ist,

dann ist safe$(c_1 ; c_2, s, h, \Gamma, C)$ wahr.