3.5 Soundness

In this section we want to prove the soundness of the inference rules with respect to the semantics. This means we want to ensure that if we can infer the well-typedness of a program, the execution of the program yields those kinds of values predicted by the inference rules.

3.5.1 Soundness of the type signature

The inference rules and the semantics for the type signatures are built in a structurally similar way. Thus, we will now show that the semantics of a phrase yields the kind of result predicted by the inference rules.

Theorem 3.1

Let Γ be a type context, $ltf \in$ `<list-type-fields>`, $a_i \in \mathcal{V}$,$T_i \in \mathcal{T}$ for $i \in \mathbb{N}_1^n$ and $n \in \mathbb{N}_0$. Assume that $\Gamma \vdash ltf : \{a_1 : T_1, \ldots, a_n : T_n\}$ can be derived.

—

Then $[\![ltf]\!]_\Gamma = \{a_1 : T_1, \ldots, a_n : T_n\}$.

Proof. Let Γ be a type context, $ltf \in$ `<list-type-fields>`, $a_i \in \mathcal{V}$,$T_i \in \mathcal{T}$ for $i \in \mathbb{N}_1^n$ and $n \in \mathbb{N}_0$. Assume $ltf : \{a_1 : T_1, \ldots, a_n : T_n\}$ can be derived.

- **Case** $ltf =$ `""` for $n = 0$: Then $[\![\texttt{ltf}]\!] = \{\}$ and therefore the conclusion holds.
- **Case** $ltf = a_1$ `":"` T_1 `","` ltf_1 for $ltf_1 \in$ `<list-type-field>`: Then by the premise of the inference rule for ltf we can assume that $\Gamma \vdash ltf_1 : \{a_2 : T_2, \ldots, a_n : T_n\}$ can be derived and by induction hypothesis $[\![ltf_1]\!]_\Gamma = \{a_2 : T_2, \ldots, a_n : T_n\}$. We can now use the semantics as describe in its definition $[\![ltf]\!] = [\![a_1 \texttt{ ":" } T_1 \texttt{ "," } ltf_1]\!] = \{a_1 : e_1, \ldots, a_n : e_n\}$ for $e_i \in \text{value}_\Gamma(T_i)$ for $i \in \mathbb{N}_0^n$, thus the conclusion $[\![ltf]\!] \in \text{value}_\Gamma(\{a_1 : T_1, \ldots, a_n : T_n\})$ follows.

□

Theorem 3.2

Let Γ be a type context, $lt \in$ `<list-type>`,$T_i \in \mathcal{T}$ for $i \in \mathbb{N}_1^n$ and $n \in \mathbb{N}_0$. Assume $\Gamma \vdash lt : (T_1, \ldots, T_n)$ can be derived.

—

Then $[\![lt]\!]_\Gamma = (T_1, \ldots, T_n)$.

Proof. See the combined proof of the conjunction of Theorem 3.2 and 3.3 below. □

Theorem 3.3

Let Γ be a type context, $t \in$ `<type>` and $T \in \mathcal{T}$. Assume $\Gamma \vdash t : T$ can be derived.

—

Then $\llbracket t \rrbracket_\Gamma = T$.

Proof. Combined proof of Theorems 3.2 and 3.3.

We prove the conjunction of Theorem 3.2 and 3.3 by simultaneous induction over the structure of the mutually recursive grammar rules for `<list-type>` and `<type>`.

Let Γ be a type context, $lt \in$ `<list-type>`$, T_i \in \mathcal{T}$ for $i \in \mathbb{N}_1^n$ and $n \in \mathbb{N}_0$. Assume $\Gamma \vdash lt : (T_1, \dots, T_n)$ can be derived. We show $\llbracket lt \rrbracket_\Gamma = (T_1, \dots, T_n)$.

- **Case** $lt =$ `""` for $n = 0$: Then $\llbracket lt \rrbracket = ()$ and thus the conclusion holds.
- **Case** $lt = t_1\ l_1$ for $t_1 \in$ `<type>` for $l_1 \in$ `<list-type>`: Then from the premise of the inference rule, we assume that $\Gamma \vdash l_1 : (T_2, \dots, T_n)$ and $\Gamma \vdash t_1 : T_1$ hold. The assumption of Theorem 3.3, namely that $\Gamma \vdash t_1 : T_1$ can be derived, now holds. By Its induction hypothesis we can therefore conclude that $\llbracket t_1 \rrbracket_\Gamma = T_1$ for $T_1 \in \mathcal{T}$. The assumption of Theorem 3.2, namely $\Gamma \vdash l_1 : (T_2, \dots, T_n)$, holds and therefore by the induction hypothesis of Theorem 3.2 we obtain $\llbracket t_1\ \ l_1 \rrbracket = (t_1, t_2, \dots, t_n)$ for $\llbracket t_i \rrbracket_\Gamma = T_i$ for $t_i \in$ `<type>`. Thus by the conclusion $\llbracket lt \rrbracket = (T_1, \dots, T_n)$ holds.

Let Γ be a type context, $t \in$ `<type>` and $T \in \mathcal{T}$. Assume $\Gamma \vdash t : T$ can be derived. We show $\llbracket t \rrbracket_\Gamma = T$.

- **Case** $t =$ `"Bool"`: Then $\llbracket t \rrbracket_\Gamma = \mathit{Bool}$ and the conclusion holds.
- **Case** $t =$ `"Int"`: Then by the premise of the inference rule for `"Int"`, we can assume that $\Gamma \vdash t : \mathit{Int}$ can be derived and therefore $\llbracket t \rrbracket_\Gamma = \mathit{Int}$. We see that the conclusion holds.
- **Case** $t =$ `"List"` t_2, for $t_2 \in$ `<type>`: By the premise of the inference rule we assume $\Gamma \vdash t_2 : T_2$ can be derived and by induction hypothesis $\llbracket t_2 \rrbracket_\Gamma = T_2$ for given $T_2 \in \mathcal{T}$. Then $\llbracket t \rrbracket_\Gamma = [e_1, \dots, e_n]$ for $e_i \in \mathrm{value}_\Gamma(T_2), i \in \mathbb{N}_0^n$ and $n \in \mathbb{N}$. Thus the conclusion holds.
- **Case** $t =$ `"("` t_1`","`t_2 `")"`, for $t_1, t_2 \in$ `<type>`: By the premise of the inference rule $\Gamma \vdash t_1 : T_1$ and $\Gamma \vdash t_2 : T_2$ hold for given $T_1, T_2 \in \mathcal{T}$. Then by induction hypothesis $\llbracket t_1 \rrbracket_\Gamma = T_1$ and $\llbracket t_2 \rrbracket_\Gamma = T_2$. Thus by the definition of the semantic the conclusion holds analogously to the cases above.
- **Case** $t =$ `"{"` ltf `"}"`, for $ltf \in$ `<list-type-field>`: Then by the premise of the inference rule $\Gamma \vdash ltf : \{a_1 : T_1, \dots, a_n : T_n\}$ for $a_i \in \mathcal{V}$,$T_i \in \mathcal{T}$,$i \in \mathbb{N}_1^n$ and $n \in \mathbb{N}_0$. Thus by Theorem 3.1 $\llbracket ltf \rrbracket_\Gamma = T$ and therefore the conclusion holds analogously to the cases above.
- **Case** $t = t_1$ `"->"` t_2, for $t_1, t_2 \in$ `<type>`: By the premise of the inference rule $\Gamma \vdash t_1 : T_1$ and $\Gamma \vdash t_2 : T_2$ hold for given $T_1, T_2 \in \mathcal{T}$. By induction hypothesis $\llbracket t_i \rrbracket_\Gamma = T_i$ for $i \in \{1, 2\}$. Thus by the definition of the semantic the conclusion holds analogously to the cases above.

- **Case** $t = c\ lt$ for $lt \in$ `<list-type>` and $c \in$ `<upper-var>`: By the premise of the inference rule we know $(c, T') \in \Gamma$ with $T' \in \mathcal{T}$ and can assume that $\Gamma \vdash lt : (T_0, \ldots, T_n)$ can be derived. Therefore the assumption of Theorem 3.2, namely that $\Gamma \vdash lt : (T_0, \ldots, T_n)$ can be derived, holds and by applying its induction hypothesis, we know $[\![lt]\!]_\Gamma = (T_1, \ldots, T_n)$ for $T_i \in \mathcal{T}$, $i \in \mathbb{N}^n$ and $n \in \mathbb{N}_0$. Thus by the definition of the semantic the conclusion holds.
- **Case** $t = a$ for $a \in \mathcal{V}$: Then by the definition of the semantic the conclusion holds analogously to the cases above.

□

3.5.2 Soundness of the variable context

In our previous sections we had two different meanings for Δ. We now want to show, that these two definitions correlate.

Definition 3.1: Similar Variable context

Let Γ, Δ be type contexts and Δ' a variable context.

—

We say Δ' is *similar to* Δ *with respect to* Γ iff the following holds:

$$\forall T \in \mathcal{T}. \forall a \in \mathcal{V}. (a, T) \in \Delta \Rightarrow \exists e \in \text{value}_\Gamma(T). (a, e) \in \Delta'.$$

3.5.3 Soundness of the expression semantics

We can now use the definition of well-formed variable contexts, to prove the soundness of the expression semantics.

Theorem 3.4

Let Γ,Δ be type contexts, Δ' be a variable context similar to Δ with respect to Γ and $lef \in$ `<list-exp-field>`. Assume $\Gamma, \Delta \vdash lef : T$ can be derived for $T = \{a_1 : T_1, \ldots, a_n : T_n\} \in \mathcal{T}$, $a_i \in \mathcal{V}, T_i \in \mathcal{T}$, for all $i \in \mathbb{N}_1^n$, and $n \in \mathbb{N}_0$.

—

Then $[\![lef]\!]_{\Gamma,\Delta'} \in \text{value}_\Gamma(T)$.

Proof. See the combined proof of the conjunction of Theorem 3.4, 3.7 and 3.8 below.

□

Theorem 3.5

Let $b \in$ `<bool>`.

—

Then $[\![b]\!] \in \text{value}_{\varnothing}(\mathit{Bool})$.

Proof. Let $b \in$ `<bool>`.

- **Case** $b =$ `"True"`: Then $[\![\texttt{b}]\!] = \mathit{True}$. Thus the conclusion holds.
- **Case** $b =$ `"False"`: Then $[\![\texttt{b}]\!] = \mathit{False}$. Thus the conclusion holds.

□

Theorem 3.6

Let $i \in$ `<int>`.

—

Then $[\![i]\!] \in \text{value}_{\varnothing}(\mathit{Int})$.

Proof. Let $i \in$ `<int>`.

- **Case** $i =$ `"0"`: Then $[\![\texttt{i}]\!] = 0$. Thus the conclusion holds.
- **Case** $i = n$ for $n \in \mathbb{N}$: Then $[\![\texttt{i}]\!] = \mathit{Succ}^n\ 0$. Thus the conclusion holds.
- **Case** $i =$ `"-"` n for $n \in \mathbb{N}$: Then $[\![\texttt{i}]\!] = \mathit{Neg}\ \mathit{Succ}^n\ 0$. Thus the conclusion holds.

□

Theorem 3.7

Let Γ, Δ be type contexts, Δ' be a variable context similar to Δ with respect to Γ and $le \in$ `<list-exp>`. Assume $\Gamma, \mathit{Delta} \vdash le : \mathit{List}\ T$ can be derived for $T \in \mathcal{T}$.

—

Then $[\![le]\!]_{\Gamma,\Delta'} \in \text{value}_{\Gamma}(\mathit{List}\ T)$.

Proof. See the combined proof of the conjunction of Theorem 3.4, 3.7 and 3.8 below. □

Theorem 3.8

Let Γ, Δ be type contexts, Δ' be a variable context similar to Δ with respect to Γ. Let $e \in$ `<exp>` and $T \in \mathcal{T}$. Assume $\Delta, \Gamma \vdash e : T$ can be derived.

—

Then $[\![e]\!]_{\Gamma,\Delta'} \in \text{value}_{\Gamma}(T)$.

Proof. See the combined proof of the conjunction of Theorems 3.4, 3.7 and 3.8 below.

We prove the conjunction of Theorem 3.4, 3.7 and 3.8 by simultaneous induction over the structure of the mutually recursive grammar rules for `<list-exp-field>`, `<list-exp>` and `<exp>`.

Let Γ,Δ be type contexts, Δ' be a variable context similar to Δ with respect to Γ and $\mathit{lef} \in$ `<list-exp-field>`. Assume the judgment $\Gamma, \Delta \vdash \mathit{lef} : T$ can be derived for $T = \{a_1 : T_1, \ldots, a_n : T_n\} \in \mathcal{T}$, $a_i \in \mathcal{V}, T_i \in \mathcal{T}$, for all $i \in \mathbb{N}_1^n$ and given $n \in \mathbb{N}_0$. We show $[\![\mathit{lef}]\!]_{\Gamma,\Delta'} \in \mathrm{value}_\Gamma(T)$.

- **Case** $\mathit{lef} = a_1$ `"="` e_1 for $e_1 \in$ `<exp>` and $n = 1$: Then by the premise of the inference rule we assume $\Gamma, \Delta \vdash e_1 : T_1$ can be derived and therefore the assumption of Theorem 3.8 holds. By applying said theorem we can therefore conclude $[\![e_1]\!]_{\Gamma,\Delta'} \in \mathrm{value}_\Gamma(T_1)$. Then $[\![\mathit{lef}]\!]_{\Gamma,\Delta'} = \{a_1 = e_1\}$ and therefore the conclusion holds.

- **Case** $\mathit{lef} = a_1$ `"="` e_1 `","` lef_0 for $e_1 \in$ `<exp>` and $\mathit{lef}_0 \in$ `<list-exp-field>`: Then by the premise of the inference rule we assume $\Gamma, \Delta \vdash \mathit{lef}_0 : \{a_2 : T_2, \ldots, a_n : T_n\}$ and $\Gamma, \Delta \vdash e_1 : T_1$ can both be derived. Thus the assumption of Theorem 3.8 holds and by the induction hypothesis of said theorem $[\![e_1]\!]_{\Gamma,\Delta'} \in \mathrm{value}_\Gamma(T_1)$. By $\Gamma, \Delta \vdash \mathit{lef}_0 : T$ the assumption for the induction hypothesis of Theorem 3.4 holds and therefore by appling the theorem we obtain $[\![\mathit{lef}_0]\!]_{\Gamma,\Delta'} \in \mathrm{value}_\Gamma(\{a_2 : T_2, \ldots, a_n : T_n\})$. Then $[\![\mathit{lef}]\!]_{\Gamma,\Delta'} = \{a_1 = e_1, \ldots, a_n = e_n\}$ for $e_i \in \mathrm{value}_\Gamma(T_i)$ and thus the conclusion holds.

Let Γ, Δ be type contexts, Δ' be a variable context similar to Δ with respect to Γ and $\mathit{le} \in$ `<list-exp>`. Assume $\Gamma, \Delta \vdash \mathit{le} : \mathit{List}\ T$ can be derived for given $T \in \mathcal{T}$. We show $[\![\mathit{le}]\!]_{\Gamma,\Delta'} \in \mathrm{value}_\Gamma(\mathit{List}\ T)$.

- **Case** $\mathit{le} =$ `""`: Then $[\![$`""`$]\!] = \mathit{Empty}$ and thus the conclusion holds.

- **Case** $\mathit{le} = e,\ \mathit{le}_1$ for $e \in$ `<exp>` and $\mathit{le}_1 \in$ `<list-exp>`: Then by the premise of the inference rule we assume $\Gamma, \Delta \vdash e : T$ and $\Gamma, \Delta \vdash \mathit{le}_1 : \mathit{List}\ T$ can be derived. The assumption of Theorem 3.8, namely that $\Gamma, \Delta \vdash e : T$ can be derived, holds and by appling that theorem $[\![e]\!]_{\Gamma,\Delta'} \in \mathrm{value}_\Gamma(T)$. The assumption of Theorem 3.7, namely that $\Gamma, \Delta \vdash \mathit{le}_1 : \mathit{List}\ T$ can be derived, also holds and by appling said theorem we conclude $[\![\mathit{le}_1]\!]_{\Gamma,\Delta'} \in \mathrm{value}_\Gamma(\mathit{List}\ T)$. By using the definition of the semantic $[\![\mathit{le}]\!]_{\Gamma,\Delta'} = \mathit{Cons}\ e\ [\![\mathit{le}_1]\!]_{\Gamma,\Delta'}$ and therefore the conclusion holds.

//TODO: proof 3.8 □

3.5.4 Soundness of the statement semantics

Statements are modelled as operations on either the type context or the variable context. We will now show that their semantics conforms to the result of the inference rules.

Theorem 3.9

Let $\mathit{lsv} \in$ `<list-statement-var>`, $a_i \in \mathbb{N}_1^n$ for $n \in \mathbb{N}_0$. Assume $\mathit{lsv} : (a_1, \ldots, a_n)$ can be derived.

—

Then $[\![\texttt{lsv}]\!] \in \mathcal{V}^*$.

Proof. Let $\mathit{lsv} \in$ `<list-statement-var>`, $a_i \in \mathbb{N}_1^n$ for $n \in \mathbb{N}_0$. Assume $\mathit{lsv} : (a_1, \ldots, a_n)$ can be derived.

- **Case** $\mathit{lsv} =$ `""` and $n = 0$: Then $[\![\mathit{lsv}]\!] = ()$ and thus the conclusion holds.
- **Case** $\mathit{lsv} = a_1\ \mathit{lsv}_1$ for $\mathit{lsv}_1 \in$ `<list-statement-var>`: Then by the inference rule of lsv, we assume that $\mathit{lsv}_1 : (a_2, \ldots, a_n)$ can be derived. Then by induction hypothesis $[\![\mathit{lsv}_1]\!] = (a_2, \ldots, a_n)$, and therefore $[\![\mathit{lsv}]\!] = (a_1, \ldots, a_n)$. Thus the conclusion holds.

□

Theorem 3.10

Let $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2$ be type contexts and Δ_1' be a variable context similar to Δ_1 respectively with respect to Γ. Let $s \in$ `<statement>` and assume $\Gamma_1, \Delta_1, s \vdash \Gamma_2, \Delta_2$ can be derived.

—

Then $[\![s]\!](\Gamma_1, \Delta_1') = (\Gamma_2, \Delta_2')$ for a variable context Δ_2' similar to Δ_2 with respect to Γ.

Proof. Let $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2$ be type contexts and Δ_1', Δ_2' be a variable context similar to Δ_1, Δ_2 respectively with respect to Γ_1, Δ_2 respectively. Let $s \in$ `<statement>` and assume $\Gamma_1, \Delta_1, s \vdash \Gamma_2, \Delta_2$ can be derived.

- **Case** $s = \mathit{mss}\ a$ `"="` e for $\mathit{mss} \in$ `<maybe-statement-sort>`, $a \in \mathcal{V}$, $e \in$ `<exp>`, $\Gamma_1 = \Gamma_2$ and $\Delta_2 = \mathrm{insert}_{\Delta_1}(\{(a, T)\})$ for $T \in \mathcal{T}$: Then from the premise of the inference rule, we assume that $\Gamma_1, \mathit{mss} \vdash e : T$ and $\Gamma_1, \Delta_2 \vdash e : T$ can both be derived. Then by Theorem 3.8, we know $[\![e]\!]_{\Gamma_1, \Delta_1'} \in \mathrm{value}_\Gamma(T)$. Let $\Delta_2' = \Delta_1' \cup \{(a, e)\}$. Then $[\![s]\!](\Gamma_1, \Delta_1') = (\Gamma_2, \Delta_2')$. We know Δ_1 is similar to Δ_1' with respect to Γ and therefore from $[\![e]\!]_{\Gamma_1, \Delta_1'} \in \mathrm{value}_\Gamma(T)$ we can follow directly that Δ_2' is similar to Δ_2. Thus the conclusion holds.
- **Case** $s =$ `"type alias"` $c\ \mathit{lsv}$ `"="` t for $\mathit{lsv} \in$ `<list-statement-variable>`, $c \in \mathcal{V}$ such that $\Delta_1 = \Delta_2$ and $(c, \_) \notin \Gamma_1$: Let $\Delta_1' = \Delta_2'$. From $[\![s]\!](\Gamma_1, \Delta_1') = (\Gamma_2, \Delta_2')$ the conclusion trivially holds.

□

Theorem 3.11

Let $\Gamma_1, \Delta_1, \Gamma_2, \Delta_2$ be type contexts and Δ_1', be a variable context similar to Δ_1 with respect to Γ. Let $\mathit{ls} \in$ `<list-statement>` such that $\Gamma_1, \Delta_1, \mathit{ls} \vdash \Gamma_2, \Delta_2$ can be derived.

—

Then $[\![ls]\!](\Gamma_1, \Delta'_1) = (\Gamma_2, \Delta'_2)$ for a variable context Δ'_2 similar to Δ_2 with respect to Γ.

Proof. $\Gamma_1, \Delta_1, \Gamma_2, \Delta_2$ be type contexts and Δ'_1, be a variable context similar to Δ_1 with respect to Γ. Let $ls \in$ `<list-statement>` such that $\Gamma_1, \Delta_1, ls \vdash \Gamma_2, \Delta_2$ can be derived.

- **Case** $ls =$ `""` for $\Gamma_1 = \Gamma_2$ and $\Delta_1 = \Delta_2$: Let $\Delta'_1 = \Delta'_2$. Then $[\![ls]\!] = id$ and therefore the conclusion holds.

- **Case** $ls = s$ `";"` ls_1 for $s \in$ `<statement>` and $ls_1 \in$ `<statement-list>`: From the premise of the inference rule, we assume $\Gamma_1, \Delta_1, s \vdash \Gamma_3, \Delta_3$ and $\Gamma_3, \Delta_3, ls_1 \vdash \Gamma_2, \Delta_2$ for some type contexts Γ_2, Δ_2. We know by Theorem 3.10 that $[\![s]\!](\Gamma_1, \Delta'_1) = (\Gamma_3, \Delta'_3)$ for a given variable context Δ'_3 similar to Δ_3 with respect to Γ. Also by Theorem 3.11 we know $[\![ls_1]\!](\Gamma_3, \Delta'_3) = (\Gamma_2, \Delta'_2)$ for a given Δ'_2 similar to Δ_2 with respect to Γ. Thus $[\![ls]\!] = [\![s]\!] \circ [\![ls_1]\!]$ and therefore the conclusion holds.

□

3.5.5 Soundness of the Program Semantic

A program is a sequence of statements. Starting with an empty type context, and an empty variable context, one statement at the time will be applied, resulting in a value e, a type T and a type context Γ such that $e \in \text{value}_\Gamma(T)$.

Theorem 3.12

Let $p \in$ `<program>` and $T \in \mathcal{T}$. Assume $p : T$ can be derived.

—

Then there exists a type context Γ such that $[\![p]\!] \in \text{value}_\Gamma(T)$.

Proof. We prove the conjunction of Theorem 3.11 and 3.12 by simultaneous induction over the structure of the mutually recursive grammar rules grammar rules for `<list-statement>` and `<program>`.

Let ls mms `"main="` $e \in$ `<program>`, $ls \in$ `<list-statement>`, $mms \in$ `<maybe-main-sign>` and $e \in$ `<exp>`. Assume $p : T$ for $T \in \mathcal{T}$ and $\varnothing, \varnothing, ls \vdash \Gamma, \Delta$ can be derived for type contexts Γ and Δ.

The assumption of Theorem 3.11, namely that $\varnothing, \varnothing, ls \vdash \Gamma, \Delta$ can be derived, holds. By appling said theorem we obtain $[\![ls]\!](\varnothing, \varnothing) = (\Gamma, \Delta')$ for a variable context Δ' similar to Δ with respect to Γ. Therefore $[\![p]\!] = [\![e]\!]_{\Gamma,\Delta'} \in \text{value}_\Gamma(T)$ and the conclusion holds. □