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 a if we can infer the type of a program, the program has also got a semantic.

3.5.1 Soundness of the type signiture signiture

The judgement and the semantics for the type signitures are build structually similar therefore we can easily show their equivalance.

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 $n \in \mathbb{N}_0$.

—

The Judgment $\Gamma \vdash ltf : \{a_1 : T_1, \ldots, a_n : T_n\}$ can be derived if and only if $[\![lft]\!]_\Gamma = \{a_1 : T_1, \ldots, a_n : T_n\}$.

Proof. Γ be a type context.

- **Case** $ltf =$ `""`: Then $\Gamma \vdash ltf : \{\ \}$ and $[\![lft]\!]_\Gamma = \{\ \}$.
- **Case** $lft = a_0$ `":"` T_0 `","` lft_1 for given $a_0 \in \mathcal{V}$, $T_0 \in \mathcal{T}$ and $ltf_1 \in$ `<list-type-field>` where $\Gamma \vdash ltf_1 : \{a_1 : T_1, \ldots, a_n : T_n\}$ and $[\![ltf_1]\!]_\Gamma = \{a_1 : T_1, \ldots, a_n : T_n\}$ for given $T_i \in \mathcal{T}, a_i \in \mathcal{V}$ and $i \in \mathbb{N}_1^n, n \in \mathbb{N}_0$:
  Then $\Gamma \vdash lft : \{a_0 : T_0, a_1 : T_1 \ldots a_n : T_n\}$ and $[\![ltf]\!]_\Gamma = \{a_0 : T_0, a_1 : T_1 \ldots a_n : T_n\}$.

□

Theorem 3.2

Let Γ be a type context, $T_i \in \mathcal{T}$ for $i \in \mathbb{N}_1^n$ and $n \in \mathbb{N}_0$.

—

The Judgment $\Gamma \vdash lt : (T_1, \ldots, T_n)$ can be derived if and only if $[\![lt]\!] = (T_1, \ldots, T_n)$.

Proof. Γ be a type context.

- **Case** $l =$ `""`: then $\Gamma \vdash l : ()$ and $[\![l]\!]_\Gamma = ()$.
- **Case** $l = \ t_0\ l_1$ for $l_1 \in$ `<list-type>` that $\Gamma \vdash l_1 : (T_1, \ldots, T_n)$ where $T_i \in \mathcal{T}$,$i \in \mathbb{N}_1^n$ and $n \in \mathbb{N}_0$, $t_0 \in$ `<type>` such that $\Gamma \vdash t_0 : T_0$ and $[\![t_0]\!]_\Gamma = T_0$ and $T_0 \in \mathcal{T}$:
  Then $\Gamma \vdash l : (T_0, T_1, \ldots, T_n)$.

□

Theorem 3.3

Let Γ be a type context, $t \in$ `<type>` and $T \in \mathcal{T}$.

—

The Judgment $\Gamma \vdash t : T$ can be derived if and only if $[\![t]\!]_\Gamma = T$.

Proof. Let Γ be a type context, $t \in$ `<type>` and $T \in \mathcal{T}$.

- **Case** $t =$ `"Bool"`: Then $\Gamma \vdash t : \mathit{Bool}$ and $[\![t]\!]_\Gamma = \mathit{Bool}$.
- **Case** $t =$ `"Int"`: Then $\Gamma \vdash t : \mathit{Int}$ and $[\![t]\!]_\Gamma = \mathit{Int}$.
- **Case** $t =$ `"List"` t_2, for $t_2 \in$ `<type>`: By induction hypothesis, assume $\Gamma \vdash t_2 : T_2$ and $[\![t_2]\!]_\Gamma = T_2$ for given $T_2 \in \mathcal{T}$. Then $\Gamma \vdash t : \mathit{List}\ T_2$ and $[\![t]\!]_\Gamma = \mathit{List}\ T_2$.
- **Case** $t =$ `"("` t_1`","`t_2 `")"`, for $t_1, t_2 \in$ `<type>`: By induction hypothesis, assume $\Gamma \vdash t_1 : T_1$ and $\Gamma \vdash t_2 : T_2$ for given $T_1, T_2 \in \mathcal{T}$. Then $\Gamma \vdash t : (T_1, T_2)$ and $[\![t]\!]_\Gamma = (T_1, T_2)$.
- **Case** $t =$ `"{"` ltf `"}"`, for $\mathit{ltf} \in$ `<list-type-field>`: Then $\Gamma \vdash \mathit{ltf} : T$ and $[\![\mathit{ltf}]\!]_\Gamma = T$ for $T = \{a_1 : T_1, \dots, a_n : T_n\}$, $a_i \in \mathcal{V}$,$T_i \in \mathcal{T}$ for $i \in \mathbb{N}_1^n$ and $n \in \mathbb{N}_0$. Therefore $\Gamma \vdash t : T$ and $[\![t]\!]_\Gamma = T$.
- **Case** $t = t_1$ `"->"` t_2, for $t_1, t_2 \in$ `<type>`: By induction hypothesis, assume $\Gamma \vdash t_i : T_i$ and $[\![t_i]\!]_\Gamma = T_i$ for $i \in \{1,2\}$ and given $T_1, T_2 \in \mathcal{T}$. Then $\Gamma \vdash t : T_1 \to T_2$ and $[\![t]\!]_\Gamma = T_1 \to T_2$.
- **Case** $t = c\ l$ for $l \in$ `<list-type>` and $c \in$ `<upper-var>` with $(c, T') \in \Gamma$ with $T' \in \mathcal{T}$: Then $\Gamma \vdash l : (T_0, \dots, T_n)$ and $[\![l]\!]_\Gamma = (T_1, \dots, T_n)$ for $T_i \in \mathcal{T}$,$i \in \mathbb{N}^n$ and $n \in \mathbb{N}_0$. Then $\Gamma \vdash t : \overline{T'}\ T_1\ \dots\ T_n$ and $[\![t]\!]_\Gamma = \overline{T'}\ T_1\ \dots\ T_n$, where $\overline{T'}$ represents the application constructor of T'
- **Case** $t = a$ for $a \in \mathcal{V}$: Then $\Gamma \vdash t : \forall\ b.b$ and $[\![t]\!]_\Gamma = \forall\ b.b$.

$\square$

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: Well formed variable context

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

—

We say Δ' is *well defined with respect to* Δ $:\Leftrightarrow$

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

We will now show that our semantic ensures that all used variable contexts are well formed.

//Todo

3.5.3 Soundness of the Expression Semantic

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

Theorem 3.4

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

—

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

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

- **Case** $lef = a_1$ `"="` e for $e \in$ `<type>` and $n = 1$: Then $\Gamma, \Delta \vdash e : T_1$ and therfore $[\![e]\!]_{\Gamma,\Delta'} \in \text{value}_\Gamma(T)$. Thus the assumption holds.
- **Case** $lef = a_1$ `"="` e `","` lef_0 for $e \in$ `<type>` and $lef_0 \in$ `<list-exp-field>`: Then $\Gamma, \Delta \vdash lef_0 : T$,$\Gamma, \Delta \vdash e : T_1$ and therefore $[\![e]\!]_{\Gamma,\Delta'} \in \text{value}_\Gamma(T_1)$ and by induction hypothesis $[\![lef_0]\!]_{\Gamma,\Delta'} \in \text{value}_\Gamma(\{a_2 : T_2, \ldots, a_n : T_n\})$. Thus the assumption holds.

□

Theorem 3.5

Let $mes \in$ `<maybe-exp-sign>`.

—

Then $[\![mes]\!] = ()$.

Proof. //Todo □

Theorem 3.6

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

—

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

Proof. //Todo ☐

Theorem 3.7

Let $i \in$ `<int>`. Let $i : \mathit{Int}$.

—

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

Proof. //Todo ☐

Theorem 3.8

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

—

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

Proof. //Todo ☐

Theorem 3.9

Let Γ, Δ be type contexts, Δ' be a well formed variale context with respect to Δ. Let $e \in$ `<exp>` and $T \in \mathcal{T}$. Let $\Delta, \Gamma \vdash e : T$ be valid.

—

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

Proof. //Todo ☐

3.5.4 Soundness of the Statement Semantic

Statements are modeled as operations on either the type context or the variable context. We will now show that their definitions described in the semantics are the same as the definition in the inference rules.

Theorem 3.10

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

—

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

Proof. //Todo ☐

Theorem 3.11

Let $\Gamma_1, \Delta_1, \Gamma_2, \Delta_2$ be type contexts and Δ_1', Δ_2' be well formed variable contexts with respect to Δ_1 and Δ_2 respectively. Let $ls \in$ `<list-statement>` such that $\Gamma_1, \Delta_1, ls \vdash \Gamma_2, \Delta_2$ holds.

—

Then $[\![ls]\!](\Gamma_1, \Delta_1') = (\Gamma_2, \Delta_2')$.

Proof. //Todo □

Theorem 3.12

Let Γ be a type context, $mss \in$ `<maybe-statement-sign>`, $a \in \mathcal{V}, T \in \mathcal{T}$. Let $\Gamma, mss \vdash a : T$ hold.

—

Then $[\![mss]\!] = ()$.

Proof. //Todo □

Theorem 3.13

Let $\Gamma_1, \Gamma_2, \Delta_1, \Delta_2$ be type contexts and Δ_1', Δ_2' be well defined variable contexts with respect to Δ_1, Δ_2. Let $s \in$ `<statement>` and $\Gamma_1, \Delta_1, s \vdash \Gamma_2, \Delta_2$ hold.

—

Then $[\![s]\!](\Gamma_1, \Delta_1') = (\Gamma_2, \Delta_2')$.

Proof. //Todo □

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 3.14

Let Γ be a type context, $mms \in$ `<maybe-main-sign>` and $T \in \mathcal{T}$ such that $\Gamma, mms \vdash \texttt{main} : T$ holds.

—

Then $[\![mms]\!] = ()$.

Proof. //Todo □

Theorem 3.15

Let $p \in$ `<program>` and $T \in \mathcal{T}$ such that $p : T$.

—

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

Proof. //Todo $\square$