## 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 $\Gamma$ 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 $\Gamma$ 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 hypothesis 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 $[\![ltf]\!] = [\![a_1$ `":"` $T_1$ `","` $ltf_1]\!] = \{a_1 : e_1, \ldots, a_n : e_n\}$ for $e_i \in \text{value}_\Gamma(T_i)$ for $i \in \mathbb{N}_0^n$, thus the hypothesis $[\![ltf]\!] \in \text{value}_\Gamma(\{a_1 : T_1, \ldots, a_n : T_n\})$ follows.

□

**Theorem 3.2**

Let $\Gamma$ 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 Theorem 3.3. □

**Theorem 3.3**

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

—

Then $[\![t]\!]_\Gamma = T$.

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

- **Case** $t =$ `"Bool"`: Then $[\![t]\!]_\Gamma = \mathit{Bool}$ and the hypothesis 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 $[\![t]\!]_\Gamma = \mathit{Int}$. We see that the hypothesis 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 $[\![t_2]\!]_\Gamma = T_2$ for given $T_2 \in \mathcal{T}$. Then $[\![t]\!]_\Gamma = [e_1, \ldots, e_n]$ for $e_i \in \text{value}_\Gamma(T_2), i \in \mathbb{N}_0^n$ and $n \in \mathbb{N}$. Thus the hypothesis 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 $[\![t_1]\!]_\Gamma = T_1$ and $[\![t_2]\!]_\Gamma = T_2$. Thus by the definition of the semantic the hypothesis holds analogously to the cases above.
- **Case** $t =$ `"{"` *ltf* `"}"`, for *ltf* $\in$ `<list-type-field>`: Then by the premise of the inference rule $\Gamma \vdash \mathit{ltf} : \{a_1 : T_1, \ldots, a_n : T_n\}$ 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 $[\![\mathit{ltf}]\!]_\Gamma = T$ and therefore the hypothesis 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 $[\![t_i]\!]_\Gamma = T_i$ for $i \in \{1, 2\}$. Thus by the definition of the semantic the hypothesis holds analogously to the cases above.
- **Case** $t = c$ *lt* for *lt* $\in$ `<list-type>` and $c \in$ `<upper-var>`:We will now prove Theorem 3.2 by using the induction hypothesis.

  Assume $\Gamma \vdash \mathit{lt} : (T_1, \ldots, T_n)$ can be derived, for $T_i \in \mathcal{T}$ for $i \in \mathbb{N}_1^n$ and $n \in \mathbb{N}_0$.

  - **Case** *lt* $=$ `""` for $n = 0$: Then $[\![\mathit{lt}]\!] = ()$ and thus the hypothesis holds.
  - **Case** *lt* $= t_1\ l_1$ for $l_1 \in$ `<list-type>`: Then from the premise of the inference rule, we assume that $\Gamma \vdash l_1 : (T_2, \ldots, T_n)$ and $\Gamma \vdash t_1 : T_1$ hold. By our outer induction hypothesis $[\![t_1]\!]_\Gamma = T_1$ for $T_1 \in \mathcal{T}$ and therefore $[\![t_1\ l_1]\!] = (t_1, t_2, \ldots, t_n)$ for $[\![t_i]\!]_\Gamma = T_i$ for $t_i \in$ `<type>`. Thus by the hypothesis $[\![\mathit{lt}]\!] = (T_1, \ldots, T_n)$ holds.

  We can therefore conculde that Theorem 3.2 holds. By the premise of the inference rule $(c, T') \in \Gamma$ with $T' \in \mathcal{T}$ and $\Gamma \vdash \mathit{lt} : (T_0, \ldots, T_n)$. By Theorem 3.2 we know $[\![\mathit{lt}]\!]_\Gamma = (T_1, \ldots, T_n)$ 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 hypothesis holds.
- **Case** $t = a$ for $a \in \mathcal{V}$: Then by the definition of the semantic the hypothesis holds analogously to the cases above.

□

### 3.5.2 Soundness of the variable context

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

**Definition 3.1: Similar Variable context**

Let $\Gamma, \Delta$ be type contexts and $\Delta'$ a variable context.

—

We say $\Delta'$ is *similar to* $\Delta$ *with respect to* $\Gamma$ 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 $\Gamma$,$\Delta$ be type contexts, $\Delta'$ be a variable context similar to $\Delta$ with respect to $\Gamma$ 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 Theorem 3.8. □

**Theorem 3.5**

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

—

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

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

- **Case** $b =$ `"True"`: Then $[\![$`b`$]\!] = True$. Thus the hypothesis holds.
- **Case** $b =$ `"False"`: Then $[\![$`b`$]\!] = False$. Thus the hypothesis holds.

□

**Theorem 3.6**

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

—

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

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

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

□

**Theorem 3.7**

Let $\Gamma, \Delta$ be type contexts, $\Delta'$ be a variable context similar to $\Delta$ with respect to $\Gamma$ and $le \in$ `<list-exp>`.

—

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

*Proof.* See Theorem 3.8. □

**Theorem 3.8**

Let $\Gamma, \Delta$ be type contexts, $\Delta'$ be a variable context similar to $\Delta$ with respect to $\Gamma$. 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.* //TODO: proof 3.8

//TODO: integrate proof 3.4 into 3.8

//TODO: integrate proof 3.7 into 3.8

It follows the proof of theorem 3.4: Let $\Gamma$,$\Delta$ be type contexts, $\Delta'$ be a variable context similar to $\Delta$ with respect to $\Gamma$ and $lef \in$ `<list-exp-field>`. Assume the judgment $\Gamma, \Delta \vdash lef : T$ 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 given $n \in \mathbb{N}_0$.

- **Case** $lef = a_1$ `"="` $e$ for $e \in$ `<exp>` and $n = 1$: Then by the premise of the inference rule $\Gamma, \Delta \vdash e : T_1$ and therfore by our premise $[\![e]\!]_{\Gamma,\Delta'} \in \text{value}_{\Gamma}(T)$. Thus by the definition of the semantic the hypothesis holds.

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

□

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

—

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

*Proof.* //TODO: proof 3.9 □

**Theorem 3.10**

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

—

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

*Proof.* See Theorem 3.12. □

**Theorem 3.11**

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

—

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

*Proof.* //TODO: proof 3.11 □

### 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 $\Gamma$ such that $e \in \text{value}_\Gamma(T)$.

**Theorem 3.12**

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

—

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

*Proof.* //TODO: proof 3.12

//TODO: integrate proof 3.10 into 3.12 □