TaPL Seminar

3.1 - 3.4

Kosuke Kiuchi

March 20th, 2023

3 Untyped Arithmetic Expressions

3.1 Introduction

■Language

```
t ::= true
      false
      if t then t else t
      0
      succ t
      pred t
      iszero t
```

Symbol `t` in the right-hand sides is called a *metavariable*.
It is a place-holder for some particular term.
"Meta" is because it is a variable of the *metalanguage*, not *object language*.

■Program

In the present language, program = term.

```
  if false then 0 else 1;
▶ 1
  iszero (pred (succ 0));
▶ true
```

The results of evaluations are either boolean constants or numbers.
Such terms are called *values*.
Notice that the syntax can make terms like `succ true` and `if 0 then 0 else 0`.

3.2 Syntax

■Ways to define terms

- Inductively
- Rule Inferred
- Concretely

■Inductive Definition

Definition 1 (3.2.1 Terms, Inductively)**.** The set of *terms* is the smallest set $\mathcal{T}$ such that

1. $\{\texttt{true, false, 0}\} \subseteq \mathcal{T}$;
2. if $\texttt{t}_1 \in \mathcal{T}$, then $\{\texttt{succ t}_1\texttt{, pred t}_1\texttt{, iszero t}_1\} \subseteq \mathcal{T}$;
3. if $\texttt{t}_1 \in \mathcal{T}$, $\texttt{t}_2 \in \mathcal{T}$, and $\texttt{t}_3 \in \mathcal{T}$, then $\texttt{if t}_1\texttt{ then t}_2\texttt{ else t}_3 \in \mathcal{T}$.

■Definition by Inference Rules

Definition 2 (3.2.2 Terms, by Inference Rules)**.** The set of terms is defined by the following rules:

$$\texttt{true} \in \mathcal{T} \qquad \texttt{false} \in \mathcal{T} \qquad \texttt{0} \in \mathcal{T}$$

$$\frac{\texttt{t}_1 \in \mathcal{T}}{\texttt{succ t}_1 \in \mathcal{T}} \qquad \frac{\texttt{t}_1 \in \mathcal{T}}{\texttt{pred t}_1 \in \mathcal{T}} \qquad \frac{\texttt{t}_1 \in \mathcal{T}}{\texttt{iszero t}_1 \in \mathcal{T}}$$

$$\frac{\texttt{t}_1 \in \mathcal{T} \quad \texttt{t}_2 \in \mathcal{T} \quad \texttt{t}_3 \in \mathcal{T}}{\texttt{if t}_1\texttt{ then t}_2\texttt{ else t}_3 \in \mathcal{T}}$$

Each rule is called *inference rule*. Each rule is read, "If we have established the statements in the premise(s) listed above the line, then we may derive the conclusion below the line."

- Rules with no premises are often called *axioms*.
- The term *inference rule* includes both axioms and rules with one or more premises.
- Axioms are usually written with no bar.

■Concrete Definition

More "concrete" style that gives an explicit procedure for *generating* the elements of $\mathcal{T}$.

Definition 3 (3.2.3 Terms, Concretely)**.** For each natural number i, define a set $\mathcal{S}_i$ as follows:

$$\begin{aligned} \mathcal{S}_0 &= \emptyset \\ \mathcal{S}_{i+1} &= \quad \{\texttt{true, false, 0}\} \\ &\quad \cup\{\texttt{succ t}_1\texttt{, pred t}_1\texttt{, iszero t}_1 | \texttt{t}_1 \in \mathcal{S}_i\} \\ &\quad \cup\{\texttt{if t}_1\texttt{ then t}_2\texttt{ else t}_3 | \texttt{t}_1\texttt{, t}_2\texttt{, t}_3 \in \mathcal{S}_i\} \end{aligned}$$

Finally, let $\mathcal{S} = \bigcup_i \mathcal{S}_i$.

■Ex. 3.2.4 [★★] How many elements does $\mathcal{S}_3$ have?

$$\begin{aligned} \mathcal{S}_0 =& \ \emptyset \\ \therefore |\mathcal{S}_0| =& \ 0 \\ \mathcal{S}_{i+1} =& \ \{\texttt{true, false, 0}\} \\ & \cup\{\texttt{succ t}_1\texttt{, pred t}_1\texttt{, iszero t}_1 | \texttt{t}_1 \in \mathcal{S}_i\} \\ & \cup\{\texttt{if t}_1\texttt{ then t}_2\texttt{ else t}_3 | \texttt{t}_1\texttt{, t}_2\texttt{, t}_3 \in \mathcal{S}_i\} \\ \therefore |\mathcal{S}_{i+1}| =& \ 3 + 3 \times |\mathcal{S}_i| + |\mathcal{S}_i|^3 \end{aligned}$$

$$\begin{aligned} |\mathcal{S}_1| =& 3 + 3 \times 0 + 0^3 &= 3 \\ |\mathcal{S}_2| =& 3 + 3 \times 3 + 3^3 &= 39 \\ |\mathcal{S}_3| =& 3 + 3 \times 39 + 39^3 &= 59439 \end{aligned}$$

■Ex. 3.2.5 [★★] Show that $\mathcal{S}_i \subseteq \mathcal{S}_{i+1}$

Prove inductively. Assume that $\texttt{t} \in \mathcal{S}_i$.

- If $\texttt{t}$ is either $\texttt{true, false, 0}$, obvious.

- If $\mathtt{t}$ has the form $\mathtt{succ\ t_1}$, $\mathtt{t_1} \in \mathcal{S}_{i-1}$ holds. From induction hypothesis, $\mathtt{t_1} \in \mathcal{S}_i$.
  Therefore $\mathtt{succ\ t_1} \in \mathcal{S}_{i+1}$. The same holds for `pred` and `iszero`.
- If $\mathtt{t}$ has the form $\mathtt{if\ t_1\ then\ t_2\ else\ t_3}$, $\mathtt{t_1}, \mathtt{t_2}, \mathtt{t_3} \in \mathcal{S}_{i-1}$ holds.
  From induction hypothesis, $\mathtt{t_1}, \mathtt{t_2}, \mathtt{t_3} \in \mathcal{S}_i$.
  Therefore $\mathtt{if\ t_1\ then\ t_2\ else\ t_3} \in \mathcal{S}_{i+1}$

■Two Views Define the Same Set

Proposition 4 (3.2.6)**.** $\mathcal{T} = \mathcal{S}$

Proof. Read p.28. □

3.3 Induction on Terms

■Inductive

If $\mathtt{t} \in \mathcal{T}$, then one of three things must be true:

1. $\mathtt{t}$ is a constant.
2. $\mathtt{t}$ has the form $\mathtt{succ\ t_1,\ pred\ t_1}$, or $\mathtt{iszero\ t_1}$ for some *smaller* term $\mathtt{t_1}$.
3. $\mathtt{t}$ has the form $\mathtt{if\ t_1\ then\ t_2\ else\ t_3}$ for some *smaller* terms $\mathtt{t_1,\ t_2\ and\ t_3}$

We can

- give *inductive definitions* of functions.
- give *inductive proofs* of properties of terms.

■Inductive Definitions of Consts(t)

Definition 5 (3.3.1)**.**

$$
\begin{aligned}
Consts(\mathtt{true}) &= \{\mathtt{true}\} \\
Consts(\mathtt{false}) &= \{\mathtt{false}\} \\
Consts(\mathtt{0}) &= \{\mathtt{0}\} \\
Consts(\mathtt{succ\ t_1}) &= Consts(\mathtt{t_1}) \\
Consts(\mathtt{pred\ t_1}) &= Consts(\mathtt{t_1}) \\
Consts(\mathtt{iszero\ t_1}) &= Consts(\mathtt{t_1}) \\
Consts(\mathtt{if\ t_1\ then\ t_2\ else\ t_3}) &= Consts(\mathtt{t_1}) \cup Consts(\mathtt{t_2}) \cup Consts(\mathtt{t_3})
\end{aligned}
$$

■Inductive Definition of size(t)

Definition 6 (3.3.2 size(t))**.**

$$
\begin{aligned}
size(\mathtt{true}) &= 1 \\
size(\mathtt{false}) &= 1 \\
size(\mathtt{0}) &= 1 \\
size(\mathtt{succ\ t_1}) &= size(\mathtt{t_1}) + 1 \\
size(\mathtt{pred\ t_1}) &= size(\mathtt{t_1}) + 1 \\
size(\mathtt{iszero\ t_1}) &= size(\mathtt{t_1}) + 1 \\
size(\mathtt{if\ t_1\ then\ t_2\ else\ t_3}) &= size(\mathtt{t_1}) + size(\mathtt{t_2}) + size(\mathtt{t_3}) + 1
\end{aligned}
$$

The size of $\mathtt{t}$ is the number of nodes in its abstract syntax tree.

■Inductive Definition of depth(t)

Definition 7 (3.3.2 depth($\mathtt{t}$)).

$$\begin{aligned}
&depth(\mathtt{true}) &&= 1\\
&depth(\mathtt{false}) &&= 1\\
&depth(\mathtt{0}) &&= 1\\
&depth(\mathtt{succ\ t_1}) &&= depth(\mathtt{t_1}) + 1\\
&depth(\mathtt{pred\ t_1}) &&= depth(\mathtt{t_1}) + 1\\
&depth(\mathtt{iszero\ t_1}) &&= depth(\mathtt{t_1}) + 1\\
&depth(\mathtt{if\ t_1\ then\ t_2\ else\ t_3}) &&= \max(depth(\mathtt{t_1}), depth(\mathtt{t_2}), depth(\mathtt{t_3})) + 1
\end{aligned}$$

The depth of $\mathtt{t}$ is the smallest i such that $\mathtt{t} \in \mathcal{S}_i$.

■Inductive Proof of a Simple Fact

Lemma 8 (3.3.3). $|Consts(t)| \leq size(t)$

Proof. By induction on the depth of $\mathtt{t}$.
Case: $\mathtt{t}$ is a constant
Immediate: $|Consts(\mathtt{t})| = |\{\mathtt{t}\}| = 1 = size(\mathtt{t})$.
Case: $\mathtt{t} = \mathtt{succ\ t_1}$, $\mathtt{pred\ t_1}$ or $\mathtt{iszero\ t_1}$
By the IH, $|Consts(\mathtt{t_1})| \leq size(\mathtt{t_1})$.
$\therefore |Consts(\mathtt{t})| = |Consts(\mathtt{t_1})| \leq size(\mathtt{t_1}) < |Consts(\mathtt{t})|$ Case: $\mathtt{t} = \mathtt{if\ t_1\ then\ t_2\ else\ t_3}$
By the IH, $|Consts(\mathtt{t_1})| \leq size(\mathtt{t_1})$, $|Consts(\mathtt{t_2})| \leq size(\mathtt{t_2})$ and $|Consts(\mathtt{t_3})| \leq size(\mathtt{t_3})$.

$$\begin{aligned}
\therefore |Consts(\mathtt{t})| &= |Consts(\mathtt{t_1}) \cup Consts(\mathtt{t_2}) \cup Consts(\mathtt{t_3})|\\
&\leq |Consts(\mathtt{t_1})| + |Consts(\mathtt{t_2})| + |Consts(\mathtt{t_3})|\\
&\leq size(\mathtt{t_1}) + size(\mathtt{t_2}) + size(\mathtt{t_3})\\
&< size(\mathtt{t}).
\end{aligned}$$

□

■Thress Inductions on Terms

Theorem 9 (3.3.4 Principles of Induction on Terms). *Induction on depth:*

$\forall s \in \mathcal{T}, (\forall r \in \mathcal{T}$ *s.t.* $depth(r) < depth(s), P(r) \to P(s))$
$\to \forall s \in \mathcal{T}, P(s)$.

Induction on size:

$\forall s \in \mathcal{T}, (\forall r \in \mathcal{T}$ *s.t.* $size(r) < size(s), P(r) \to P(s))$
$\to \forall s \in \mathcal{T}, P(s)$.

Structural induction (構造的帰納法):

$\forall s \in \mathcal{T}, (\forall r \in \mathcal{T}$ *s.t.* r *is a immediate subterm of* $s, P(r) \to P(s))$
$\to \forall s \in \mathcal{T}, P(s)$.

■Proof: Exercise (★★)

Maybe use the concrete definition and induction on natural numbers......

$$\begin{aligned}
\mathcal{S}_0 &= \emptyset\\
\mathcal{S}_{i+1} &= \quad \{\mathtt{true,\ false,\ 0}\}\\
&\quad \cup \{\mathtt{succ\ t_1,\ pred\ t_1,\ iszero\ t_1} | \mathtt{t_1} \in \mathcal{S}_i\}\\
&\quad \cup \{\mathtt{if\ t_1\ then\ t_2\ else\ t_3} | \mathtt{t_1,\ t_2,\ t_3} \in \mathcal{S}_i\}
\end{aligned}$$

$$\begin{aligned}
&\text{let } \mathcal{P}_i = (\forall \mathtt{s} \in \mathcal{S}_i, (\forall \mathtt{r} \in \mathcal{S}_i \textit{ s.t. } depth(\mathtt{r}) < depth(\mathtt{s}), P(\mathtt{r}) \to P(\mathtt{s}))\\
&\qquad\qquad \to \forall \mathtt{s} \in \mathcal{S}_i, P(\mathtt{s})).\\
&\mathcal{P}_0 \wedge (\forall i \in \mathbb{N}, \mathcal{P}_i \to \mathcal{P}_{i+1}) \to \forall i \in \mathbb{N}, \mathcal{P}_i. \text{ (induction on natural numbers)}
\end{aligned}$$

■Power of Structural Induction

Structural induction is often easier than other inductions.

Structural Induction. By induction on $\mathtt{t}$.

Case: $\mathtt{t = true}$

…show $P(\mathtt{true})$ …

Case: $\mathtt{t = succ\ t_1}$

…show $P(\mathtt{succ\ t_1})$ using $P(\mathtt{t_1})$ …

Case: $\mathtt{t = if\ t_1\ then\ t_2\ else\ t_3}$

…show $P(\mathtt{if\ t_1\ then\ t_2\ else\ t_3})$ using $P(t_1), P(t_2), P(t_3)$ …

□

3.4 Semantic Styles

■Three Semantic Styles

- Operational semantics (操作的意味論)
- Denotational semantics (表示的意味論)
- Axiomatic semantics (公理的意味論)

■Operational Semantics

Define an *abstract machine* and specify the behavior.

$$\mathtt{if\ true\ then\ t_2\ else\ t_3} \Rightarrow \mathtt{t_2} \qquad \mathtt{if\ false\ then\ t_2\ else\ t_3} \Rightarrow \mathtt{t_3}$$

$$\frac{\mathtt{t_1} \to \mathtt{t_1'}}{\mathtt{if\ t_1\ then\ t_2\ else\ t_3} \Rightarrow \mathtt{if\ t_1'\ then\ t_2\ else\ t_3}}$$

■Denotational Semantics

The meaning of a term is taken to be some mathematical object.

$$[\![0]\!] = 0 \qquad [\![\mathtt{succ\ t}]\!] = [\![t]\!] + 1 \qquad [\![\mathtt{pred\ t}]\!] = [\![t]\!] - 1$$

■Axiomatic Semantics

Closely related to Hoare logic.