

Untyped Arithmetic Expressions: Syntax (I)

$t ::=$	terms:
true	constant true
false	constant false
if t then t else t	conditional
0	constant zero
succ t	successor
pred t	predecessor
iszero t	zero test

1

Syntax (II)

3.2.1: Definition [Terms, Inductively] The set of *terms* is the smallest set \mathcal{T} such that

1. $\{\text{true}, \text{false}, 0\} \subseteq \mathcal{T}$;
2. if $t_1 \in \mathcal{T}$, then $\{\text{succ } t_1, \text{pred } t_1, \text{iszero } t_1\} \subseteq \mathcal{T}$;
3. if $t_1, t_2, t_3 \in \mathcal{T}$, then $\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \in \mathcal{T}$.

3.2.3: Definition [Terms, Concretely] For each natural number i , define a set S_i as follows:

$$\begin{aligned} S_0 &= \emptyset, \\ S_{i+1} &= \{\text{true}, \text{false}, 0\} \\ &\quad \cup \{\text{succ } t_1, \text{pred } t_1, \text{iszero } t_1 \mid t_1 \in S_i\} \\ &\quad \cup \{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \mid t_1, t_2, t_3 \in S_i\}. \end{aligned}$$

Let $S = \bigcup_{i \in \mathbb{N}} S_i$.

2

Inductive (Recursive) Definitions of Functions on Terms

3.3.2: Definition The *size* of a term t , written $\text{size}(t)$, is defined as follows:

$$\text{size}(\text{true}) = 1$$

$$\text{size}(\text{false}) = 1$$

$$\text{size}(0) = 1$$

$$\text{size}(\text{succ } t_1) = \text{size}(t_1) + 1$$

$$\text{size}(\text{pred } t_1) = \text{size}(t_1) + 1$$

$$\text{size}(\text{iszero } t_1) = \text{size}(t_1) + 1$$

$$\text{size}(\text{if } t_1 \text{ then } t_2 \text{ else } t_3) = \text{size}(t_1) + \text{size}(t_2) + \text{size}(t_3) + 1.$$

3

Definitions of Functions (Cont.)

3.3.3: Definition The *depth* of a term t , written $\text{depth}(t)$, is defined as follows:

$$\text{depth}(\text{true}) = 1$$

$$\text{depth}(\text{false}) = 1$$

$$\text{depth}(0) = 1$$

$$\text{depth}(\text{succ } t_1) = \text{depth}(t_1) + 1$$

$$\text{depth}(\text{pred } t_1) = \text{depth}(t_1) + 1$$

$$\text{depth}(\text{iszero } t_1) = \text{depth}(t_1) + 1$$

$$\text{depth}(\text{if } t_1 \text{ then } t_2 \text{ else } t_3) = \max(\text{depth}(t_1), \text{depth}(t_2), \text{depth}(t_3)) + 1.$$

4

Structural Induction on Terms

Suppose P is a predicate on terms, i.e., $P \subseteq \mathcal{T}$.

The *principle of structural induction for terms* says that:

if, for each term s ,
 given $P(r)$ for all immediate subterms r of s ,
 we can show $P(s)$,
then $P(s)$ holds for all $s \in \mathcal{T}$.

5

More Induction Principles for Terms

The *principle of induction on depth for terms* says that:

if, for each term s ,
 given $P(r)$ for all r such that $\text{depth}(r) < \text{depth}(s)$,
 we can show $P(s)$,
then $P(s)$ holds for all $s \in \mathcal{T}$.

The *principle of induction on size for terms* says that:

if, for each term s ,
 given $P(r)$ for all r such that $\text{size}(r) < \text{size}(s)$,
 we can show $P(s)$,
then $P(s)$ holds for all $s \in \mathcal{T}$.

6

Untyped Boolean Expressions: Evaluation

Syntax

$t ::=$	terms:
true	constant true
false	constant false
if t then t else t	conditional
$v ::=$	values:
true	true value
false	false value

Evaluation: $t \rightarrow t'$

if true then t_2 else $t_3 \rightarrow t_2$ (E-IfTrue)

if false then t_2 else $t_3 \rightarrow t_3$ (E-IfFalse)

$$\frac{t_1 \rightarrow t'_1}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \rightarrow \text{if } t'_1 \text{ then } t_2 \text{ else } t_3} \quad (\text{E-If})$$

7

Untyped Boolean Expressions: Evaluation (Cont.)

Suppose $P \subseteq \mathcal{T} \times \mathcal{T}$. We sometimes write “ $P(t_1, t_2)$ ” for “ $(t_1, t_2) \in P$ ”. The *principle of induction on \rightarrow* says that,

for all $t_1, t_2 \in \mathcal{T}$, if $t_1 \rightarrow t_2$, then $P(t_1, t_2)$,

follows from showing

(E-IfTrue) for all $t_2, t_3 \in \mathcal{T}$, $P(\text{if true then } t_2 \text{ else } t_3, t_2)$;

(E-IfFalse) for all $t_2, t_3 \in \mathcal{T}$, $P(\text{if false then } t_2 \text{ else } t_3, t_3)$;

(E-If) for all $t_1, t'_1, t_2, t_3 \in \mathcal{T}$, if $t_1 \rightarrow t'_1$ and (†) $P(t_1, t'_1)$, then $P(\text{if } t_1 \text{ then } t_2 \text{ else } t_3, \text{if } t'_1 \text{ then } t_2 \text{ else } t_3)$.

We refer to (†) as the *inductive hypothesis*.

Untyped Arithmetic Expressions: Evaluation

New Syntactic Forms

$t ::= \dots$	terms:
0	constant zero
$\text{succ } t$	successor
$\text{pred } t$	predecessor
$\text{iszero } t$	zero test
$v ::= \dots$	values:
nv	numeric value
$nv ::=$	numeric values:
0	zero value
$\text{succ } nv$	successor value

9

Untyped Arithmetic Expressions: Evaluation (Cont.)

New Evaluation Rules: $t \rightarrow t'$

$\frac{t_1 \rightarrow t'_1}{\text{succ } t_1 \rightarrow \text{succ } t'_1}$	(E-Succ)
$\text{pred } 0 \rightarrow 0$	(E-PredZero)
$\text{pred } (\text{succ } nv_1) \rightarrow nv_1$	(E-PredSucc)
$\frac{t_1 \rightarrow t'_1}{\text{pred } t_1 \rightarrow \text{pred } t'_1}$	(E-Pred)
$\text{iszero } 0 \rightarrow \text{true}$	(E-IszeroZero)
$\text{iszero } (\text{succ } nv_1) \rightarrow \text{false}$	(E-IszeroSucc)
$\frac{t_1 \rightarrow t'_1}{\text{iszero } t_1 \rightarrow \text{iszero } t'_1}$	(E-Iszero)

10