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

Syntax (II)

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

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

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

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

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

Inductive (Recursive) Definitions of Functions on Terms

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

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

Definitions of Functions (Cont.)

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

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

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 $s \in \mathcal{T}$.

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 $\mathrm{depth}(r) < \mathrm{depth}(s)$,
> >
> > 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 $\mathrm{size}(r) < \mathrm{size}(s)$,
> >
> > we can show $P(s)$,
>
> then $P(s)$ holds for all $s \in \mathcal{T}$.

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: $\boxed{t \to t'}$

$$\text{if true then } t_2 \text{ else } t_3 \to t_2 \qquad \text{(E-IfTrue)}$$

$$\text{if false then } t_2 \text{ else } t_3 \to t_3 \qquad \text{(E-IfFalse)}$$

$$\frac{t_1 \to t_1'}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \to \text{if } t_1' \text{ then } t_2 \text{ else } t_3} \qquad \text{(E-If)}$$

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* $\longrightarrow$ says that,

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

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 $t_1 \longrightarrow t_1'$ 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

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

Untyped Arithmetic Expressions: Evaluation (Cont.)

New Evaluation Rules: $\boxed{t \longrightarrow t'}$

$$\frac{t_1 \longrightarrow t_1'}{\mathsf{succ}\ t_1 \longrightarrow \mathsf{succ}\ t_1'} \qquad \text{(E-Succ)}$$

$$\mathsf{pred}\ 0 \longrightarrow 0 \qquad \text{(E-PredZero)}$$

$$\mathsf{pred}\ (\mathsf{succ}\ nv_1) \longrightarrow nv_1 \qquad \text{(E-PredSucc)}$$

$$\frac{t_1 \longrightarrow t_1'}{\mathsf{pred}\ t_1 \longrightarrow \mathsf{pred}\ t_1'} \qquad \text{(E-Pred)}$$

$$\mathsf{iszero}\ 0 \longrightarrow \mathsf{true} \qquad \text{(E-IszeroZero)}$$

$$\mathsf{iszero}\ (\mathsf{succ}\ nv_1) \longrightarrow \mathsf{false} \qquad \text{(E-IszeroSucc)}$$

$$\frac{t_1 \longrightarrow t_1'}{\mathsf{iszero}\ t_1 \longrightarrow \mathsf{iszero}\ t_1'} \qquad \text{(E-Iszero)}$$