# Syntactic Objects

# Surface Syntax

if true then false else true

# Surface Syntax

sort

if true then false else true : Exp

# Surface Syntax

sort

if true then false else true : Exp

true : Exp

false : Exp

# Surface Syntax

$$\mathsf{if} \ - \ \mathsf{then} \ - \ \mathsf{else} - : (\mathsf{Exp} \times \mathsf{Exp} \times \mathsf{Exp}) \to \mathsf{Exp}$$

$$\mathsf{true} : \mathsf{Exp}$$

$$\mathsf{false} : \mathsf{Exp}$$

$$\mathsf{if\ true\ then\ false\ else\ true} : \mathsf{Exp}$$

# Abstract Syntax

$$\mathsf{if}(\, - \,,\; - \,, - \,) : (\mathsf{Exp} \times \mathsf{Exp} \times \mathsf{Exp}) \rightarrow \mathsf{Exp}$$

$$\mathsf{true} : \mathsf{Exp}$$

$$\mathsf{false} : \mathsf{Exp}$$

$$\mathsf{if}(\mathsf{true}, \mathsf{false}, \mathsf{true}) : \mathsf{Exp}$$

# Abstract Syntax Trees

$$\mathsf{plus} : \mathsf{Exp} \times \mathsf{Exp} \to \mathsf{Exp}$$

$$\mathsf{num} : \mathbb{N} \to \mathsf{Exp}$$

$$\mathsf{num}[42] : \mathsf{Exp}$$

$$\mathsf{num}[4] : \mathsf{Exp}$$

$$\mathsf{plus}(\mathsf{num}[4], \mathsf{num}[42]) : \mathsf{Exp}$$

# Abstract Syntax Trees: Variables

$$\mathsf{times} : \mathsf{Exp} \times \mathsf{Exp} \to \mathsf{Exp}$$

$$\mathsf{num} : \mathbb{N} \to \mathsf{Exp}$$

variable $\longrightarrow$ $x : \mathsf{Exp}$

$$\mathsf{num}[4] : \mathsf{Exp}$$

$$\mathsf{times}(x, \mathsf{num}[4]) : \mathsf{Exp}$$

# Abstract Syntax Trees: Variable Substitution

$$\mathsf{plus} : \mathsf{Exp} \times \mathsf{Exp} \to \mathsf{Exp}$$

$$\mathsf{times} : \mathsf{Exp} \times \mathsf{Exp} \to \mathsf{Exp}$$

$$\mathsf{num} : \mathbb{N} \to \mathsf{Exp}$$

$$\mathsf{num}[42] : \mathsf{Exp}$$

$$\mathsf{num}[4] : \mathsf{Exp}$$

$$\mathsf{times}(\mathsf{plus}(\mathsf{num}[4], \mathsf{num}[42]), \mathsf{num}[4]) : \mathsf{Exp}$$

# Abstract Syntax Trees: Variable Substitution

$$\mathsf{plus} : \mathsf{Exp} \times \mathsf{Exp} \rightarrow \mathsf{Exp}$$

$$\mathsf{times} : \mathsf{Exp} \times \mathsf{Exp} \rightarrow \mathsf{Exp}$$

$$\mathsf{num} : \mathbb{N} \rightarrow \mathsf{Exp}$$

$$\mathsf{num}[42] : \mathsf{Exp}$$

$$\mathsf{num}[4] : \mathsf{Exp}$$

$$\mathsf{times}(\mathsf{plus}(\mathsf{num}[4], \mathsf{num}[42]), \mathsf{num}[4]) : \mathsf{Exp}$$

# Abstract Syntax Trees Defined

Let $S$ be a finite set of **sorts**, $\{O_s\}_{s \in S}$ be a sort-indexed family of operators $o$ of sort $s$ with arity $\mathsf{ar}(o) = (s_1, \ldots, s_n)$, and $\{\chi_s\}_{s \in S}$ be a sort-indexed family of variables $x$ of sort $s$. The family $A[\chi] = A[\chi_s]_{s \in S}$ of abstract syntax trees (ASTs) of sort $s$ is defined as follows:

- if $x \in \chi_s$, then $x \in A[\chi]_s$
- if $o \in O_s$ with $\mathsf{ar}(o) = (s_1, \ldots, s_n)$ and $a_1 \in A[\chi]_{s_1}, \ldots, a_n \in A[\chi]_{s_n}$, then $o(a_1, \ldots, a_n) \in A[\chi]_s$

# Abstract Syntax Trees Defined

Let $S$ be a finite set of **sorts**, $\{O_s\}_{s \in S}$ be a sort-indexed family of operators $o$ of sort $s$ with arity $\mathsf{ar}(o) = (s_1, \ldots, s_n)$, and $\{\chi_s\}_{s \in S}$ be a sort-indexed family of variables $x$ of sort $s$. The family $A[\chi] = A[\chi_s]_{s \in S}$ of abstract syntax trees (ASTs) of sort $s$ is defined as follows:

- if $x \in \chi_s$, then $x \in A[\chi]_s$
- if $o \in O_s$ with $\mathsf{ar}(o) = (s_1, \ldots, s_n)$ and $a_1 \in A[\chi]_{s_1}, \ldots, a_n \in A[\chi]_{s_n}$, then $o(a_1, \ldots, a_n) \in A[\chi]_s$

Exercise: Come up with three example ASTs.

# Abstract Syntax Trees: Structural Induction

To show that a property P for every AST it suffices to show $\mathsf{P}(a)$ holds for every $a \in A[\chi]$, which holds when:

1. (Base Case) if $x \in \chi_s$, then $\mathsf{P}_s(x)$, and
2. (Step Case) if $o \in O_s$ with $\mathsf{ar}(o) = (s_1, \ldots, s_n)$ , then if $\mathsf{P}_{s_1}(a_1), \ldots, \mathsf{P}_{s_n}(a_n)$ all hold, then $\mathsf{P}_s(o(a_1, \ldots, a_n))$ holds.

# Abstract Syntax Trees: Structural Induction

**Lemma:** If $\mathcal{X} \subseteq \mathcal{Y}$, then $A[\mathcal{X}] \subseteq A[\mathcal{Y}]$.

Proof. By structural induction.

1. (Base Case) If $x \in \mathcal{X}_s$ which implies that $x \in A[\mathcal{X}]_s$, then by assumption $x \in \mathcal{Y}$, and hence, by definition $x \in A[\mathcal{Y}]_s$.

2. (Step Case) Suppose $o \in \mathcal{O}_s$, $\mathsf{ar}(o) = (s_1, \ldots, s_n)$, $\mathcal{X} \subseteq \mathcal{Y}$. Then by induction: $a_1 \in A[\mathcal{X}]_{s_1} \iff a_1 \in A[\mathcal{Y}], \ldots, a_n \in A[\mathcal{X}] \iff a_n \in A[\mathcal{Y}]_{s_n}$. Then by definition $o(a_1, \ldots, a_n) \in A[\mathcal{X}]_s \iff o(a_1, \ldots, a_n) \in A[\mathcal{Y}]_s$.

# Abstract Syntax Trees: Substitution

Variables are given their meaning through substitution.

$$
\begin{aligned}
&[\mathsf{num}[42]/x](\mathsf{plus}(x, \mathsf{mult}(\mathsf{num}[3], x)) \\
&\quad = \mathsf{plus}([\mathsf{num}[42]/x]x, [\mathsf{num}[42]/x]\mathsf{mult}(\mathsf{num}[3], x)) \\
&\quad = \mathsf{plus}([\mathsf{num}[42]/x]x, \mathsf{mult}([\mathsf{num}[42]/x]\mathsf{num}[3], [\mathsf{num}[42]/x]x)) \\
&\quad = \mathsf{plus}(\mathsf{num}[42], \mathsf{mult}(\mathsf{num}[3], \mathsf{num}[42]))
\end{aligned}
$$

# Abstract Syntax Trees: Substitution

Substitution $[b_1/x]b_2 \in A[\mathscr{X}]_{s_2}$ on any AST $A[\mathscr{X}]$ where $x$ is a variable of sort $s_1$, $b_1 \in A[\mathscr{X}]_{s_1}$, $b_2 \in A[\mathscr{X}, x]_{s_2}$ is defined as follows:

1. $[b_1/x]x = b_1$
2. $[b_1/x]y = y$, when $x \neq y$
3. $[b_1/x]o(a_1, \ldots, a_n) = o([b_1/x]a_1, \ldots, [b_1/x]a_n)$

# Abstract Syntax Trees: Extension

Let $\mathscr{X}$ be a sort-indexed family of variables. Then: $(\mathscr{X}, x)$ where $x$ is a variable of sort $s$ such that $x \notin \mathscr{X}_s$, to stand for the sort-indexed family $\mathscr{Y}$ such that $\mathscr{Y}_s = \mathscr{X}_s \cup \{x\}$ and $\mathscr{Y}_{s'} = \mathscr{X}_{s'}$ for all $s' \neq s$.

This is also known a **adjoining.**