Grammar express trees
We can define specific languages with our grammars
We can perform induction on our grammars
We can write inductive relations as inference rules

```
Exp : e := e + e
    | e * e
    | e / e
```

regular tree grammars
\[ L = \{ e \} \]

\[ L(e) = n \in \text{Exp} \times R \]

Base: \[ e = n \in R \]

\[ \{ e \} = n \]

\[ \{ e \} = n \in \text{Exp} \times R \to \text{conclusion} \]

Plus: \[ \{ e_1 \} = n \]

\[ \{ e_1 \} = n \in \text{Exp} \times R \]

\[ \{ e_1 + e_2 \} = n_1 + n_2 \in R \times R \]

\[ (e_1 \otimes e_2) = \\]
For a relation $R = X \times Y$

to be a function, you must have the following:

1) For all $x \in X$, $\exists y \in Y$. $R(x, y)$

2) If $R(x, y)$ and $R(x, y')$ then $y = y'$

Then: Let $\forall x \in \mathbb{R}$, $\exists \theta$. Let $\theta := n$

By induction over $e$:

Case: $e = \forall x \in \mathbb{R}$

$\exists a \in \mathbb{R}$

$\forall x \in \mathbb{R}$
\[ C \cap B = \emptyset \]

Case: \( e \cap e \cap \emptyset \)

\[
\begin{align*}
\text{IH} & \quad \text{IH} \\
C_e \cap B = n_1 & \quad C_{e \cap e} = n_2 \\
\hline
\text{Plus} & \quad \text{IH}
\end{align*}
\]

\[ C_{e \cap e \cap \emptyset} = n_1 + n_2 \]

Thus: If \( C_e \cap B = n \) and \( C_{e \cap e} = n' \) then \( n = n' \).

**Proof:** By induction over the decimation of \( C_e \cap B = n \)

Case: the last code was bad

Example:

\[ C_e \cap B = n \quad \text{then} \quad e \in \emptyset \]
Let's start with: \[ 2x + (3 + 4) \]

Deduction

Deduction

Aside: What if we have an additional rule \[ \Phi \text{ or } \Box \]

We would say by inversion, the last rule is \[ \Phi \text{ or } \Box \]

Inversion is the shorthand for the argument:

We have concluded via some rule that \( \Phi \). The only rules that make a conclusion that looks like this are: \( \Phi \text{ or } \Box \)

In \( T = n' \)

by induction, the last rule was \( \Box \)

\[ T = n' \text{ then } n = n' \]
Case 1:

\[
\frac{\Sigma e_1 \cdot n_1}{\Sigma e_2 \cdot n_2} = r \\Rightarrow \Sigma e_1 + e_2 = n_1 + n_2
\]

\[
\Sigma e_1 + e_2 = n_1
\]

By inspection, the last rule applied was plus.

\[
\Sigma e_1 + e_2 = n_1 + n_2
\]

\[
\Sigma e_1 \cdot n_1 = n_1
\]

\[
\Sigma e_1 \cdot n_1 = n_1
\]

\[
\Sigma e_1 \cdot n_1 = n_1
\]
So by \( I_4 \)
\[ n_1^{'} = n_1^{'} \]

\[ \mathcal{A}_{2,1} = n_2^{'} \]
\[ \mathcal{C}_{2,1} = n_2^{'} \]

So \( \mathcal{A} = \mathcal{C} \)

\[ n = n_1^{'} = n_2^{'} = n^{'} \]

\[ n^{'} = n^{'} \]

\[ \mathcal{E} = \_ \]

\[ u = 0 \]

\[ \text{acts on } S \]

\[ n = 0 \]
\( \text{three}(0) \)

\( \text{three}(n) \)

\( \text{three}(\text{SSS } n) \)

\( \text{three}(\text{SSSS } n) \)

\( \text{step}(0) \)

\( \text{step}(n) \)

\( \text{step}(\text{SSSSS } n) \)

Thus, \( \text{step}(n) \Rightarrow \text{three}(n) \)

My induction over the definition of \( \text{step}(n) \)

Base:

\( \text{zero}(n) \) \( \Rightarrow \) \( n \equiv 0 \)

\( \text{three}(n) \)
Index

case 1, last row 1

\[
\frac{\text{IHS} \cdot n'}{5 \text{S}55555} \quad \text{say I}
\]

by IH, I am true \( n' \)

\[
\frac{\text{three} \cdot n'}{5 \text{S}55} \quad \text{true I}
\]

\[
\frac{\text{three} \cdot 5 \text{SSS} \cdot n'}{5 \text{S}55555} \quad \text{true} \text{I}
\]