“The *denotational semantics* of a language defines the meaning of abstract language elements as a collection of state-transforming functions.”

–Tucker and Noonan, 2007
Abstract Syntax Review

Abstract syntax is a notation that allows parser to remove nonessential symbols and generate a tree that contains only the essential elements of the computation.

Program = Declarations decpart; Statements body
Program State

A variable and its value can be modeled as an ordered pair, e.g. \( \langle i, 5 \rangle \)

A state is a set of ordered pairs

\[
\text{state} = \{ \langle \text{var}_1, \text{val}_1 \rangle, \langle \text{var}_2, \text{val}_2 \rangle, \ldots \langle \text{var}_m, \text{val}_m \rangle \}\]
Example: Factorial

```c
int main () {
    int n, i, f;
    n = 3;
    i = 1;
    f = 1;
    while (i < n) {
        i = i + 1;
        f = f * i;
    }
    return 0;
}
```

<table>
<thead>
<tr>
<th>Step</th>
<th>Before Statement</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>{&lt;n, undef&gt;, &lt;i, undef&gt;, &lt;f, undef&gt;}</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>{&lt;n, 3&gt;, &lt;i, undef&gt;, &lt;f, undef&gt;}</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>{&lt;n, 3&gt;, &lt;i, 1&gt;, &lt;f, undef&gt;}</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>{&lt;n, 3&gt;, &lt;i, 1&gt;, &lt;f, 1&gt;}</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>{&lt;n, 3&gt;, &lt;i, 1&gt;, &lt;f, 1&gt;}</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>{&lt;n, 3&gt;, &lt;i, 2&gt;, &lt;f, 1&gt;}</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>{&lt;n, 3&gt;, &lt;i, 2&gt;, &lt;f, 2&gt;}</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>{&lt;n, 3&gt;, &lt;i, 2&gt;, &lt;f, 2&gt;}</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>{&lt;n, 3&gt;, &lt;i, 3&gt;, &lt;f, 2&gt;}</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>{&lt;n, 3&gt;, &lt;i, 3&gt;, &lt;f, 6&gt;}</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
<td>{&lt;n, 3&gt;, &lt;i, 3&gt;, &lt;f, 6&gt;}</td>
</tr>
</tbody>
</table>
Meaning Function of Clite

Let $State$ represent the set of all program states, then the meaning $M$ of Clite notations are defined by:

- $M : Program \rightarrow State$
- $M : Statement \times State \rightarrow State$
- $M : Expression \times State \rightarrow Value$
The meaning of a Program is defined to be the meaning of its body when given an initial state consisting of the variables of \texttt{decpart}, each initialized to the undef value corresponding to its declared type.

\[ M : \text{Program} \rightarrow \text{State} \]
\[ M(\text{Program } p) = M(p.\text{body}, \text{InitialState}(p.\text{decpart})) \]
$M(\text{Program } p) = M(p.\text{body}, \text{InitialState}(p.\text{decpart}))$

- \text{InitialState} is an auxiliary function that \textit{creates a state containing all the variables} declared in \texttt{p.decpart} paired with their \textit{default values}.

- Example:

  If \texttt{int x, y} is \texttt{p.decpart}, then

  \[
  \text{InitialState}(p.\text{decpart}) = \{ \langle x, \text{undef} \rangle, \langle y, \text{undef} \rangle \} \]