Outline for a One-Pass Load-and-Go Assembler

Note: The following outline shows only how the modified symbol table (ST) is used in the one-pass assembler. The other details are similar to those for the two-pass assembler.

I. Symbol seen in the operand field:

Note: In the following description, L denotes the LC value of the current instruction and X denotes the symbol seen in the operand field.

```
<table>
<thead>
<tr>
<th>Is X in the Symbol Table (ST)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
</tr>
</tbody>
</table>
```

<table>
<thead>
<tr>
<th>Is X defined?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
</tr>
</tbody>
</table>

Use the LC value of X

See Item A below

See Item B below

---

**Item A:**

1. Insert X into ST with Defined = false.

2. Create a node containing the LC value L and let that node be the first node of the linked list corresponding to X in ST.

**Item B:**

1. Create a node containing the LC value L.

2. Insert the node into the linked list corresponding to X in ST.
II. Symbol seen in the label field:

Note: In the following description, L denotes the LC value of the current instruction and X denotes the symbol seen in the label field.

![Decision Tree]

Item A:

1. Insert X into ST with Defined = true and LC value = L.
2. Set the linked list’s head pointer to NULL. (We won’t need a list for the symbol X.)

Item B:

1. In the ST, change the Defined value for X to true and the LC value for X to L.
2. Patch bytes using the linked list for X.
3. Delete the nodes in the linked list for X and set the head pointer to NULL. (From this point, we won’t need a list for the symbol X.)