1 Computer Science, Algorithms, & Programs

Some definitions:

- computer science: the study of algorithms and their implementation
- program: a sequence of instructions expressed in a language that can be executed directly, or indirectly, by a computer

Key takeaways:

- Python is a high level language. To run a python program on your computer, you need another program! That program will *translate* your python code into a low-level language that the computer can understand.
- There is a difference between an algorithm and a program.
- When solving problems in this class, you must first design an algorithm. Once you have an algorithm, you can think about translating into a computer program.
- Novice programmers tend too spend too little time thinking carefully about their algorithm.

2 Picobot cheatsheet

Detailed instructions are provided with your homework... this is just a quick cheatsheet for class.

**Surroundings**: are always considered in NEWS order. An x represents empty space, the appropriate direction letter (N, E, W, and S) represents a wall blocking that direction. For example, NxWS means there is a wall in every direction but East.

**State**: picobot’s state is simply a number between 0 and 99. Picobot always starts in state 0.

**Rules**: Picobot rules look like this:

<table>
<thead>
<tr>
<th>StateNow</th>
<th>Surroundings</th>
<th>MoveDirection</th>
<th>NewState</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>xxxS</td>
<td>N</td>
<td>1</td>
</tr>
</tbody>
</table>

This rule means “if Picobot starts in state 0 and sees the surroundings xxxS, it should move North and change to state 1.”

**Wildcards**: the asterisk * can be used inside surroundings to mean “I don’t care whether there is a wall or not in that position.” For example,

<table>
<thead>
<tr>
<th>StateNow</th>
<th>Surroundings</th>
<th>MoveDirection</th>
<th>NewState</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>xE**</td>
<td>N</td>
<td>3</td>
</tr>
</tbody>
</table>

means “if there is no wall North and there is a wall to the East and I don’t care about West or South, then move North and change to state 3.”
Exercises

Some picobot problems have been presented in class. Try this problem on your own.

1. Create a program that will get picobot to the SOUTHEAST corner of the empty room. Your rules should work regardless of Picobot’s starting position!

Write your program here:

\[
\text{StateNow} \quad \text{Surroundings} \quad \rightarrow \quad \text{MoveDirection} \quad \text{NewState}
\]