In Search of Strong AI

A journal of thoughts and explorations on the subject by Sean Patrick Hannifin

~ Part 1 ~
Foundations of Learning

1st edition, written January 2013

Introduction

My goal is to achieve Strong AI. I would like to find a set of algorithms that will allow a computer program to learn *any* given task.

The is obviously a lofty goal. But it is not impossible. After all, we humans learn things and create knowledge. What algorithms are our brains using to do this? There *must* be an answer.

What follows are my attempts at a *start* to an answer. I share these thoughts for anyone interested in following my progress, or interested in contributing thoughts to the matter.

I obviously do not believe there is anything to fear in achieving or understanding Strong AI, or the foundations of how human intelligence works. Robots will not try to take over the world. Humanity will only benefit from an understanding.

If the ideas I describe below seem familiar, please let me know if anyone has ever proposed these principles before so that I might study his work. I have created my own terms and definitions for my explorations, but it is possible I am merely and unknowingly reinventing the wheel. Perhaps these ideas have already been explored by some philosopher a century ago. If so, the philosopher is obscure to me and I would certainly benefit from knowing his name.

The terms I have created will be introduced in bold, and capitalized when mentioned.

Elements of Knowledge

I will call this system, by which I mean the following set of terms, definitions, and algorithms, the **Truth Searcher**.

The Truth Searcher works in binary. A given statement is either true or false. We make a **Guess** as to whether a statement is true or false based on our perceptions and what we've learned so far.

A **Perception** is also either true or false, and can be represented by a variable, such as *A*, *B*, *C*, etc. A Perception can be any binary answer (such as yes or no, or true or false) to any binary question about an observation. For example: Is the sky blue? Can I see clouds? Is the temperature below freezing? Note that a Perception *must* be falsifiable. A Perception that we can never know to be true or false is useless.

With multiple Perceptions, we have a **Perception Set**, such as *ABCD* (each of which is either true or false), and we need a way to make a Guess (true or false) based on these Perceptions.

To decide whether our Guess will be true or false, we need a **Theory**. A Theory is the set of Perceptions that must be true for us to make our Guess true.

For example, our Theory may be *AC*. So if Perception *A* is true and Perception *C* is true, then our Guess is true, and our other Perceptions (*B* and *D*) don't matter; they are just noise. Of course, our Theory could be wrong, so our Guess could be wrong. There is no way to guarantee our Theory is right; we can only improve our Theory when we know that it is wrong.

If our Theory is *AC*, this is a **Truth Set**, a set of Perceptions that must be true for our Guess to be true. A Theory may be made of multiple Truth Sets. For example, our Theory might be *AC* or *AD*. We can express our Theory like this:

AC || AD

In this example, *AC* is Truth Set 1, and *AD* is Truth Set 2. (Of course, the '||' means 'or'.)

Learning

We improve our Theory by learning, and we learn by making an **Observation** and discovering that our Guess is wrong. (We learn *nothing* when our Theory leads to a Guess that is correct, a Guess that matches our Observation. Note that this does *not* mean our Theory is correct. Again, we can *never* guarantee that, but we *can* get closer to the truth by correcting our mistakes. In this way, all noticeable mistakes provide us with learning experiences. You can see that this is very similar to the scientific method, eh?)

We begin with the null hypothesis. That is, we begin with no Truth Sets. Our default Guess is false. (True and false can of course be reversed. That is, our default Guess could be true and our Truth Sets could be False Sets instead.)

There are two ways to make a mistake:

1. Our Guess is false but our Observation is true.
2. Our Guess is true but our Observation is false.

Now let us describe how we can correct our mistakes.

When our Guess is false but our Observation is true, we create a new Truth Set based on all Perceptions that are true. After all, we know that our Observation was true for some reason.

If the Truth Set we add is already found in an existing Truth Set in our Theory, we *split* that Truth Set into two Truth Sets: one Truth Set for the Perceptions that were true, and one for the Perceptions that were false.

If we split a Truth Set, we need to remember what Perceptions we are splitting from it just in case we are splitting wrongfully. A wrongful Truth Set split happens when there are more than one Truth Sets we could split. We don't know which one we *should* split, so we split all of them and add the Perceptions we split to a **Split List**, a list for each Truth Set to keep track of the Perceptions we split from it. If we later discover that a split was wrongful, we will remember what Perceptions to add back.

When our Guess is true but our Observation is false, we eliminate the Truth Set that made our Guess true. When we do this, we must put the Truth Set on an **Ignore List** so that we do not end up adding it to our Theory again later. (Without an Ignore List, our learning algorithm could be forced into an infinite loop. Because an Ignore List can get rather large, this could explain why some tasks are difficult for humans to learn. We have to remember to ignore certain perceptions and focus on other ones.)

After each step, we must purge any Truth Sets in our Theory that are on our Ignore List.

Finally, we must also purge *redundant* Truth Sets after we make a correction to our Theory. For example, suppose we have the following Theory:

ACD || EF || E

In this example, Truth Set 2 is redundant because our Guess will be true is Perception *E* is true, as given by Truth Set 3. Whenever a Truth Set appears in another Truth Set, we can purge the larger one.

That's really all there is to it.

Limitations

Our Truth Searcher is simple enough, and is powerfully efficient, in my opinion. I believe it provides an important foundation in the search for Strong AI. But it is certainly incomplete, as there are some obvious limitations.

The main limitation is that our Theories are limited by available Perceptions. There is no way to create a Perception, and part of knowledge creation, or Theory creation, probably deals with searching for the right Perceptions to consider. Possible Perceptions may be infinite.

To demonstrate this limitation, suppose an Observation for a given statement is true if *any* three of our Perceptions is true. To form a proper Theory for such a truth, our Truth Searcher would have to learn *all possible* combinations of three Perceptions, a list which can be rather excessive, if not so completely computationally inefficient that the procedure is useless. (It can be shown that our Truth Searcher could learn the game of chess, but only by solving the game by brute force, which is not very helpful, nor does it represent how humans learn the game. So Perception creation must be part of the human learning process, and will probably be needed for Strong AI.)

Another limitation is that our Truth Searcher has no way of dealing with probabilities, or decisions involving chance. That is, how do we know when the Perceptions we are considering are not enough to make a useful Theory?

Another limitation is that, for each task we wish to learn, we must figure out how to express knowledge and observations as binary sets. If we want our Truth Finder to learn to play chess, how do we do it? (Again, I suspect the answer lies in creating a binary Perception, but how do we do that?)

I do not yet know the answers to these questions, but I am working on them.

Examples

I have created a computer program to simulate the workings of our Truth Searcher. (The point of searching for Strong AI is to create useful computer programs, after all.)

Instead of using letter variables for Perceptions, my computer program uses numbers. There are eight Boolean variables, which are represented by the numbers 0 through 7. (Of course, the amount of variables is arbitrary, but we will keep it small so that the procedures are easier to follow.) My program sets each Perception to a random Boolean value before making a Guess. My program would be more efficient if it used another method to seek out mistakes, but random Boolean values will serve well enough for demonstration purposes.

For the first example, I set the Observation to be true if Perception *0* is true. Therefore, the optimal Theory to find is simply: *0*.

Here's the simulation:

```
Test #0:
0:true | 1:false | 2:true | 3:false | 4:true | 5:true | 6:false | 7:false |
Guess: false, Observe: true
0 2 4 5 ||
0 2 4 5 ||
```

Our program began with a null hypothesis, thus the Guess was false. But Perception *0* was true, so our Observation was true. Our program added a Truth Set containing all the Perceptions that were true: *0245*. It also pruned the Truth Sets, but of course there was nothing to prune, so our Theory remained the same. (Bold Theories represent the pruned Theory, the final Theory after that test.)

Variable order obviously doesn't matter; our program sometimes disorders them because of how it iterates through them when splitting, but I will refer to them in numerical order.

You will notice each test is numbered. Because there is no way to guarantee our Theory is correct, our program simply stops when Guesses match Observations for a certain number of tests in a row. In these examples, I will skip all the tests that are Observed to be correct, as our program learns nothing from those tests. That is, no changes are made to our Theory after such tests. You can see how many times our Theory worked for a random set of Booleans by observing the change in test numbers.

Finally, notice that *every* time we make a change in our Theory, our Theory continues to hold for *all* previous Observations.

Moving on:

```
Test #1:
0:true | 1:true | 2:true | 3:false | 4:true | 5:false | 6:false | 7:false |
Guess: false, Observe: true
0 2 4 5 || 0 1 2 4 ||
0 2 4 5 || 0 1 2 4 ||
```

Our program adds Truth Set *0124*.

```
Test #3:
0:true | 1:false | 2:false | 3:true | 4:true | 5:false | 6:true | 7:false |
Guess: false, Observe: true
0 2 4 5 || 0 1 2 4 || 0 3 4 6 ||
0 2 4 5 || 0 1 2 4 || 0 3 4 6 ||
```

Our program adds Truth Set *0346*.

```
Test #5:
0:true | 1:false | 2:false | 3:false | 4:true | 5:false | 6:false | 7:false |
Guess: false, Observe: true
2 5 || 1 2 || 3 6 || 4 0 || 4 0 || 4 0 ||
2 5 || 1 2 || 3 6 || 4 0 ||
```

Our program splits all three Truth Sets based on true Perceptions *0* and *4*, creating three new Truth Sets that are the same: *04*. Because they are redundant, two of them are removed.

```
Test #6:
0:false | 1:false | 2:true | 3:true | 4:true | 5:true | 6:false | 7:true |
Guess: true, Observe: false
2 5 || 1 2 || 3 6 || 4 0 ||
1 2 || 3 6 || 4 0 ||
```

Our program removes Truth Set *25*. This is because Truth Set *25* made our Guess true, but we Observed false, so we know Truth Set *25* cannot be right.

```
Test #7:
0:false | 1:true | 2:true | 3:true | 4:false | 5:false | 6:false | 7:false |
Guess: true, Observe: false
1 2 || 3 6 || 4 0 ||
3 6 || 4 0 ||
```

Our program removes Truth Set *12*.

```
Test #8:
0:true | 1:false | 2:false | 3:true | 4:false | 5:true | 6:false | 7:false |
Guess: false, Observe: true
3 6 || 4 0 || 0 3 5 ||
3 6 || 4 0 || 0 3 5 ||
```

Our program adds Truth Set *035*.

```
Test #13:
0:false | 1:false | 2:false | 3:true | 4:true | 5:true | 6:true | 7:false |
Guess: true, Observe: false
3 6 || 4 0 || 0 3 5 ||
4 0 || 0 3 5 ||
```

Our program removes Truth Set *36*.

```
Test #15:
0:true | 1:false | 2:true | 3:true | 4:false | 5:false | 6:false | 7:false |
Guess: false, Observe: true
4 0 || 0 3 5 || 0 2 3 ||
4 0 || 0 3 5 || 0 2 3 ||
```

Our program adds Truth Set *023*.

```
Test #28:
0:true | 1:true | 2:false | 3:true | 4:false | 5:false | 6:false | 7:true |
Guess: false, Observe: true
4 0 || 0 3 5 || 0 2 3 || 0 1 3 7 ||
4 0 || 0 3 5 || 0 2 3 || 0 1 3 7 ||
```

Our program adds Truth Set *0137*.

```
Test #32:
0:true | 1:false | 2:false | 3:false | 4:false | 5:false | 6:false | 7:true |
Guess: false, Observe: true
4 || 3 5 || 2 3 || 1 3 || 0 || 0 || 0 || 7 0 ||
4 || 3 5 || 2 3 || 1 3 || 0 ||
```

Our program splits all four Truth Sets based on true Perceptions *0* and *7*, creating new Truth Sets *0* and *07*. All but one of these Truth Sets is redundant, so remove them, leaving us with Truth Set *0*.

We now have the only Truth Set we actually need, Truth Set *0*. However, based on our program's previous tests, it can still not eliminate the other Truth Sets; as far as our program knows, they could still be true.

```
Test #34:
0:false | 1:false | 2:false | 3:true | 4:false | 5:true | 6:false | 7:false |
Guess: true, Observe: false
4 3 6 || 3 5 || 2 3 || 1 3 || 0 ||
4 3 6 || 2 3 || 1 3 || 0 ||
```

Our program removes Truth Set *35*.

```
Test #35:
0:false | 1:false | 2:true | 3:true | 4:true | 5:true | 6:true | 7:true |
Guess: true, Observe: false
4 3 6 || 2 3 || 1 3 || 0 ||
1 3 || 0 ||
```

Our program removes Truth Sets 4*36* and *23*.

```
Test #37:
0:false | 1:true | 2:false | 3:true | 4:false | 5:false | 6:false | 7:true |
Guess: true, Observe: false
1 3 || 0 ||
0 ||
```

Our program removes Truth Set *13*.

And after only thirty-eight random tests, our program has found the Theory that matches our Observation's algorithm! Of course, our program has no way of *proving* this, and may continue testing, but the Theory will obviously hold up for *all* future tests.

Now let's have our program find a more complicated Theory, perhaps: *2 || 14 || 345*. This time, I will simply copy the test results; you should be able to figure out what our program is doing after each test, because I know you are a wise and clever reader.

```
Test #0:
0:false | 1:true | 2:false | 3:false | 4:true | 5:false | 6:true | 7:true |
Guess: false, Observe: true
1 4 6 7 ||
1 4 6 7 ||

Test #1:
0:false | 1:true | 2:false | 3:false | 4:true | 5:false | 6:false | 7:false |
Guess: false, Observe: true
6 7 || 4 1 ||
6 7 || 4 1 ||

Test #3:
0:true | 1:false | 2:true | 3:true | 4:true | 5:true | 6:false | 7:false |
Guess: false, Observe: true
6 7 || 4 1 || 0 2 3 4 5 ||
6 7 || 4 1 || 0 2 3 4 5 ||

Test #4:
0:true | 1:false | 2:false | 3:false | 4:false | 5:false | 6:true | 7:true |
Guess: true, Observe: false
6 7 || 4 1 || 0 2 3 4 5 ||
4 1 || 0 2 3 4 5 ||

Test #7:
0:true | 1:false | 2:true | 3:true | 4:true | 5:false | 6:false | 7:true |
Guess: false, Observe: true
4 1 || 0 2 3 4 5 || 0 2 3 4 7 ||
4 1 || 0 2 3 4 5 || 0 2 3 4 7 ||

Test #9:
0:false | 1:false | 2:true | 3:true | 4:false | 5:false | 6:true | 7:true |
Guess: false, Observe: true
4 1 || 0 2 3 4 5 || 0 2 3 4 7 || 2 3 6 7 ||
4 1 || 0 2 3 4 5 || 0 2 3 4 7 || 2 3 6 7 ||

Test #14:
0:true | 1:false | 2:true | 3:false | 4:false | 5:true | 6:true | 7:false |
Guess: false, Observe: true
4 1 || 3 4 || 3 4 7 || 3 7 || 5 2 0 || 2 0 || 6 2 ||
4 1 || 3 4 || 3 7 || 2 0 || 6 2 ||

Test #15:
0:true | 1:true | 2:false | 3:true | 4:false | 5:true | 6:true | 7:true |
Guess: true, Observe: false
4 1 || 3 4 || 3 7 || 2 0 || 6 2 ||
4 1 || 3 4 || 2 0 || 6 2 ||

Test #39:
0:false | 1:false | 2:true | 3:false | 4:true | 5:true | 6:false | 7:false |
Guess: false, Observe: true
4 1 || 3 4 || 2 0 || 6 2 || 2 4 5 ||
4 1 || 3 4 || 2 0 || 6 2 || 2 4 5 ||

Test #59:
0:false | 1:false | 2:true | 3:false | 4:true | 5:false | 6:false | 7:true |
Guess: false, Observe: true
4 1 || 3 4 || 2 0 || 6 2 || 2 4 5 || 2 4 7 ||
4 1 || 3 4 || 2 0 || 6 2 || 2 4 5 || 2 4 7 ||
```

```
Test #69:
0:false | 1:true | 2:true | 3:false | 4:false | 5:true | 6:false | 7:true |
Guess: false, Observe: true
4 || 3 4 || 0 || 6 || 4 || 4 || 1 || 2 || 2 || 5 2 || 7 2 ||
```
0 || 6 || 4 || 1 || 2 ||

```
Test #71:
0:false | 1:true | 2:false | 3:false | 4:false | 5:true | 6:false | 7:false |
Guess: true, Observe: false
0 || 6 || 4 || 1 4 7 6 || 2 ||
```
0 || 6 || 4 || 2 ||

```
Test #76:
0:true | 1:false | 2:false | 3:true | 4:false | 5:true | 6:true | 7:false |
Guess: true, Observe: false
0 3 4 7 || 6 3 7 || 4 || 2 ||
```
6 3 7 || 4 || 2 ||

```
Test #96:
0:false | 1:false | 2:false | 3:false | 4:true | 5:true | 6:false | 7:true |
Guess: true, Observe: false
6 3 7 || 4 || 2 ||
```
6 3 7 || 2 ||

```
Test #104:
0:true | 1:true | 2:false | 3:true | 4:true | 5:true | 6:true | 7:false |
Guess: false, Observe: true
6 3 7 || 2 || 0 1 3 4 5 6 ||
```
6 3 7 || 2 || 0 1 3 4 5 6 ||

```
Test #110:
0:true | 1:true | 2:false | 3:false | 4:true | 5:true | 6:true | 7:false |
Guess: false, Observe: true
6 3 7 || 2 || 3 || 6 5 4 1 0 ||
```
2 || 3 || 6 5 4 1 0 ||

```
Test #113:
0:false | 1:false | 2:false | 3:true | 4:false | 5:false | 6:false | 7:false |
Guess: true, Observe: false
2 || 3 || 6 5 4 1 0 ||
```
2 || 6 5 4 1 0 ||

```
Test #122:
0:true | 1:true | 2:false | 3:true | 4:true | 5:true | 6:false | 7:false |
Guess: false, Observe: true
2 || 6 5 4 1 0 || 0 1 3 4 5 ||
```
2 || 6 5 4 1 0 || 0 1 3 4 5 ||

```
Test #133:
0:true | 1:true | 2:false | 3:true | 4:true | 5:false | 6:false | 7:false |
Guess: false, Observe: true
2 || 6 5 || 5 || 0 1 4 || 4 3 1 0 ||
```
2 || 5 || 0 1 4 ||

```
Test #138:
0:false | 1:true | 2:false | 3:true | 4:false | 5:true | 6:true | 7:true |
Guess: true, Observe: false
2 || 5 || 0 1 4 ||
```
2 || 0 1 4 ||

```
Test #146:
0:false | 1:false | 2:false | 3:true | 4:true | 5:true | 6:false | 7:false |
Guess: false, Observe: true
2 || 0 1 4 || 3 4 5 ||
```
2 || 0 1 4 || 3 4 5 ||

```
Test #155:
0:false | 1:true | 2:false | 3:false | 4:true | 5:true | 6:false | 7:true |
Guess: false, Observe: true
2 || 0 1 4 || 3 4 5 || 1 4 5 7 ||
2 || 0 1 4 || 3 4 5 || 1 4 5 7 ||

Test #175:
0:false | 1:true | 2:false | 3:false | 4:true | 5:true | 6:false | 7:false |
Guess: false, Observe: true
2 || 0 || 3 || 7 || 4 1 || 5 4 || 5 4 1 ||
2 || 7 || 4 1 || 5 4 ||

Test #181:
0:true | 1:false | 2:false | 3:false | 4:true | 5:true | 6:false | 7:true |
Guess: true, Observe: false
2 || 7 || 4 1 || 5 4 3 ||
2 || 4 1 || 5 4 3 ||
```

And after less than two-hundred tests, we have the Theory we were looking for. Obviously, the more complicated the expression of the truth (based on our Perceptions), the more tests it will take to close in on it, especially when the values of our test Perceptions are completely random.

Conclusion

The Truth Searcher was partly inspired by the writings of Karl Popper, whose thoughts on science and knowledge I have only begun to study. I was introduced to Karl Popper by David Deutsch through his fantastic book, *The Beginning of Infinity: Explanations that Transform the World* (2011), which I highly recommend to all.

Thanks for reading my thoughts and explorations. If you have any questions or comments, I can be reached at seanthebest@gmail.com, though I can't promise quick replies, because I am often rather ~~lazy~~ busy.