Introduction

C. R. Ramakrishnan

CSE 505
**A Puzzle**

**Knights and Liars:** Knights always tell the truth; Liars always lie.

<table>
<thead>
<tr>
<th>Zoe:</th>
<th>“Mel is a liar”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mel:</td>
<td>“Neither I nor Zoe are liars”</td>
</tr>
</tbody>
</table>

Who’s lying?
Zoe: “Mel is a liar”

Mel: “Neither I nor Zoe are liars”
Why!? Logic!!

Zoe: “Mel is a liar”

Mel: “Neither I nor Zoe are liars”

\[ z \iff \neg m \] (1)
Why!? Logic!!

Zoe: “Mel is a liar”

Mel: “Neither I nor Zoe are liars”

\[ z \iff \neg m \] (1)

\[ m \iff m \land z \] (2)
Why!? Logic!!

Zoe: “Mel is a liar”

Mel: “Neither I nor Zoe are liars”

\[
\begin{align*}
z & \iff \neg m \\
m & \iff m \land z
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>(1)</th>
<th>(2)</th>
<th>(1) \land (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>z</td>
<td>m</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>m</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>z</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>m</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

Computing with Logic
Introduction
Why!? Logic!!

**Zoe:** “Mel is a liar”

**Mel:** “Neither I nor Zoe are liars”

\[ z \iff \neg m \quad (1) \]

\[ m \iff m \land z \quad (2) \]

<table>
<thead>
<tr>
<th>z</th>
<th>m</th>
<th>(1)</th>
<th>(2)</th>
<th>(1) \land (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
Zoe: “Mel is a liar”

Mel: “Neither I nor Zoe are liars”

\[ z \iff \neg m \]  
\[ m \iff m \land z \]  

<table>
<thead>
<tr>
<th>z</th>
<th>m</th>
<th>(1)</th>
<th>(2)</th>
<th>(1) \land (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td></td>
<td>F</td>
</tr>
</tbody>
</table>
Zoe: “Mel is a liar”

Mel: “Neither I nor Zoe are liars”

\[
\begin{align*}
z & \Leftrightarrow \neg m \\
m & \Leftrightarrow m \land z
\end{align*}
\]

<table>
<thead>
<tr>
<th>(z)</th>
<th>(m)</th>
<th>(1)</th>
<th>(2)</th>
<th>(1) &amp; (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td></td>
</tr>
</tbody>
</table>
Zoe: “Mel is a liar”

Mel: “Neither I nor Zoe are liars”

\[ z \Leftrightarrow \neg m \quad (1) \]

\[ m \Leftrightarrow m \land z \quad (2) \]

<table>
<thead>
<tr>
<th>z</th>
<th>m</th>
<th>(1)</th>
<th>(2)</th>
<th>(1) \land (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
**Zoe:** “Mel is a liar”

**Mel:** “Neither I nor Zoe are liars”

\[ z \iff \neg m \] 
\[ m \iff m \land z \] 

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>z</strong></td>
<td><strong>m</strong></td>
<td>(1)</td>
<td>(2)</td>
<td>(1) \land (2)</td>
</tr>
<tr>
<td><strong>T</strong></td>
<td><strong>T</strong></td>
<td><strong>F</strong></td>
<td><strong>T</strong></td>
<td><strong>F</strong></td>
</tr>
<tr>
<td><strong>T</strong></td>
<td><strong>F</strong></td>
<td><strong>T</strong></td>
<td><strong>T</strong></td>
<td><strong>T</strong></td>
</tr>
</tbody>
</table>
**Zoe:** “Mel is a liar”

**Mel:** “Neither I nor Zoe are liars”

\[ z \iff \neg m \quad (1) \]

\[ m \iff m \land z \quad (2) \]

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>z</strong></td>
<td><strong>m</strong></td>
<td>(1)</td>
<td>(2)</td>
<td>(1) $$\land$$ (2)</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Zoe: “Mel is a liar”
Mel: “Neither I nor Zoe are liars”

\[ z \iff \neg m \quad (1) \]

\[ m \iff m \land z \quad (2) \]

<table>
<thead>
<tr>
<th>z</th>
<th>m</th>
<th>(1)</th>
<th>(2)</th>
<th>(1) \land (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>
Why Programming?

It was a dark and stormy night. Three men, Alex, Bob & Carl, taking refuge from the rains, came to a hotel, one after another. One was a “knight”: he always spoke the truth; another was a “liar”: he always lied; and the third was a “knave”: he alternated lying with speaking the truth.

When they arrived at the hotel, the manager was not at the desk. When she came to the front, she said she had only one room available, and will give it to the person who arrived first; the other two have to make do with the chairs in the lobby.

...
The conversation went like this:

**Alex:** I came first.
... 

*The conversation went like this:*

**Alex:** *I came first.*

**Bob:** *No, he did not! I came first.*
The conversation went like this:

Alex: I came first.
Bob: No, he did not! I came first.
Carl: No, he did not! I came first.
The conversation went like this:

**Alex:** I came first.

**Bob:** No, he did not! I came first.

**Carl:** No, he did not! I came first.

**Alex:** No, he did not! I came first.
The conversation went like this:

Alex: I came first.
Bob: No, he did not! I came first.
Carl: No, he did not! I came first.
Alex: No, he did not! I came first.
Carl: Well, Bob did not come second.
... 

The conversation went like this:

Alex: I came first.
Bob: No, he did not! I came first.
Carl: No, he did not! I came first.
Alex: No, he did not! I came first.
Carl: Well, Bob did not come second.
Bob: That’s true!
The conversation went like this:

**Alex:** I came first.

**Bob:** No, he did not! I came first.

**Carl:** No, he did not! I came first.

**Alex:** No, he did not! I came first.

**Carl:** Well, Bob did not come second.

**Bob:** That’s true!

Who came first? Who came second? Who came third?
AND OVER THERE WE HAVE THE LABYRINTH GUARDS. ONE ALWAYS LIES, ONE ALWAYS TELLS THE TRUTH, AND ONE STABS PEOPLE WHO ASK TRICKY QUESTIONS.
Who owns the Zebra?

There are five houses in a row amongst which are distributed five colors, five drinks, five cigarettes, five pets and five nationalities subject to the following constraints:

1. The Englishman lives in the red house.
Who owns the Zebra?

There are five houses in a row amongst which are distributed five colors, five drinks, five cigarettes, five pets and five nationalities subject to the following constraints:

1. The Englishman lives in the red house.
2. The Spaniard owns a dog.

...
Who owns the Zebra?

There are five houses in a row amongst which are distributed five colors, five drinks, five cigarettes, five pets and five nationalities subject to the following constraints:

1. The Englishman lives in the red house.
2. The Spaniard owns a dog.
3. The Norwegian lives in the first house.
Who owns the Zebra?

There are five houses in a row amongst which are distributed five colors, five drinks, five cigarettes, five pets and five nationalities subject to the following constraints:

1. The Englishman lives in the red house.
2. The Spaniard owns a dog.
3. The Norwegian lives in the first house.
4. Kools are smoked in the yellow house.

...
Who owns the Zebra?

There are five houses in a row amongst which are distributed five colors, five drinks, five cigarettes, five pets and five nationalities subject to the following constraints:

1. The Englishman lives in the red house.
2. The Spaniard owns a dog.
3. The Norwegian lives in the first house.
4. Kools are smoked in the yellow house.

Who owns the zebra?
What addresses may be accessed as \*p?

points_to(P, Q) :-
    assign_stmt(plain(P), addr(Q)).  \% p = &q

points_to(P, X) :-
    assign_stmt(plain(P), plain(Q)), \% p = q
    points_to(Q, X).

points_to(P, X) :-
    assign_stmt(plain(P), star(Q)), \% p = *q
    points_to(Q, R),
    points_to(R, X).

points_to(X, Y) :-
    assign_stmt(star(P), plain(Q)), \% *p = q
    points_to(P, X),
    points_to(Q, Y).