Iron man using JARVIS
How does Jarvis work

• Natural Language
• Classical Logic: and, if then, or, not (Boolean)
  • This is efficient

• Modalities: notions of necessity, possibility, belief, etc..
  • If I flip my bottle it is possible it will stand upright

• Jarvis must compute the truth value of the modal formula
• But current modal reasoners are not efficient
AUTOMATED THEOREM PROVING FOR MODAL LOGIC

Ethan Nguyen (u6672582)
Supervisor: Rajeev Gore
Course: COMP3770
• Incorporate SAT solvers into Modal Logic theorem provers
AUTOMATED THEOREM PROVING

Computer assisted reasoning

Is a Boolean formula valid?
• Validity: all possible truth assignments of the atoms make the formula true

<table>
<thead>
<tr>
<th>p</th>
<th>¬p</th>
<th>p v ¬p</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>
SATISFIABILITY

• Satisfiability: there exists a truth assignment of the atoms which make the formula true

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>p ∨ q</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
FROM SATISFIABILITY TO VALIDITY

- Is $A$ valid?
- $A$ is valid if $\neg A$ is not satisfiable:

<table>
<thead>
<tr>
<th>p</th>
<th>$\neg(p \lor \neg p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

- SAT problem: is formula $A$ satisfiable?
• Tableau is also called truth trees:
MODERN SAT SOLVERS ARE EFFICIENT

- Over the last 20 years, very efficient SAT solvers have been created
- MiniSAT can handle hundreds of propositional variables
AIM

- Incorporate SAT solvers into Modal Logic theorem provers
- We expect this would result in a speed increase
CLASSICAL MODAL LOGIC

- An extension of propositional logic which is concerned with reasoning about necessities and possibilities
BOX, DIAMOND

- Two unary operators: Box(□) and Diamond (<>)
- □p: it is necessary that p
- <>p: it is possible that p
• Boolean logic is evaluated via truth tables
• How do we extend truth tables to handle modalities?
• Modal formula are evaluated in a Kripke model
  • Worlds (W)
  • Accessibility Relation (R)
  • Valuations (V)
KRIPKE MODEL EXAMPLE

Worlds (W); Accessibility Relation (R); Valuations (V)

Root World

Accessibility relation

Successor World
Valuation: the bottle is standing

Accessibility relation

Successor World
Valuation: the bottle is not standing
Let $p$ be the bottle is standing.

- $\square p$ is true in $W$, if and only if $p$ is true in all worlds accessible from $W$
- $\Diamond p$ is true in $W$, if and only if $p$ is true in some world accessible from $W$
HIGH LEVEL ALGORITHM

- SAT solver in each world
- Learn from other worlds

W0
V: the bottle is standing

R
W1
V: the bottle is not standing

R
W2
V: the bottle is not standing
• Negate A
• Manipulate a modal formula \( \neg A \) into:
  • \( \neg A = MC_1 \land MC_2 \land MC_3 \)
  • \( MC_1: \) Boolean formula (no modalities)
  • \( MC_2: \) Boolean implies box (\( a \rightarrow 
\square b \))
  • \( MC_3: \) Boolean implies diamond (\( c \rightarrow <>d \))
• And then…
\[ \neg A = \neg MC_1 \land \neg MC_2 \land \neg MC_3 \]

- **MC1**: Boolean formula (no modalities)
- **MC2**: Boolean implies box \( (a \Rightarrow []b) \)
- **MC3**: Boolean implies diamond \( (c \Rightarrow <>d) \)

Check MC1 for satisfiability. Do this by feeding MC1 into a sat solver. If it is not satisfiable we are done.

No model

Root world
\[ \neg A = \neg MC1 \land \neg MC2 \land \neg MC3 \]

- **MC1**: Boolean formula (no modalities)
- **MC2**: Boolean implies box \((a \implies [b])\)
- **MC3**: Boolean implies diamond \((c \implies \Diamond d)\)

But if there is a model (i.e. MC1 is satisfiable)

- **MC2**: \(a \implies [b]\)
- **MC3**: 
  - \(q \implies \Diamond \neg b\)
  - \(r \implies \Diamond d\)

![Root world diagram](image)
\[ \sim A = \Lambda MC_1 \land \Lambda MC_2 \land \Lambda MC_3 \]

- **MC1**: Boolean formula (no modalities)
- **MC2**: Boolean implies box \((a \Rightarrow []b)\)
- **MC3**: Boolean implies diamond \((c \Rightarrow <>d)\)

- Learned \(b \Rightarrow \sim q\)
- Termination: root world is unsatisfiable, or you make all worlds satisfiable (happy)
### RESULTS

- It is still buggy
- Some results:

<table>
<thead>
<tr>
<th>subclass</th>
<th>BDDTab</th>
<th>BUC</th>
<th>SUC</th>
<th>NUC</th>
<th>Norm</th>
<th>Reorder</th>
<th>R to L</th>
<th>FaCT++</th>
<th>InKreSAT</th>
<th>&quot;*&quot;SAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>branch_n</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>19</td>
<td>17</td>
<td>10</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>branch_p</td>
<td>21</td>
<td>1</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>10</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>lin_n</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>14</td>
</tr>
<tr>
<td>path_n</td>
<td>21</td>
<td>2</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>path_p</td>
<td>21</td>
<td>3</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>10</td>
<td>21</td>
</tr>
<tr>
<td>ph_n</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>13</td>
<td>21</td>
<td>12</td>
</tr>
<tr>
<td>ph_p</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>7</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>poly_p</td>
<td>21</td>
<td>9</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>13</td>
</tr>
<tr>
<td>t4p_p</td>
<td>21</td>
<td>0</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>
CONCLUSION

- Determining validity for modal logic
- SAT solver instead of tableau
- Modal Logic has a lot of applications

- Current implementation is naïve, uses list
- Repetition
QUESTIONS