Sipser: Chapter 7.1 – 7.3
This game sucks and I can prove it.
Sudoku
$L_D = \{ \langle s \rangle \mid s \text{ is an instance of Sudoku } \}$
and $s$ has a solution
\( L_D = \{ \langle s \rangle \mid s \text{ is an instance of Sudoku } \} \)
and \( s \) has a solution

\( L_V = \{ \langle s, c \rangle \mid s \text{ is an instance of Sudoku } \} \)
and \( c \) is a solution to \( s \)
<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>3</td>
<td>4</td>
<td></td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>2</td>
<td></td>
<td>1</td>
<td>9</td>
<td>5</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>8</td>
<td></td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>9</td>
<td></td>
<td>7</td>
<td>6</td>
<td>1</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>6</td>
<td></td>
<td>8</td>
<td>5</td>
<td>3</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3</td>
<td></td>
<td>9</td>
<td>2</td>
<td>4</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1</td>
<td></td>
<td>5</td>
<td>3</td>
<td>7</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>7</td>
<td></td>
<td>4</td>
<td>1</td>
<td>9</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
<td>2</td>
<td>8</td>
<td>6</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

$\mathbf{L_v}$
\[
\begin{align*}
L_D &= \{ \langle s \rangle \mid s \text{ is an instance of Sudoku} \} \\
x 100 \quad \text{and } s \text{ has a solution}
\end{align*}
\]

\[
\begin{align*}
L_V &= \{ \langle s, c \rangle \mid s \text{ is an instance of Sudoku} \} \\
x 100 \quad \text{and } c \text{ is a solution to } s
\end{align*}
\]
$L_D = 2^n$

$LV = n^2$
\[ L_D = k^n \]

\[ L_V = n^k \]
\[ L_D = \{ \langle s \rangle \mid s \text{ is an instance of Sudoku } \} \]
and \( s \) has a solution

\[ L_V = \{ \langle s, c \rangle \mid s \text{ is an instance of Sudoku } \} \]
and \( c \) is a solution to \( s \)
\[ L_D = \begin{cases} 
\text{Generate the next candidate solution, } c, \text{ for } s \\
\text{Run } L_V(\langle s, c \rangle) \\
\text{If } L_V \text{ accepts, } \text{.ACCEPT} \\
\text{If there are no more candidates, REJECT} 
\end{cases} \]
$S = \begin{cases} 
\text{Generate the next candidate solution, } c, \text{ for } s \\
\text{If there are no more candidates, REJECT} \\
\text{Run } L_{V}(\langle s, c \rangle) \\
\text{If } L_{V} \text{ accepts, return } c 
\end{cases}$
The CKY Algorithm for CFGs
\[(A \quad B) \times (E \quad F) = (AE+BG \quad AF+BH) \]
\[(C \quad D) \times (G \quad H) = (CE+DG \quad CF+DH)\]
\[ L_D = \begin{cases} 
\text{Generate the next candidate solution, } c, \text{ for } s \\
\text{If there are no more candidates, REJECT} \\
\text{Run } L_V(\langle s, c \rangle) \\
\text{If } L_V \text{ accepts, ACCEPT} 
\end{cases} \]
Finite State Transducer
\[ L_D = \{ \text{Generate candidate solutions} \} \]

\[ \text{Run } L_V(\langle s, c \rangle) \]
\[ \text{run } L_V(\langle s, c \rangle) \]
\[ \text{run } L_V(\langle s, c \rangle) \]
\[ \text{run } L_V(\langle s, c \rangle) \]
\[ \text{run } L_V(\langle s, c \rangle) \]
\[ \text{run } L_V(\langle s, c \rangle) \]
\[ \text{run } L_V(\langle s, c \rangle) \]
\[ \text{run } L_V(\langle s, c \rangle) \]
\[ \text{run } L_V(\langle s, c \rangle) \]
\[ \text{If accept, ACCEPT} \]

Run Time?
The class NP:

Verifiable in polynomial time

* with a deterministic TM
The class NP:

Verifiable in polynomial time

\[ \downarrow \]

Decidable in non-deterministic polynomial time
The class \textbf{NP}:

Verifiable in polynomial time

\downarrow

Decidable in non-deterministic polynomial time
The class NP:

Verifiable in polynomial time

\[
\downarrow
\]

Decidable in non-deterministic polynomial time
The Marriage Problem  Sudoku
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>3</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 4 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 2 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\[ L_D = \{ \langle m \rangle \mid m \text{ is an instance of The Marriage Problem and } m \text{ has a solution} \} \]
$$
L_D = \{ \langle m \rangle \mid m \text{ is an instance of The Marriage Problem and } m \text{ has a solution} \}
$$

$$
L_V = \{ \langle m, c \rangle \mid m \text{ is an instance of The Marriage Problem and } c \text{ is a solution to } m \}
$$
\[ L_D = \{ \langle m \rangle \mid m \text{ is an instance of The Marriage Problem and } m \text{ has a solution} \} \]

\[ L_V = \{ \langle m, c \rangle \mid m \text{ is an instance of The Marriage Problem and } c \text{ is a solution to } m \} \]
\[ L_D = \begin{cases} 
\text{Generate the next candidate solution, } c, \text{ for } m \\
\text{If there are no more candidates, REJECT} \\
\text{Run } L_V(\langle m, c \rangle) \\
\text{If } L_V \text{ accepts, ACCEPT} 
\end{cases} \]
\[ S = \begin{cases} 
\text{Generate the next candidate solution, } c, \text{ for } m \\
\text{If there are no more candidates, REJECT} \\
\text{Run } L_V(\langle m, c \rangle) \\
\text{If } L_V \text{ accepts, return } c 
\end{cases} \]
\[
L_D = \begin{array}{c}
\text{Generate candidate solutions} \\
\end{array}
\]

Run \(L_v(\langle m, c \rangle)\)

Run \(L_v(\langle m, c \rangle)\)

Run \(L_v(\langle m, c \rangle)\)

Run \(L_v(\langle m, c \rangle)\)

Run \(L_v(\langle m, c \rangle)\)

Run \(L_v(\langle m, c \rangle)\)

Run \(L_v(\langle m, c \rangle)\)

Run \(L_v(\langle m, c \rangle)\)

Run \(L_v(\langle m, c \rangle)\)

Run \(L_v(\langle m, c \rangle)\)

If accept, ACCEPT

\[
\]
\( L_D = \{ \langle m \rangle \mid m \text{ is an instance of The Marriage Problem and } m \text{ has a solution} \} \)

\( L_V = \{ \langle m, c \rangle \mid m \text{ is an instance of The Marriage Problem and } c \text{ is a solution to } m \} \)
Hall’s Marriage Theorem:

Suppose $G$ is a bipartite graph with partition $(A, B)$. There is a matching that covers $A$ if and only if, for every subset $X \subseteq A$, $N(X) \geq |X|$ where $N(x)$ is the number of neighbors of $x$. 
Hall’s Marriage Theorem:

Suppose $G$ is a bipartite graph with partition $(A, B)$. There is a matching that covers $A$ if and only if, for every subset $X \subseteq A$, $N(X) \geq |X|$ where $N(x)$ is the number of neighbors of $X$. 
Hall’s Marriage Theorem:

Suppose $G$ is a bipartite graph with partition $(A, B)$. There is a matching that covers $A$ if and only if, for every subset $X \subseteq A$, $N(X) \geq |X|$ where $N(x)$ is the number of neighbors of $X$. 
An “M-augmenting” path
An “M-augmenting” path
An “M-augmenting” path
An “M-augmenting” path
An “M-augmenting” path

{5 4}  {3 6}  
{6 4 2}  {6}  
{5 2 1}  {1 4 5 6}  
{4 3}  {1 3}  
{5}  {4 5}  
{5 2 1}  {2 5 6}  

An "M-augmenting" path
Hall’s Marriage Theorem:

**Lemma:** If $M$ is a maximum matching in graph $G$, there can be no $M$-augmenting path

**Algo:** $M$-augmenting paths can be found in polynomial time via breadth-first search
\[ L_D = \{ \langle m \rangle \mid m \text{ is an instance of The Marriage Problem and } m \text{ has a solution} \} \]

Polynomial Time

\[ L_V = \{ \langle m, c \rangle \mid m \text{ is an instance of The Marriage Problem and } c \text{ is a solution to } m \} \]

Polynomial Time
The Marriage Problem:

1.) Verifiable in polynomial time
2.) Decidable in polynomial time

* with a deterministic TM
The class P:

1.) Verifiable in polynomial time
2.) Decidable in polynomial time

* with a deterministic TM
The class $P$: 

1.) Verifiable in polynomial time 

2.) Decidable in polynomial time
The class P:

1.) Verifiable in polynomial time
2.) Decidable in polynomial time

In NP?
The class P:

1.) Verifiable in polynomial time
2.) Decidable in polynomial time

In NP?
The Marriage Problem: NP  P

Sudoku: NP
P = NP ?
The Marriage Problem  Sudoku  The Travelling Salesman
$$L_D = \{ \langle t \rangle \mid t \text{ is an instance of The Travelling Salesman} \}$$

Yes!

$$L_V = \{ \langle t, c \rangle \mid t \text{ is an instance of The Travelling Salesman and } c \text{ is a solution to } t \}$$
$$L_D = \{ \langle t \rangle \mid t \text{ is an instance of The Travelling Salesman and } t \text{ has a solution } < x \}$$

$$L_V = \{ \langle t, c \rangle \mid t \text{ is an instance of The Travelling Salesman and } c \text{ is a solution to } t \}$$
\[ S = \begin{cases} 
\text{Generate the next candidate solution, } c, \text{ for } t & \\
\text{If there are no more candidates, REJECT} & \\
\text{Run } L_V(\langle t, c \rangle) & \\
\text{If } L_V \text{ accepts, return } c & 
\end{cases} \]
$L_V = \begin{cases} 
\text{Generate the next candidate solution, } c', \text{ for } t \\
\quad \rightarrow \text{ If there are no more candidates, ACCEPT} \\
\text{Calculate } \langle t, c' \rangle \\
\quad \rightarrow \text{ If } \langle t, c' \rangle \text{ is lower than } \langle t, c \rangle, \text{ REJECT} 
\end{cases}$
\[ L_V = \text{Generate candidate solutions} \]

If lower, REJECT
\[ L_D = \{ \langle t \rangle \mid t \text{ is an instance of The Travelling Salesman and } t \text{ has a solution} \} \]

\[ L_V = \{ \langle t, c \rangle \mid t \text{ is an instance of The Travelling Salesman and } c \text{ is a solution to } t \} \]
\( S = \{ \langle t \rangle \mid t \text{ is an instance of The Travelling Salesman and } t \text{ has a solution} \} \)

Exponential Time

\( L_V = \{ \langle t, c \rangle \mid t \text{ is an instance of The Travelling Salesman and } c \text{ is a solution to } t \} \)

Exponential Time
The class EXPTIME:

Decision problems that can be solved in exponential time.

TS: No known polynomial time verification procedure
EXPTIME

NP

P
The Marriage $\leq$ Sudoku $\leq$ The Travelling Salesman
Find the prime factors of 39203

The Marriage ≤ Sudoku ≤ The Travelling Salesman
Find the prime factors of 39203

The Marriage ≤ Sudoku ≤ The Travelling Salesman