Representing Problems as SAT
Order n Latin Squares (= Order n quasigroups)

- n x n matrix
- elements from $[n] = \{1, \ldots, n\}$ // or any other set of size $n$
- no row or column has two cells with the same elements

Ex:

\[
\begin{array}{ccc}
  a & d & b \\
  d & e & a \\
  c & b & d \\
  a & c & a \\
\end{array}
\]

\[
\begin{array}{ccc}
  a & d & b \\
  c & a & d \\
  c & b & d \\
  d & a & e \\
\end{array}
\]

Atoms: $C_{ijk}$ for $i,j,k \in [n]$ // $C_{ijk}$ means $C_{i,j} = k$

Clauses:
1. $\bigwedge_{i,j \in [n]} C_{ijk} \bigvee C_{ijk}$ // Every cell gets a value
2. $\bigwedge_{i,j,k,l \in [n]} \left( C_{ijk} \bigvee C_{ijl} \right)$ // no row has the same element in two columns
3. $\bigwedge_{i,j,k,l \in [n]} \left( C_{ijk} \bigvee C_{ijl} \right)$ // no column has the same element in two rows

Let $\Gamma_n = \text{conjunction of clause in } 1, 2, 3$. 
- Satisfying assignments for $\Gamma_n$ are (il) with order n Latin Squares
**Q:** Why no \( \bigwedge_{i,j,k,k',k'' \in [n]} (C_{ijk} \lor C_{ijk'}) \) \( \text{No cell gets two elements} \)

**Quasigroup Completion**  
\( (= \text{Latin Square Completion}) \)

**Instance:** \( nxn \) matrix \( I \) with elements from \([n] \cup \{ \emptyset \} \)

**Question:** Is there an \( nxn \) Latin Square \( C \) with elements from \([n] \) st. \( C_{ij} = I_{ij} \)  
for every \( i,j \in [n] \) st. \( I_{ij} \neq \emptyset \)  
//i.e. is there a completion of \( I \) that is a Latin Square.

**Example:** \( I = \begin{bmatrix} 1 & 2 & 3 & \emptyset \\ 2 & 3 & \emptyset & 1 \\ 3 & \emptyset & 1 & 2 \\ \emptyset & 2 & 3 \end{bmatrix} \)

Let \( \Gamma_1 = \Gamma_4 \cup \{ (C_{111}), (C_{122}), (C_{123}), (C_{212}), (C_{223}), \\ (C_{313}), (C_{324}), (C_{331}), (C_{423}), (C_{443}) \} \). 

Satisfying assignments for \( \Gamma_1 \) are solutions for \( I \).

Let \( \Gamma_2 = \Gamma_n \cup \{ (C_{ijkl}) \mid I_{ijkl} \neq \emptyset \} \).

Then satisfying assignments for \( \Gamma_2 \) are solutions for \( I \).
Boolean Pythagorean Triples Problem

Q: Is there a bipartition of $N = \{1, 2, \ldots, 3\}$ s.t. neither block contains a Pythagorean Triple? 

\[ (a, b, c) \text{ s.t. } a^2 + b^2 = c^2 \]

Thm: There is a 2-colouring of $[7824]$ with no monochromatic PT.

- Every 2-colouring of $[7825]$ has a monochromatic PT.

(Open problem solved using a complex SAT-solving scheme in 2016)

\[ \text{Atoms: } \{ x_i \mid i \in [n] \} \]

\[ F_n = \bigwedge_{a,b,c \in [n]} ((x_a \lor x_b \lor x_c) \land (\overline{x_a} \lor \overline{x_b} \lor \overline{x_c})) \]

\[ \text{s.t. } a^2 + b^2 = c^2 \quad \text{a,b,c are not in the same block.} \]

\[ F_n \text{ “says”: There is a 2 colouring of } [n] \text{ with no monochrome PT.} \]

- $F_{7824}$ easily shown satisfiable.

- $F_{7825}$ has 6,494 variables, 18,944 clauses.

- Not large by industrial standards but hard.

- Biggest problem: how to generate a checkable proof of unsatisfiability.
Applications to Digital Circuits

A circuit \( C \) with \( k \) inputs and \( l \) outputs computes a function \( f_C : \{0,1\}^k \rightarrow \{0,1\}^l \).

We can represent the computation performed by the circuit with a formula that describes input/output relation in each gate:

- atoms correspond to (values on) wires
- clauses restrict those values (in a manner similar to the Tseitin transformation to CNF.)

\[ \varphi_C = (z \leftrightarrow (x_1 \land x_2)) \land (y \leftrightarrow (z \lor x_3)) \]

Claim: \( \alpha \vdash \varphi_C \iff f_C(\alpha | x) = \alpha(y) \)
Circuit Equivalence

Instance: Two circuits C, D

Question: \( f_C = f_D \)?

Take: \( \varphi_C \land \varphi_D \) to define the computation of the combined circuit

\[
\varphi_{C \# D} = (\varphi_C \land \varphi_D) \land \neg ((y_1 \leftrightarrow u_1) \land (y_2 \leftrightarrow u_2) \land \ldots \land (y_e \leftrightarrow u_e))
\]

- \( C = D \iff \varphi_{C \# D} \) is unsatisfiable.

- A satisfying assign. for \( \varphi_{C \# D} \) shows \( C \neq D \).

  More: it gives an input for which the circuits give different outputs.
Automated Test Pattern Generation (A.T.P. G)

Correct Circuit

We want an input that distinguishes these two circuits.

\[ \Phi_{\text{flaw}} = (\neg z \land (Y' \leftrightarrow (z' \lor x_3))) \]

\[ \Phi_{\text{test}} = \Phi_c \land \Phi_{\text{flaw}} \land \neg (Y \leftrightarrow Y') \]

A satisfying assignment in \( \Phi_{\text{test}} \) gives an input on which the two circuits have different outputs, i.e., a test pattern for the flaw.
Automated (Propositional STRIPS) Planning

A problem instance consists of:

- **Set** $F$ of "facts" (propositional atoms)
  
  A "state of the world" is a t.a. for $F$.

- **Initial State** $I$: a t.a. for $F$.

- A set $A$ of actions:
  
  - each $a \in A$ defined by two sets of literals over $F$:
    
    - $\text{pre}(a)$ – preconditions of $a$
    
    - $\text{eff}(a)$ – effects of $a$

- **Goal** $G$: a set of literals.
A plan is a sequence of actions $P = (a_1, a_2, \ldots, a_L)$ s.t. there is a sequence of states:

$$S_P = \langle S_0, S_1, \ldots, S_L \rangle$$

with:

- $S_0 = I$
- for $1 \leq i \leq L$, $a_i$ is executable in $S_{i-1}$

$$S_{i-1} \models \text{pre}(a_i)$$
- for each $1 \leq i \leq L$, $S_i$ is the result of executing action $a_i$ in state $S_{i-1}$, i.e.,

$$S_i(f) = \begin{cases} 
\text{true} & \text{if } f \in \text{eff}(a_i) \\
\text{false} & \text{if } f \not\in \text{eff}(a_i) \\
S_{i-1}(f) & \text{o.w.}
\end{cases}$$
- $S_L \models G (S_L$ is a goal state).
• Given a planning instance \( \Pi \) and maximum plan length \( L \),
  existence of a plan of length \( I \leq L \) is NP-complete.

• We write formulas \( \Phi^I_T \) s.t.
  \( \Phi^I_T \) is satisfiable iff there is a plan for \( \Pi \)
  with at most \( T \) time steps.

• We use two sets of atoms:
  - State atoms: \( f_t \), for each fact \( f \in F \), \( t \in \{0, \ldots, T\} \)
  - Action atoms: \( a_t \), for each \( a \in A \), \( t \in \{1, \ldots, T\} \)

• \( \Phi^I \) is the conjunction of the sets:
  1. State at time \( 0 \) is \( I \), the initial:
     \( f_0 \) for each fact \( f \in F \), \( (f_0) \) if \( f \in I \), \( (\neg f_0) \) o.w.
2. **Goal States:** State at time $T$ satisfies the goal condition.  
For $f \in F$, $(f_t)$ if $f \in \textit{S}$, $(\neg f_t)$ if $\neg f \in \textit{S}$.

3. **Action Preconditions:**  
For each action $a$, each time $t \in \{1, \ldots, T\}$, have clauses equivalent to:

\[ a_t \rightarrow \bigwedge \{ L_{t-1} \mid \text{pre}(a) \} \]

4. **Action Effects:**  
For each action $a$, each time $t \in \{1, \ldots, T\}$

\[ a_t \rightarrow \bigwedge \{ L_t \mid \text{eff}(a) \} \]
5. **Explanatory Frame Axioms:**

For each fact \( f \in F \), and each time \( t \in \{1, \ldots, T\} \), have clauses equivalent to:

\[
(f_{t-1} \land \neg f_t) \rightarrow \bigvee_{a} \neg a_t \quad \text{At} \quad \{a \mid f \in \text{eff}(a)\}
\]

and

\[
(\neg f_{t-1} \land f_t) \rightarrow \bigvee_{a} a_t \quad \text{At} \quad \{a \mid f \in \text{eff}(a)\}
\]

6. **Serializability of Actions**

For each pair \( a, b \) of distinct actions with \( \text{pre}(a) \cup \text{eff}(b) \) unsatisfiable, and each time \( t \),

\[
(\neg a_t \lor \neg b_t)
\]
End