## Summary/TL;DR

**Motivation**
- Different planning representations for classical planning (e.g., FDR, STRIPS)
- Computational complexity independent of chosen representation
- However: accidental complexity (of the chosen representation and model) can impact planner performance

**Contribution**
- Merge-and-shrink (M&S) framework for task reformulation:
  - Task representation based on factored transition system (FTS)
  - Various transformations for satisficing and optimal planning
  - Plan reconstruction methods
  - Theoretical result: M&S reformulations dominate previous FDR-based reformulation methods
  - Adaptation of delete-relaxation and M&S heuristics to FTS representation

## FTS Task Reformulations with M&S

**Task Reformulation**
Partial function \( \rho \) on task \( \Pi \) s.t. \( \rho(\Pi) \) is solvable iff \( \Pi \) is solvable and there exists a plan reconstruction function \( \overline{\rho} \) that maps each solution \( \pi \) of \( \rho(\Pi) \) to a solution \( \overline{\rho}(\pi) \) of \( \Pi \).

\[
\begin{align*}
\text{Task } \Pi & \quad \xrightarrow{\text{reformulation } (\rho)} \quad \text{Task } \rho(\Pi) \\
\text{Plan } \pi & \quad \xrightarrow{\text{plan reconstruction } (\overline{\rho})} \quad \text{Plan } \overline{\rho}(\pi)
\end{align*}
\]

**M&S Transformations on FTS Task \( \Pi^T \)**

**Exact** transformations preserve the set of solutions (optimal planning):
- **Label reduction**: combine labels with the same transitions in all but one factor
- **Bisimulation Shrinking**: combine states in one factor if they are bisimilar (their outgoing transitions lead to equivalent states)
- **Merging**: replace two factors of \( \Pi^T \) by their product
- **Pruning**: throw away states not relevant for solutions

Transformations that preserve the existence of a solution (satisficing planning):
- **Weak bisimulation**: two states are weakly bisimilar if they have equivalent outgoing paths \( \ldots \rightarrow \tau \ldots \rightarrow \tau \ldots \rightarrow \rightarrow \) leading to equivalent states, where \( \tau \) are **internal labels** that can always be applied locally without side effects.

**Example task:** one truck, four locations, limited amount of fuel, turn on engine only with full tank \( \rightarrow \) 3 FDR variables/atomic Ts.

## Plan Reconstruction

- Given a sequence of planning tasks and reformulations and a plan of the final reformulated task, reconstruct plan of the original task.
- Treat exact transformations (merging, label reduction and bisimulation shrinking) as single transformation: only require label and state mapping
- Weak bisimulation: re-introduce \( \tau \)-label transitions whenever necessary

**Example**
- Example plan for task of Figure c):
  \( (A, 2, \text{off}) \rightarrow (A, 2, \text{on}) \rightarrow (B, 1, \text{off}) \rightarrow (D, 0, \text{on}) \)
- Execution on task of Figure b): Dtr fails in rd; insert a \( \tau \)-transition with \( \text{On} \) resulting in the plan:
  \( (A, 2, \text{off}) \rightarrow (A, 2, \text{on}) \rightarrow (B, 1, \text{on}) \rightarrow (D, 0, \text{on}) \)
- Reconstruct original plan by inverting label mapping: \( (A, 2, \text{off}) \rightarrow (A, 2, \text{on}) \rightarrow (B, 1, \text{on}) \rightarrow (D, 0, \text{on}) \)

## Theoretical Comparison to FDR Reformulations

**Dominance**
FTRS reformulation \( X \) dominates FDR reformulation \( Y \) if, given an FDR task \( \Pi^T \) and an applicable reformulation \( \rho^Y \) \( \in Y \), there exists a reformulation \( \rho^X \) \( \in X \) such that it is applicable in the corresponding FTS task atomic(\( \Pi^T \)) and \( \rho^X(\text{atomic}(\Pi^T)) = \text{atomic}(\rho^Y(\Pi^T)) \).

**Generalize actions:** substitute two FDR actions by a single one if they are equal except for a pre-condition on a binary variable.

**Safe variable abstraction:** remove any root variable in the causal graph whose free domain transition graph is strongly connected.

**Theorem:** Removing transition systems with core states after applying weak bisimulation shrinking strictly dominates safe variable abstraction.

**Merge values:** reduce the domain of an FDR variable by merging several values whenever they can be switched via actions without any side effects.

**Theorem:** Weak bisimulation shrinking strictly dominates merge values.

## Planning on the FTS Representation

- **Applicable actions:** multiple transitions with same label for single state \( \rightarrow \) for each abstract state, store set of outgoing labels
- **Successor generation:** enumerate all successors for single state and label
- **Delete relaxation heuristics:** factors can be in multiple states simultaneously

## Experiments

<table>
<thead>
<tr>
<th>Experiment</th>
<th>FDR</th>
<th>FDR</th>
<th>FDR</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \rho )</td>
<td>( \rho )</td>
<td>( \rho )</td>
<td>( \rho )</td>
</tr>
<tr>
<td>FDR</td>
<td>12</td>
<td>13</td>
<td>37</td>
<td>36</td>
</tr>
<tr>
<td>a</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>a-ls</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>m-ls</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>FDR</td>
<td>18</td>
<td>15</td>
<td>27</td>
<td>22</td>
</tr>
<tr>
<td>a</td>
<td>6</td>
<td>13</td>
<td>28</td>
<td>22</td>
</tr>
<tr>
<td>a-ls</td>
<td>15</td>
<td>18</td>
<td>31</td>
<td>24</td>
</tr>
<tr>
<td>m-ls</td>
<td>13</td>
<td>15</td>
<td>7</td>
<td>21</td>
</tr>
<tr>
<td>FDR</td>
<td>17</td>
<td>15</td>
<td>24</td>
<td>23</td>
</tr>
<tr>
<td>a</td>
<td>8</td>
<td>11</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>a-ls</td>
<td>13</td>
<td>18</td>
<td>26</td>
<td>17</td>
</tr>
<tr>
<td>m-ls</td>
<td>9</td>
<td>6</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>FDR</td>
<td>16</td>
<td>15</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>a</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>a-ls</td>
<td>15</td>
<td>16</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>m-ls</td>
<td>9</td>
<td>7</td>
<td>3</td>
<td>16</td>
</tr>
</tbody>
</table>

**b) After shrinking and removing irrelevant On.**

**Example:**
- Atomic task: truck position \( \Theta^T \), fuel \( \Theta^F \), and status \( \Theta^S \).
- After labeling reduction
- After shrinking and removing irrelevant On.
- After merging and pruning unreachable states.