A Normal Form for Classical Planning Tasks

Florian Pommerening\textsuperscript{1}    Malte Helmert

University of Basel, Switzerland

June 9, 2015

\textsuperscript{1}Supported by the GI-FB KI Travel Grant
Pop Quiz

Pop quiz on classical planning

- **Question 1 (Incomplete operators)**
  An operator $o$ unconditionally sets variable $A$ to 1.
  (a) What transition does $o$ induce in the DTG for $A$?
  (b) Does $o$ produce the fact $A \leftrightarrow 1$?

- **Question 2 (Partial goal states)**
  The only goal is to set $A$ to 1
  (a) What is the goal value of $B$?
  (b) What is the regression of the goal with operator $o$?
Pop Quiz

Pop quiz on classical planning

**Question 1** (Incomplete operators)
An operator $o$ unconditionally sets variable $A$ to 1.

(a) What transition does $o$ induce in the DTG for $A$?
   $v \rightarrow 1$ for all values $v$ of $A$

(b) Does $o$ produce the fact $A \rightarrow 1$?
   Not necessarily

**Question 2** (Partial goal states)
The only goal is to set $A$ to 1

(a) What is the goal value of $B$?
   Any value is fine

(b) What is the regression of the goal with operator $o$?
   The set of all states
Pop Quiz

Pop quiz on classical planning

- **Question 1** (Incomplete operators)
  An operator $o$ unconditionally sets variable $A$ to 1.
  (a) What transition does $o$ induce in the DTG for $A$?
      $v \rightarrow 1$ for all values $v$ of $A$
  (b) Does $o$ produce the fact $A \rightarrow 1$?
      Not necessarily

- **Question 2** (Partial goal states)
  The only goal is to set $A$ to 1
  (a) What is the goal value of $B$?
      Any value is fine
  (b) What is the regression of the goal with operator $o$?
      The set of all states

Not impossible to answer but would be easier with **complete operators and a complete goal state**
Simplification

- Restrict attention to simpler form
- Show that any task can be transformed into this form
- Transformed task should be equivalent to original
  - Meaning of “equivalent” depends on application
  - Transformation maintains important properties:
    Shortest path, landmarks, etc.
Transition Normal Form
## Transition Normal Form

**Definition (Transition Normal Form)**

A planning task is in transition normal form if

- \( \text{vars}(\text{pre}(o)) = \text{vars}(\text{eff}(o)) \) for all operators
- Every variable has a goal value
Folklore Transformation

Multiply out effects

Example

\[ o : \langle \emptyset, \{ A \mapsto 1, B \mapsto 0 \} \rangle \]

- \[ o_1 : \langle \{ A \mapsto 0, B \mapsto 0 \}, \{ A \mapsto 1, B \mapsto 0 \} \rangle \]
- \[ o_2 : \langle \{ A \mapsto 0, B \mapsto 1 \}, \{ A \mapsto 1, B \mapsto 0 \} \rangle \]
- \[ o_3 : \langle \{ A \mapsto 1, B \mapsto 0 \}, \{ A \mapsto 1, B \mapsto 0 \} \rangle \]
- \[ o_4 : \langle \{ A \mapsto 1, B \mapsto 1 \}, \{ A \mapsto 1, B \mapsto 0 \} \rangle \]

Problem: Exponential increase in task size
Transition Normalization

Alternative transformation with only linear size increase

- Allow to forget the value of any variable at any time
- New value $u$ represents “forgotten” value
- Require the value $u$ when there are no other restrictions
Transition Normalization Definition

Definition ($\text{TNF(II)}$)

- Add fresh value $u$ to each variable domain
- Forgetting operator for each fact
  - Allows transition from $V \mapsto v$ to $V \mapsto u$
  - No cost
- Precondition $V \mapsto v$ without effect on $V$
  - Add effect $V \mapsto v$
- Effect $V \mapsto v$ without precondition on $V$
  - Add precondition $V \mapsto u$
- Unspecified goal value for $V$
  - Add goal value $V \mapsto u$
Transition Normalization Example

Example

\[ o : \langle \{ B \mapsto 0 \}, \{ A \mapsto 1 \} \rangle \]

\[ \text{goal} = \{ A \mapsto 1 \} \]

- Forgetting operators (cost = 0)
  - \( \text{forget}_{A \mapsto 0} : \langle \{ A \mapsto 0 \}, \{ A \mapsto u \} \rangle \)
  - \( \text{forget}_{A \mapsto 1} : \langle \{ A \mapsto 1 \}, \{ A \mapsto u \} \rangle \)
  - \( \text{forget}_{B \mapsto 0} : \langle \{ B \mapsto 0 \}, \{ B \mapsto u \} \rangle \)
  - \( \text{forget}_{B \mapsto 1} : \langle \{ B \mapsto 1 \}, \{ B \mapsto u \} \rangle \)

- Modify precondition and effect
  - \( o' = \langle \{ A \mapsto u, B \mapsto 0 \}, \{ A \mapsto 1, B \mapsto 0 \} \rangle \)

- Modify goal
  - \( \text{goal}' = \{ A \mapsto 1, B \mapsto u \} \)
Theorem ($\Pi \rightarrow TNF(\Pi)$)

Every plan for $\Pi$ can be efficiently converted to a plan with the same cost for $TNF(\Pi)$.

**Proof idea:** insert forgetting operators where necessary

Theorem ($TNF(\Pi) \rightarrow \Pi$)

Every plan for $TNF(\Pi)$ can be efficiently converted to a plan with the same cost for $\Pi$.

**Proof idea:** remove all forgetting operators
Rest of this talk

- Properties maintained by this transformation
- When and when not to use the transformation
Effect of Transition Normalization on Heuristics
Delete Relaxation

Delete relaxation heuristic $h^+$
- Ignores delete effects of operators

Theorem
\[ \Pi \text{ and } TNF(\Pi) \text{ have the same } h^+ \text{ values on all states from } \Pi. \]
Critical Paths

Critical path heuristics $h^m$

- Considers only fact sets up to size $m$
- $h^m$-value of a set of facts $F$: cost to reach all facts in $F$
- Special case: $h^1 = h^{\text{max}}$

**Theorem**

$\Pi$ and $\text{TNF}(\Pi)$ have the same $h^m$ values for fact sets from $\Pi$.

**Corollary**

$\Pi$ and $\text{TNF}(\Pi)$ have the same $h^m$ values on all states from $\Pi$. 
Landmarks

(Disjunctive action) landmark

- Set of operators
- At least one operator occurs in each plan

Theorem

Landmarks without forgetting operators are the same in $\Pi$ and $TNF(\Pi)$.

Theorem

$\Pi$ and $TNF(\Pi)$ have the same $h^{LM}$-cut values on all states from $\Pi$ (if they break ties in the same way).
Abstractions

**Domain Transition Graphs (DTGs)**

- Model operator effects on single variables
- Used in merge-and-shrink, LAMA, etc.
- Are **not the same** in $\Pi$ and $\text{TNF}(\Pi)$

**Theorem**

Every operator in $\text{TNF}(\Pi)$ only introduces **one transition**.

**Corollary**

Worst-case **number of transitions** is **linear** instead of quadratic.
Potential Heuristics

Potential heuristics

- Recently introduced class of heuristics
- Heuristic value is weighted sum over facts in state
- Weights constrained so heuristic is admissible and consistent
- Can generate best potential heuristic

Constraints in $TNF(\Pi)$

\[
\sum_{f \in \text{goal}} P_f = 0
\]

\[
\sum_{f \in \text{pre}(o)} P_f - \sum_{f \in \text{eff}(o)} P_f \leq \text{cost}(o) \quad \text{for all operators } o
\]

- Formulation for general tasks much more complicated
Effect of Transition Normalization on other Planning Techniques
Zobrist Hashing

- **Zobrist hashing for states**
  - Associate random bit string with each fact
  - \( \text{hash}(s) = \text{XOR over bit strings for each fact in } s \)

- Change for **successor state** after applying operator
  - XOR with bit strings for all deleted facts
  - XOR with bit strings for all added facts

- In **TNF(II)** deleted and added facts are known in advance
  - Effect of an operator can be precomputed
  - Only one XOR necessary

**Similar application:** perfect hash functions for PDB heuristics
Applying operators in regression is involved
- Special cases for partial states
- Special cases for unspecified preconditions

Regression in $\text{TNF}(\Pi)$
- Switch \textit{preconditions} and \textit{effects} of each operator
- Switch \textit{initial state} with \textit{goal state}
- Same application rules as in progression
- Always work on complete states
Conclusion
Using TNF in Practice

“I want to implement a new bi-directional search algorithm. Should I work on the transition normalization?”
Using TNF in Practice

“I want to implement a new bi-directional search algorithm. Should I work on the transition normalization?”

- Not for the implementation!
  - Size of reachable search space can increase exponentially
- Intended use mostly as theoretical tool
  - Design and description of planning techniques
  - Theoretical analysis
- But also lots of practical applications
  - Techniques that are polynomial in the task description size:
    e.g., mutex discovery, relevance analysis, landmark computation, (most) heuristic computations
Transition normalization

- **Linear increase** in task size useful in practice
- **Simplifies concepts** in many areas
- Helps in **design** and **analysis** of planning techniques
- Makes it **more obvious what is going on** e.g., DTG, potential heuristics