A Normal Form for Classical Planning Tasks

Florian Pommerening[1] Malte Helmert

University of Basel, Switzerland

June 9, 2015

[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 $A \mapsto 1$?

- 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?
  $v \to 1$ for all values v of A
  (b) Does o produce the fact $A \mapsto 1$?
  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 $A \mapsto 1$?
  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

Task Transformation

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

- $\mathit{vars}(pre(o)) = \mathit{vars}(\mathit{eff}(o))$ 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 (*TNF*(Π))

- Add fresh value **u** to each variable domain
- Forgetting operator for each fact
  - Allows transition from $V \mapsto v$ to $V \mapsto \mathbf{u}$
  - 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 $V \mapsto \mathbf{u}$
- Unspecified goal value for V
  - Add goal value $V \mapsto \mathbf{u}$

Transition Normalization Example

Example

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

$$\mathit{goal} = \{A \mapsto 1\}$$

- Forgetting operators (cost = 0)
  - $\mathit{forget}_{A \mapsto 0} : \langle \{A \mapsto 0\}, \{A \mapsto \mathbf{u}\} \rangle$
  - $\mathit{forget}_{A \mapsto 1} : \langle \{A \mapsto 1\}, \{A \mapsto \mathbf{u}\} \rangle$
  - $\mathit{forget}_{B \mapsto 0} : \langle \{B \mapsto 0\}, \{B \mapsto \mathbf{u}\} \rangle$
  - $\mathit{forget}_{B \mapsto 1} : \langle \{B \mapsto 1\}, \{B \mapsto \mathbf{u}\} \rangle$
- Modify precondition and effect
  - $o' = \langle \{A \mapsto \mathbf{u}, B \mapsto 0\}, \{A \mapsto 1, B \mapsto 0\} \rangle$
- Modify goal
  - $\mathit{goal}' = \{A \mapsto 1, B \mapsto \mathbf{u}\}$

Correctness

Theorem ($\Pi \to \mathit{TNF}(\Pi)$)

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

Proof idea: insert forgetting operators where necessary

Theorem ($\mathit{TNF}(\Pi) \to \Pi$)

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

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

Π and *TNF*(Π) have the same h^+ values on all states from Π.

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: $h^1 = h^{\max}$

Theorem

Π and *TNF*(Π) have the same h^m values for fact sets from Π.

Corollary

Π and *TNF*(Π) have the same h^m values on all states from Π.

Landmarks

(Disjunctive action) landmark

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

Theorem

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

Theorem

Π and $\mathit{TNF}(\Pi)$ have the same $h^{\text{LM-cut}}$ values on all states from Π (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 Π and *TNF*(Π)

Theorem

Every operator in *TNF*(Π) 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 $\mathit{TNF}(\Pi)$

$$\sum_{f \in \text{goal}} P_f = 0$$

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

- 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
  - $\mathit{hash}(s)$ = 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 $\mathit{TNF}(\Pi)$ 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

Regression

- Applying operators in regression is involved
  - Special cases for partial states
  - Special cases for unspecified preconditions
- Regression in $TNF(\Pi)$
  - Switch preconditions and effects of each operator
  - Switch initial state with 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

Conclusion

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