Planning and Optimization

A5. Equivalent Operators and Effect Normal Form

Malte Helmert and Gabriele Röger

Universität Basel

October 6, 2016

Planning and Optimization

October 6, 2016 — A5. Equivalent Operators and Effect Normal Form

A5.1 Motivation

Motivation

Similarly to normal forms in propositional logic (DNF, CNF, NNF), we can define normal forms for effects, operators and planning tasks.

This is useful because algorithms (and proofs) then only need to deal with effects, operators and tasks in normal form.

A5.2 Equivalence Transformations

Equivalence of Operators and Effects: Definition

Definition (Equivalent Operators)

Two operators o and o' over state variables V are equivalent, written $o \equiv o'$, if $cost(o) = cost(o')$ and for all states s, s' over V, o induces the transition $s \xrightarrow{o} s'$ iff o' induces the transition $s \xrightarrow{o'} s'$.

Definition (Equivalent Effects)

Two effects e and e' over state variables V are equivalent, written $e \equiv e'$, if the operators $\langle \top, e, 0 \rangle$ and $\langle \top, e', 0 \rangle$ are equivalent.

Equivalence of Operators and Effects: Theorem

Theorem

Let o and o' be operators with $pre(o) \equiv pre(o')$, $eff(o) \equiv eff(o')$ and $cost(o) = cost(o')$. Then $o \equiv o'$.

Note: The converse is not true. (Why not?)

Equivalence Transformations for Effects

$$
\begin{aligned}
e_1 \wedge e_2 &\equiv e_2 \wedge e_1 && (1)\\
(e_1 \wedge \dots \wedge e_n) \wedge (e'_1 \wedge \dots \wedge e'_m) &\equiv e_1 \wedge \dots \wedge e_n \wedge e'_1 \wedge \dots \wedge e'_m && (2)\\
\top \wedge e &\equiv e && (3)\\
\chi \rhd e &\equiv \chi' \rhd e \qquad \text{if } \chi \equiv \chi' && (4)\\
\top \rhd e &\equiv e && (5)\\
\bot \rhd e &\equiv \top && (6)\\
\chi_1 \rhd (\chi_2 \rhd e) &\equiv (\chi_1 \wedge \chi_2) \rhd e && (7)\\
\chi \rhd (e_1 \wedge \dots \wedge e_n) &\equiv (\chi \rhd e_1) \wedge \dots \wedge (\chi \rhd e_n) && (8)\\
(\chi_1 \rhd e) \wedge (\chi_2 \rhd e) &\equiv (\chi_1 \vee \chi_2) \rhd e && (9)
\end{aligned}
$$

A5.3 Effect Normal Form

Effect Normal Form: Motivation

- CNF and DNF limit the nesting of connectives in propositional logic.
- For example, a CNF formula is
  - a conjunction of 0 or more subformulas,
  - each of which is a disjunction of 0 or more subformulas,
  - each of which is a literal.
- Similarly, we can define a normal form that limits the nesting of effects.
- This is useful because we then do not have to consider arbitrarily structured effects, e.g., when representing them in a planning algorithm.

Effect Normal Form

Definition (Effect Normal Form)

An effect e is in normal form if:

1. It is
   - a conjunctive effect
   - whose conjuncts are conditional effects
   - whose subeffects are atomic effects, and
2. no atomic effect occurs in e multiple times, and
3. there exists no state s for which $[e]_s$ contains two complementary literals (v and $\neg v$).

An operator o is in effect normal form if $eff(o)$ is in normal form.

Note: non-conjunctive effects can be considered as conjunctive effects with 1 conjunct

Effect Normal Form: Example

Example

$$c \wedge (a \rhd (\neg b \wedge (c \rhd (b \wedge \neg d \wedge \neg a)))) \wedge (\neg b \rhd \neg a)$$

transformed to normal form is

$$\begin{aligned}&(\top \rhd c) \wedge \\ &((a \wedge \neg c) \rhd \neg b) \wedge \\ &((a \wedge c) \rhd b) \wedge \\ &((a \wedge c) \rhd \neg d) \wedge \\ &((\neg b \vee (a \wedge c)) \rhd \neg a)\end{aligned}$$

Note: for simplicity, we will often write $(\top \rhd \ell)$ as ℓ, i.e., omit trivial effect conditions. We will still consider such effects to be in normal form.

Effect Conditions

For effects in normal form, it is easy to determine under which conditions an operator o makes a given literal ℓ true in the resulting state $s[\![o]\!]$.

Definition (Effect Condition)

Let e be an effect over variables V in normal form, and let ℓ be a literal over V.

The effect condition of e for ℓ, written $effcond_\ell(e)$, is defined as

$$effcond_\ell(e) = \begin{cases} \chi & \text{if } e \text{ contains the subeffect } (\chi \rhd \ell) \\ \bot & \text{otherwise} \end{cases}$$

For e in normal form and literals ℓ, we have that $s[\![o]\!] \models \ell$ iff $s \models effcond_\ell(e)$ or ($s \models \ell$ and $s \not\models effcond_{\bar{\ell}}(e)$).
(Here, $\bar{\ell}$ stands for the complementary literal of ℓ.)

Testing if Effects are in Normal Form

- In general, testing whether an effect is in normal form is a coNP-complete problem. (Why?)
- However, we do not usually need such a test. Instead, we can produce normal form in polynomial time.

Producing Effects in Normal Form (1)

Theorem

For every effect, an equivalent effect in normal form can be computed in polynomial time.

Proof sketch.

Use the following algorithm:

1. While conjunctive effects occur within conditional effects, move the conditional effects inside with equivalence (8).
2. Replace atomic effects by conditional effects with equivalence (5).
3. Flatten nested conjunctive effects with equivalence (2).
4. Flatten nested conditional effects with equivalence (7).

This ensures condition 1. of normal form. ...

Producing Effects in Normal Form (2)

Proof sketch (continued).

5. Combine conditional effects with the same subeffect with equivalence (9).

This ensures condition 2. of normal form.

6. For every state variable v where the effect includes the subeffects $(\varphi \rhd v)$ and $(\psi \rhd \neg v)$, replace the latter with $((\psi \wedge \neg\varphi) \rhd \neg v)$.

This ensures condition 3. of normal form.

7. Optionally, use equivalence (4) to simplify effect conditions.

□

A5.4 Summary

Summary

- Effect equivalences can be used to simplify operator effects.
- For effects in normal form, the only permitted nesting is atomic effects within conditional effects within conjunctive effects, and all atomic effects must be distinct.
- For effects in normal form, it is easy to determine the condition under which a given literal is made true by applying the effect in a given state.
- Every effect can be transformed into an equivalent effect in normal form in polynomial time.