Planning and Optimization

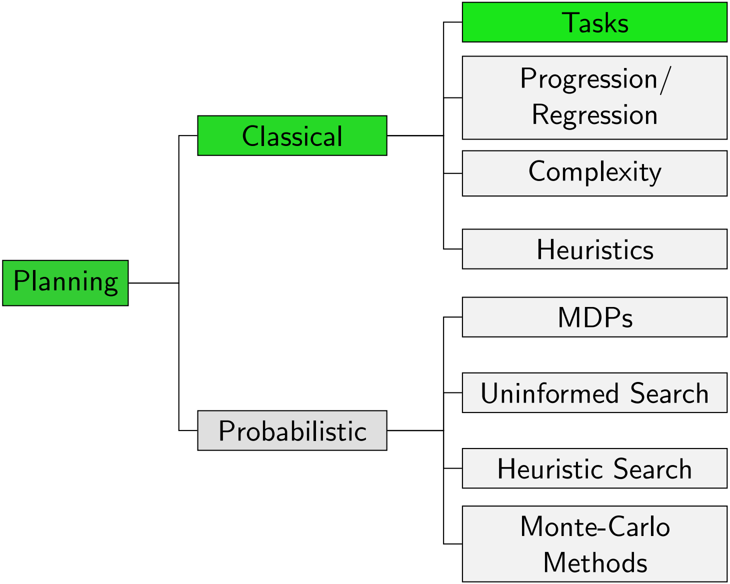
A5. Equivalent Operators and Normal Forms for Effects

Gabriele Röger and Thomas Keller

Universität Basel

October 3, 2018

Content of this Course



Reminder & Motivation

Syntax of Effects

Definition (Effect)

Effects over state variables V are inductively defined as follows:

- If $v \in V$ is a state variable, then v and $\neg v$ are effects (atomic effect).
- If $e_1, \dots, e_n$ are effects, then $(e_1 \wedge \dots \wedge e_n)$ is an effect (conjunctive effect).
  The special case with $n = 0$ is the empty effect $\top$.
- If χ is a logical formula and e is an effect, then $(\chi \triangleright e)$ is an effect (conditional effect).

Arbitrary nesting of conjunctive and conditional effects

Semantics of Effects

- *effcond*(ℓ, e): condition that must be true in the current state for the effect e to lead to the atomic effect ℓ
- add-after-delete semantics: if operator o with effect e is applicable in state s, the successor state $s[\![o]\!]$ is defined as:

$$s[\![o]\!](v) = \begin{cases} \mathbf{T} & \text{if } s \models \mathit{effcond}(v, e) \\ \mathbf{F} & \text{if } s \models \mathit{effcond}(\neg v, e) \land \neg \mathit{effcond}(v, e) \\ s(v) & \text{if } s \not\models \mathit{effcond}(v, e) \lor \mathit{effcond}(\neg v, e) \end{cases}$$

Semantics of Effects

- *effcond*(ℓ, e): condition that must be true in the current state for the effect e to lead to the atomic effect ℓ
- add-after-delete semantics: if operator o with effect e is applicable in state s, the successor state $s[\![o]\!]$ is defined as:

$$s[\![o]\!](v) = \begin{cases} \mathbf{T} & \text{if } s \models \mathit{effcond}(v, e) \\ \mathbf{F} & \text{if } s \models \mathit{effcond}(\neg v, e) \land \neg \mathit{effcond}(v, e) \\ s(v) & \text{if } s \not\models \mathit{effcond}(v, e) \lor \mathit{effcond}(\neg v, e) \end{cases}$$

- New notation
  - If we do not want to consider a precondition, we also write $s[\![e]\!]$ for $s[\![\langle \top, e \rangle]\!]$.
  - For a sequence $\pi = \langle o_1, \dots, o_n \rangle$ of operators that are consecutively applicable in s, we write $s[\![\pi]\!]$ for $s[\![o_1]\!][\![o_2]\!] \dots [\![o_n]\!]$.

Motivation

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

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

Equivalence Transformations

Equivalence of Operators and Effects: Definition

Definition (Equivalent Effects)

Two effects e and e' over state variables V are equivalent, written $e \equiv e'$, if $s[\![e]\!] = s[\![e']\!]$ for all states s.

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 $s \xrightarrow{o'} s'$.

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

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

Conflict-Free Effects

Conflict-Freeness: Motivation

- The add-after-delete semantics makes effects like $(a \rhd c) \wedge (b \rhd \neg c)$ somewhat unintuitive to interpret.

⇝ What happens in states where $a \wedge b$ is true?

- It would be nicer if $\mathit{effcond}(\neg v, e)$ were always the condition under which e makes v false (but because of add-after-delete, it is not).

⇝ introduce a normal form where the "complicated case" of add-after-delete semantics never arises

Conflict-Free Effects

Definition (Conflict-Free)

An effect e is called conflict-free if $effcond(v, e) \wedge effcond(\neg v, e)$ is unsatisfiable for all state variables v.

An operator o is called conflict-free if $eff(o)$ is conflict-free.

Testing if Effects are Conflict-Free

- In general, testing whether an effect is conflict-free is a coNP-complete problem. (Why?)
- However, we do not usually need such a test. Instead, we can produce an equivalent conflict-free effect in polynomial time.
- Algorithm: given effect e, replace each atomic effect of the form $\neg v$ by $(\neg \mathit{effcond}(v, e) \rhd \neg v)$.
  The resulting effect e' is conflict-free and $e \equiv e'$. (Why?)

Flat Effects

Flat Effects: 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.

Flat Effect

Definition (Flat Effect)

An effect e is flat if it is:

- a conjunctive effect
- whose conjuncts are conditional effects
- whose subeffects are atomic effects, and
- no atomic effect occurs in e multiple times.

An operator o is flat if $\mathit{eff}(o)$ is flat.

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

Flat Effect: Example

Example

Consider the effect

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

An equivalent flat (and conflict-free) effect is

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

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.

Producing Flat Effects

Theorem

For every effect, an equivalent flat effect and an equivalent flat, conflict-free effect can be computed in polynomial time.

Proof Sketch.

Every effect e over variables V is equivalent to
$\bigwedge_{v \in V}(\mathit{effcond}(v, e) \rhd v) \land \bigwedge_{v \in V}(\mathit{effcond}(\neg v, e) \rhd \neg v)$,
which is a flat effect.

For conflict-free and flat, use $\mathit{effcond}(\neg v, e) \land \neg \mathit{effcond}(v, e)$ instead of $\mathit{effcond}(\neg v, e)$.

In both cases, conjuncts of the form $(\chi \rhd \ell)$ where $\chi \equiv \bot$ can be omitted to simplify the effect.

Summary

Summary

- Effect equivalences can be used to simplify operator effects.
- In conflict-free effects, the "complicated case" in the add-after-delete semantics of operators does not arise.
- For flat effects, the only permitted nesting is atomic effects within conditional effects within conjunctive effects, and all atomic effects must be distinct.
- For flat, conflict-free effects, 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 flat and conflict-free effect in polynomial time.