Probabilistic Programming

Lecture #8: Weakest Preconditions

Joost-Pieter Katoen

RWTH Lecture Series on Probabilistic Programming 2022-23

Overview

1. Motivation
2. The guarded command language
3. Weakest preconditions
4. Weakest liberal preconditions

Code-level reasoning

Proving properties of programs: not by executing them,
but by reasoning at the syntax level of programs.

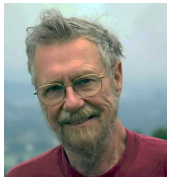
Compositionality: determine the correctness of composed program P
by reasoning about its parts in isolation and
then obtain P's correctness result by combining those parts' analyses.

Today: Dijkstra's weakest preconditions. No probabilities (yet).

Overview

1. Motivation
2. The guarded command language
3. Weakest preconditions
4. Weakest liberal preconditions

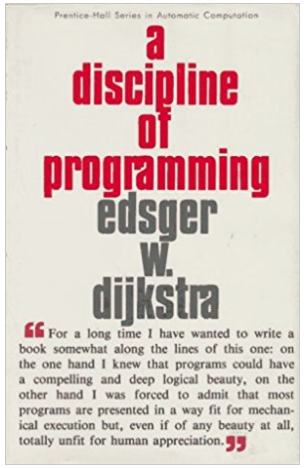
Dijkstra's guarded command language

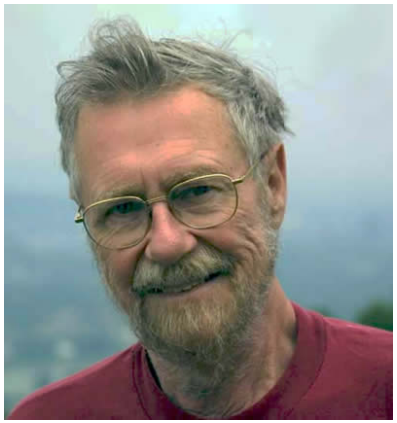


- `skip` — empty statement
- `x := E` — assignment
- `prog1 ; prog2` — sequential composition
- `if (G) prog1 else prog2` — choice
- `while (G) prog` — iteration

For simplicity: we omit non-deterministic choice.

A discipline of programming





Predicate transformers

Let the set of states be:

$$\mathbb{S} \;=\; \{\, s \mid s : \mathit{Vars} \to \mathbb{Q} \,\}$$

Let the set of predicates be:

$$\mathbb{P} \;=\; \left\{\, F \mid F : \underbrace{\mathbb{S}}_{\text{states}} \to \{0,1\} \,\right\}$$

Predicate F is typically a first-order logic formula. It equals $\{\, s \in \mathbb{S} \mid s \models F \,\}$. Thus $\mathbb{P} = 2^{\mathbb{S}}$. Let partial order $\sqsubseteq$ equal $\subseteq$. Ergo:

$(\mathbb{P}, \sqsubseteq)$ is a complete lattice where $F \sqsubseteq G$ if and only if $F \Rightarrow G$

Function $\Phi : \mathbb{P} \to \mathbb{P}$ is called a predicate transformer

Examples

- Predicates are sets of variable valuations.
- Program statements can be viewed as predicate transformers
- One is interested in preconditions that are least restrictive

Overview

1 Motivation

2 The guarded command language

3 Weakest preconditions

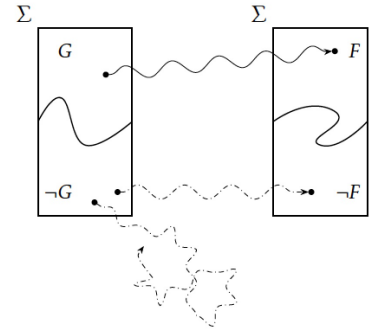
4 Weakest liberal preconditions

Weakest preconditions

For program P, let $wp[\![P]\!] : \mathbb{P} \to \mathbb{P}$ a predicate transformer.

$G = wp[\![P]\!](F)$ is P's weakest precondition w.r.t. postcondition F iff

- If P starts in a state $s \models G$, it terminates in a state $t \models F$.
- Otherwise, P either terminates in a state $t \not\models F$ or diverges



Sir Tony Hoare (1934–)

quicksort, Floyd-Hoare style reasoning, communicating sequential processes, ...

Hoare triples

- If P is started in state where G holds, and it terminates, then in its final state F holds:

  Hoare triple $\{G\}\ P\ \{F\}$ is a valid statement

- For each $G \in \mathbb{P}$ there are many $F \in \mathbb{P}$ such that

  $\{G\}\ P\ \{F\}$ is a valid statement

- For each $F \in \mathbb{P}$ there are many $G \in \mathbb{P}$ such that

  $\{G\}\ P\ \{F\}$ is a valid statement

Hoare triples are relational

Weakest preconditions versus Hoare triples

Weakest preconditions are functional

- For each $F \in \mathbb{P}$ there is a unique $G \in \mathbb{P}$ such that

$$wp[\![P]\!](F) \;=\; G$$

- Weakest preconditions respect Hoare triples:

$$\{\, wp[\![P]\!](F) \,\}\ P\ \{\, F \,\} \qquad \text{is a valid statement}$$

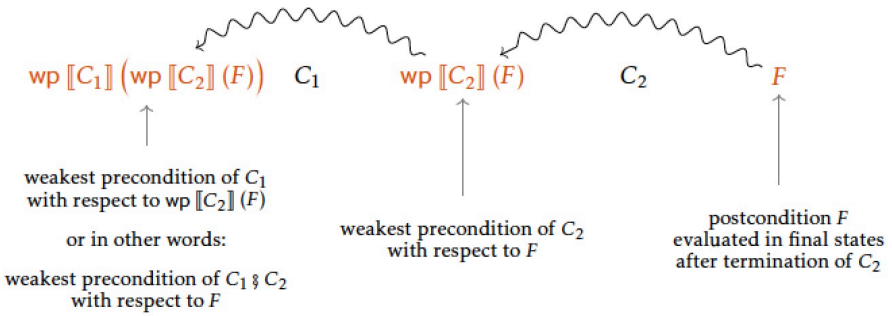
- For terminating[1] P:

$$\{\, G \,\}\ P\ \{\, F \,\} \qquad \text{is a valid statement, then} \qquad \{\, G \,\} \;\Rightarrow\; wp[\![P]\!](F)$$

[1] For diverging P, the statement $\{\text{true}\}\, P\, \{F\}$ is trivially true, but $wp[\![P]\!](F) = \text{false}$.

Example

Backward reasoning



Weakest preconditions reason in a backward manner about programs.

Weakest preconditions for GCL

Syntax program P	Weakest precondition $wp[\![P]\!](F)$
`skip`	F
$x := E$	$F[x := E]$
$P; Q$	$wp[\![P]\!]\,(wp[\![Q]\!](F))$
`if` (φ) P `else` Q	$(\varphi \wedge wp[\![P]\!](F)) \vee (\neg\varphi \wedge wp[\![Q]\!](F))$
`while` (φ) $\{P\}$	$\text{lfp}\, X.\ \underbrace{((\varphi \wedge wp[\![P]\!](X)) \vee (\neg\varphi \wedge F))}_{\text{loop characteristic function } \Phi_F(X)}$

where lfp is the least fixed point wrt. the ordering $\sqsubseteq\ =\ \Rightarrow$ on $\mathbb{P}$.

Examples of straight-line code

Possibly unbounded loops

$$wp[\![\text{while } (\varphi)\,\{\,P\,\},F]\!] \;=\; \text{lfp}\,X.\;\underbrace{((\varphi \wedge wp[\![P]\!](X)) \vee (\neg\varphi \wedge F))}_{\text{loop characteristic function } \Phi_F(X)}$$

- The function $\Phi_F : \mathbb{P} \to \mathbb{P}$ is Scott continuous on $(\mathbb{P}, \sqsubseteq)$.
- Kleene's fixed point theorem yields: $\text{lfp}\ \Phi_F = \sup_{n\in\mathbb{N}} \Phi_F^n(\text{false})$.
- $\Phi_F^n(\text{false})$ denotes the wp of running the loop n times starting from $\varnothing$, the empty set of states.

Loops in action

$wp[\![P]\!](\text{true})$ = weakest precondition under which P terminates

A loopy program example

```
while (x > 0) {
    x--
}
```

What is the weakest pre-condition on x
such that on termination x is non-negative?

Approximating while-loops

Let:

$$\begin{aligned} \text{while}^0(\varphi)\{P\}) &= \text{diverge} \\ \text{while}^{n+1}(\varphi)\{P\}) &= \text{if } (\varphi) \text{ then } P; \text{while}^n(\varphi)\{P\}) \text{ else skip} \end{aligned}$$

where `diverge` equals `while(true) skip`

Let $\Phi_F(X) = ((\varphi \wedge wp[\![P]\!](X)) \vee (\neg\varphi \wedge F))$. Then for all $n \in \mathbb{N}$ it holds:

$$\Phi_F^n(\text{false}) = wp[\![\text{while}^n(\varphi)\{P\}]\!](F)$$

Proof.

By induction on n using the inductive definition of wp. □

Overview

1. Motivation
2. The guarded command language
3. Weakest preconditions
4. Weakest liberal preconditions

Motivation

The precondition $wp[\![P]\!](F)$ is the least restrictive requirement needed to guarantee that P terminates and F holds after P.

Weakest preconditions thus guarantee termination

What if we want to also reason about possible divergence, i.e., non-termination?

This is exactly what weakest **liberal** preconditions do

Weakest liberal preconditions versus Hoare triples

Weakest liberal preconditions are functional

- For each $F \in \mathbb{P}$ there is a unique $G \in \mathbb{P}$ such that

$$wlp[\![P]\!](F) = G$$

- For terminating P:

$\{G\}\ P\ \{F\}$ is a valid statement, then $\{G\} \Rightarrow \underbrace{wp[\![P]\!](F)}_{\text{standard}}$

- For all P:

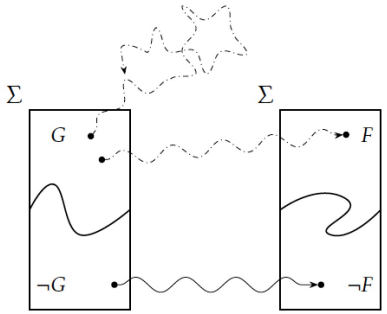
$\{G\}\ P\ \{F\}$ is a valid statement, then $\{G\} \Rightarrow \underbrace{wlp[\![P]\!](F)}_{\text{liberal}}$

Weakest liberal preconditions

For program P, let $wlp[\![P]\!] : \mathbb{P} \to \mathbb{P}$ a predicate transformer.

$G = wp[\![P]\!](F)$ is P's weakest liberal precondition w.r.t. postcondition F:

- If P starts in a state $s \models G$, it
  - either terminates in a state $t \models F$
  - or diverges
- Otherwise, P terminates in a state $t \not\models F$



Weakest liberal preconditions for GCL

Syntax program P	Weakest liberal precondition $wlp[\![P]\!](F)$
`skip`	F
$x := E$	$F[x := E]$
$P; Q$	$wlp[\![P]\!]\,(wlp[\![Q]\!](F))$
`if` (φ) P `else` Q	$(\varphi \wedge wlp[\![P]\!](F)) \vee (\neg\varphi \wedge wlp[\![Q]\!](F))$
`while` (φ) $\{P\}$	$\mathrm{gfp}\, X.\ \underbrace{((\varphi \wedge wlp[\![P]\!](X)) \vee (\neg\varphi \wedge F))}_{\text{loop characteristic function } \Psi_F(X)}$

where gfp is the greatest fixed point wrt. the ordering $\sqsubseteq\ =\ \Rightarrow$ on $\mathbb{P}$.

Possibly unbounded loops

$$wlp[\![\text{while } (\varphi)\,\{\,P\,\}]\!](F) \;=\; \mathrm{gfp}\, X.\ \underbrace{((\varphi \wedge wlp[\![P]\!](X)) \vee (\neg\varphi \wedge F))}_{\text{loop characteristic function } \Psi_F(X)}$$

- The function $\Psi_F : \mathbb{P} \to \mathbb{P}$ is Scott continuous on $(\mathbb{P}, \sqsubseteq)$.
- Kleene's fixed point theorem yields: $\mathrm{gfp}\ \Psi_F = \inf_{n \in \mathbb{N}} \Psi_F^n(\mathrm{true})$.
- $\Psi^n(\mathrm{true})$ denotes the wlp of running the loop exactly n times starting from the entire set $\mathbb{S}$ of states.

Elementary properties of Dijkstra's wp and wlp

- Duality: $wlp[\![P]\!](F) \;=\; wp[\![P]\!](F) \vee \neg wp[\![P]\!](\mathrm{true})$
- Divergence: $wp[\![\mathtt{diverge}]\!](F) = \mathrm{false}$ and $wlp[\![\mathtt{diverge}]\!](F) = \mathrm{true}$
- Strictness: $wp[\![P]\!](\mathrm{false}) = \mathrm{false}$ and $wlp[\![P]\!](\mathit{true}) = \mathrm{true}$
- Distribution $wp[\![P]\!](F \vee G) \;=\; wp[\![P]\!](F) \vee wp[\![P]\!](G)$
- Termination $wp[\![P]\!](\mathrm{true}) =$ weakest precondition s.t. P terminates

$wp[\![P]\!](F) \;\neq\; wlp[\![P]\!](F)$ implies that P may diverge.

Take-home messages

- Weakest preconditions characterize the largest set of states s.t. when starting in any such a state:
  - the program terminates and
  - does so in a state satisfying the post-condition
- They can be defined inductively on the program's syntax
  - using least fixed points to capture while-loops
- And are functions: predicate transformers
- Weakest liberal preconditions do not insist on termination

Next lecture: how to generalise this to probabilistic programs?

Next lecture

Tuesday Nov 15, 16:30