15-414: Bug Catching: Automated Program Verification

Lecture Notes on Diamonds

Frank Pfenning

Carnegie Mellon University

Lecture 9
Thursday, March 4, 2021

1 Introduction

So far in our study of dynamic logic we have focused on $[\alpha]P$, meaning that P is true after every possible run of α. In the world of deterministic programs we call this *partial correctness*: the final state satisfies P, but only if α terminates. We also sometimes talk about a *safety property*: no matter what happens, if we terminate at least P will be true.

The other modality is $\langle\alpha\rangle P$ which means that there is a run of α such that P is true. For deterministic programs (that is, programs that have at most one final state), we call this *total correctness*: α will reach a final state, and it satisfies P. We also sometimes talk about a *liveness property*: something good (that is a final state that satisfied P) will eventually happen.

In this lecture we recall the semantics of $\langle\alpha\rangle P$ more formally and then examine how to break down programs for this particular modal operator by using axioms. This will be straightforward until we encounter α^*, which requires an axiom of *convergence* as a counterpart to the axiom of *induction*.

Learning goals. After this lecture, you should be able to:

- Express liveness properties in dynamic logic;
- Reason with the axiom of convergence;
- Reason with interacting $[-]$ and $\langle-\rangle$ modalities.

2 Box vs. Diamond

Recall that we defined

$$
\begin{array}{ll}
\omega \models [\alpha]Q & \text{iff for every } \nu,\ \omega[\![\alpha]\!]\nu \text{ implies } \nu \models P \\
\omega \models \langle\alpha\rangle Q & \text{iff there exists a } \nu \text{ such that } \omega[\![\alpha]\!]\nu \text{ and } \nu \models P
\end{array}
$$

Both of these are with respect to the same semantics $\omega[\![\alpha]\!]\nu$. In the first case, if ν is reachable then P must be true; in the second case some such ν must be reachable.

Recall the definitions

$$
\begin{array}{lcl}
\mathsf{skip} & \triangleq & ?\mathsf{true} \\
\mathsf{abort} & \triangleq & ?\mathsf{false}
\end{array}
$$

From this definition we can deduce the following properties. You should make sure you understand each line.

$$
\begin{array}{ll}
[\mathsf{skip}]P & \text{iff } P \\
\langle\mathsf{skip}\rangle P & \text{iff } P \\
[\mathsf{abort}]P & \text{always} \\
\langle\mathsf{abort}\rangle P & \text{never} \\
[\alpha^*]\mathsf{true} & \text{always} \\
\langle\alpha^*\rangle\mathsf{true} & \text{always} \\
[\alpha^*]\mathsf{false} & \text{never} \\
\langle\alpha^*\rangle\mathsf{false} & \text{never}
\end{array}
$$

3 Axioms for Diamonds

We would like to break down the programs in $\langle\alpha\rangle Q$ in order to generate a verification condition in pure arithmetic. In some cases this works just as for $[\alpha]Q$, in other cases it is very different.

We start with assignment. This will always terminate in one step, so a property of all runs is the same as a property of one run.

$$\langle x \leftarrow e\rangle Q(x) \leftrightarrow \forall x'.x' = e \rightarrow Q(x') \quad (x' \text{ not in } e, Q(x))$$

Sequential composition also does not change matters in any essential way, although the reasoning is more subtle.

$$\langle\alpha\ ;\ \beta\rangle Q \leftrightarrow \langle\alpha\rangle(\langle\beta\rangle Q)$$

We argue as follows: there is a run of $\alpha\ ;\ \beta$ if there is a run of α to some intermediate state, and a run of β from there after which Q is true. And that's the same of running α to a state from which β can reach a state in which Q is true.

For nondeterministic choice $\alpha \cup \beta$, we can reach a final state either by choosing α or choosing β.

$$\langle\alpha \cup \beta\rangle Q \leftrightarrow \langle\alpha\rangle Q \vee \langle\beta\rangle Q$$

This is somehow dual to the axiom for $[\alpha \cup \beta]$:

$$[\alpha \cup \beta]Q \leftrightarrow [\alpha]Q \wedge [\beta]Q$$

Here we reasoned: to show that Q is true after every run of $\alpha \cup \beta$ it must be true after every possible run of α and also after every possible run of β.

Finally, for guards they are opposites in a different say.

$$\begin{array}{lcl} \langle ?P \rangle Q & \leftrightarrow & P \wedge Q \\ [?P]Q & \leftrightarrow & P \rightarrow Q \end{array}$$

Finally, we come to repetition. There is a simple analogue of the axiom to unroll a loop, but turning conjunction into a disjunction. That's because in order to reach a final state it is sufficient to unroll any fixed number of times (including zero).

$$\begin{array}{lcl} \langle \alpha^* \rangle Q & \leftrightarrow & Q \vee \langle \alpha \rangle \langle \alpha^* \rangle Q \\ [\alpha^*]Q & \leftrightarrow & Q \wedge [\alpha][\alpha^*]Q \end{array}$$

As before, this finite unrolling is of limited utility.

4 Convergence

In practice, unrolling a loop a finite number of times is insufficient to prove most programs. Instead, we work with the induction axiom and then invariants when proving $[\alpha^*]Q$. Recall:

$$\begin{array}{lcl} [\alpha^*]Q & \leftrightarrow & Q \wedge [\alpha^*](Q \rightarrow [\alpha]Q) \\ & \leftarrow & J \wedge \Box(J \rightarrow [\alpha]J) \wedge \Box(J \rightarrow Q) \end{array}$$

What is the analogue for $\langle \alpha^* \rangle Q$? Unfortunately, it is not as simple as induction but requires some quantity that reduces each time around the loop. That already might have been predicted from the fact that we use `variant` contracts in Why3.

To capture this logically we assume that a formula V is parameterized by an integer variable n, written as $V(n)$. We prohibit the variable n from appearing in programs; instead we use V to *relate* n to expressions occurring in the program. The axiom of convergence then says

> It is possible to reach a poststate with $V(0)$ after some iterations of α
> if (1) initially $V(n)$ for some $n \geq 0$,
> and (2) at each iteration, assuming $V(n)$ for $n > 0$
> implies we can reach a poststate with $V(n-1)$.

Translating this into an axioms gives us

$$\begin{array}{lcl} \langle \alpha^* \rangle V(0) & \leftarrow & (\exists n.\, n \geq 0 \wedge V(n)) \\ & & \wedge\, [\alpha^*](\forall n.\, n > 0 \wedge V(n) \rightarrow \langle \alpha \rangle V(n-1)) \\ & & (n \text{ not in } \alpha) \end{array}$$

It is interesting that this axiom incorporates $[\alpha^*]P$ because we need to make sure that no matter how many iterations we need until we reach 0 the decrease will always take place.

To make this effective we take one more step: we think of $V(n)$ as the *formula variant* of the iteration and use it to prove an arbitrary postcondition Q. As before, this replaces $[\alpha^*]P$ by $\Box P$, and makes sure the variant formula implies the postcondition. This is slightly different than the *variant expression* we use in Why3, which we address in the next section.

$$\begin{array}{rcl}\langle\alpha^*\rangle Q & \leftarrow & (\exists n.\, n \geq 0 \wedge V(n)) \\ & & \wedge\, \Box(\forall n.\, n > 0 \wedge V(n) \rightarrow \langle\alpha\rangle V(n-1)) \\ & & \wedge\, \Box(V(0) \rightarrow Q) \\ & & (n \text{ not in } \alpha \text{ or } Q)\end{array}$$

As an example, let's prove

$$x \geq 0 \rightarrow \langle(x \leftarrow x-1)^*\rangle x = 0$$

In order to apply convergence we have to define the variant formula $V(n)$. In this case, it is easy and we choose

$$V(n) = (x = n)$$

that is, n just tracks the value of x. We proceed:

To prove (init): $x \geq 0 \rightarrow \exists n.\, n \geq 0 \wedge x = n$ True (pick $n = x$)

To prove (step): $x \geq 0 \rightarrow \Box(\forall n.\, n > 0 \wedge x = n \rightarrow \langle x \leftarrow x-1\rangle x = n-1)$
True if $\forall n.\, n > 0 \wedge x = n \rightarrow \forall x'.\, x' = x-1 \rightarrow x' = n-1$
True if $\forall n.\, n > 0 \wedge x = n \rightarrow x-1 = n-1$ By arithmetic

To prove (post): $x \geq 0 \rightarrow \Box(x = 0 \rightarrow x = 0)$
True if $x = 0 \rightarrow x = 0$

To illustrate how we have to think about picking $V(n)$, consider the slightly more complicated example

$$x \geq 0 \rightarrow \langle(x \leftarrow x-2)^*\rangle 0 \leq x < 2$$

Consider what variant formula $V(n)$ might allow us to do this proof.

We pick $V(n) = 2n \leq x < 2n + 2$. Then $V(0) = 2 \cdot 0 \leq x < 2 \cdot 0 + 2$ and $V(n-1) = 2(n-1) \leq x < 2(n-1) + 2$. We reason:

To prove (init): $x \geq 0 \to \exists n.\, n \geq 0 \wedge 2n \leq x < 2n + 2$ True (pick $n = \mathsf{div}\, x\, 2$)

To prove (step): $x \geq 0 \to \Box(\, \forall n.\, n > 0 \wedge 2n \leq x < 2n + 2$
$\to \langle x \leftarrow x - 1 \rangle 2(n-1) \leq x < 2(n-1) + 2)$
True if $\forall n.\, n > 0 \wedge 2n \leq x < 2n + 2 \to 2n - 2 \leq x - 2 < 2n$ By arithmetic

To prove (post): $x \geq 0 \to \Box(2 \cdot 0 \leq x < 2 \cdot 0 + 2 \to 0 \leq x < 2)$
True if $0 \leq x < 2 \to 0 \leq x < 2$

5 Interactions Between Box and Diamond

Already, the axiom of convergence mixes $[\alpha]P$ and $\langle\alpha\rangle P$. This interaction is a bit tricky, so we consider a few simpler cases on how these modalities interact.

$$[\alpha](P \to Q) \to ([\alpha]P \to [\alpha]Q) \qquad \text{Valid}$$

If P implies Q in every poststate of α, then if P is also true in every poststate, so must Q be.

$$\langle\alpha\rangle(P \to Q) \to (\langle\alpha\rangle P \to \langle\alpha\rangle Q) \qquad \text{Not valid}$$

There is a poststate in which P implies Q and also a poststate in which P is true. Since these two poststate may be different, we cannot be certain that there will be a poststate in which Q is true.

$$[\alpha](P \to Q) \to (\langle\alpha\rangle P \to \langle\alpha\rangle Q) \qquad \text{Valid}$$

If P implies Q in every poststate of α, then this will also be true in the poststate in which P is true. Therefore, Q will be true in that poststate.

In the next two we explore the consequence of an invariant J

$$[\alpha]J \to (\langle\alpha\rangle(J \to Q) \to \langle\alpha\rangle Q) \qquad \text{Valid}$$

If J is true in every poststate of J, and there is a poststate where J implies Q, then Q must be true in that poststate.

$$[\alpha]J \to (\langle\alpha\rangle Q \to \langle\alpha\rangle(J \wedge Q)) \qquad \text{Valid}$$

If J is true in every poststate of J, and there is a poststate where Q is true, then both J and Q must be true in that poststate.

6 From Variant Formulas to Variant Expressions

We generalize the axiom of convergence with *variant formulas* to one with *variant expressions* allowing "big steps" where the expressions may decrease by more than 1. In this formulation we explicitly highlight an invariant J together with the variant expression e. Both of these may mention program variables but not the new variable n which tracks the value of the variant in the axiom. This closely approximates what the verification condition generator for Why3 does for while-loops.

One of the key ideas here is that the invariant may help us to establish the variant.

$$\begin{array}{lcl} \langle \alpha^* \rangle Q & \leftarrow & J \\ & & \wedge \, \Box(J \rightarrow e \geq 0) \\ & & \wedge \, \Box(\forall n. \, J \wedge e = n \rightarrow \langle \alpha \rangle (J \wedge e < n)) \\ & & \wedge \, \Box(J \rightarrow Q) \\ & & (n \text{ not in } J, e, \text{or } Q) \end{array}$$

In the version for while loops, we recall that

$$\mathsf{while}\, P\, \alpha \triangleq (?P \, ; \, \alpha)^* \, ; \, ?\neg P$$

which leads us to

$$\begin{array}{lcl} \langle \mathsf{while}\, P\, \alpha \rangle Q & \leftarrow & J \\ & & \wedge \, \Box(J \wedge P \rightarrow e \geq 0) \\ & & \wedge \, \Box(\forall n. \, J \wedge P \wedge e = n \rightarrow \langle \alpha \rangle (J \wedge e < n)) \\ & & \wedge \, \Box(J \wedge \neg P \rightarrow Q) \\ & & (n \text{ not in } J, P, e, \text{or } Q) \end{array}$$

As an example (which we did not go through in lecture) you may consider the following correctness statement for computing Fibonacci numbers, using simultaneous assignment as a shorthand.

$$x \geq 0 \rightarrow \langle a \leftarrow 0 \, ; \, b \leftarrow 1 \, ; \, i \leftarrow 0 \, ; \, \mathsf{while}\, (i < x) \, (a, b \leftarrow b, a + b \, ; \, i \leftarrow i + 1) \rangle \, a = \mathsf{fib}\, x$$

To conduct this proof we pick

$$\begin{array}{lcll} e & = & (x - i) & \text{variant expression} \\ J & = & (0 \leq i \leq x \wedge a = \mathsf{fib}(i) \wedge b = \mathsf{fib}(i + 1)) & \text{invariant} \end{array}$$

It is then a mechanical exercise to verify the conditions of the axiom for while with invariants and variant expressions.

7 Extending Our Why3 Formalization of Dynamic Logic

We can extend our formalization of dynamic logic by adding $\langle \alpha \rangle Q$ as a new kind of formula and prove the new axioms (not including convergence). We only show some excerpts; our live code can be found in the file ndl.mlw.

```
axiom models_dia : forall omega alpha q.
  models omega (Dia alpha q) <-> exists nu. run omega alpha nu /\ models nu q

lemma dia_seq : forall omega alpha beta q.
  models omega (Dia (Seq alpha beta) q) <-> models omega (Dia alpha (Dia beta q))

lemma dia_union : forall omega alpha beta q.
  models omega (Dia (Union alpha beta) q) <-> models omega (Or (Dia alpha q) (Dia beta q))

(* unrolling *)
lemma dia_star : forall omega alpha q.
  models omega (Dia (Star alpha) q) <-> models omega (Or q (Dia alpha (Dia (Star alpha) q)))

(* added after lecture *)
lemma dia_guard : forall omega t q.
  models omega (Dia (Guard t) q) <-> models omega (And (Test t) q)
```