Weakest Preconditions

William Schultz

February 22, 2023

The notion of *weakest precondition* in program analysis derives from the work of Dijkstra [1]. He introduced the *Guarded Command Language* as a simple modeling language for program specification e.g.

Dijkstra introduced the Guarded Command Language (GCL) with the grammar

$$
\begin{aligned}
S \;::=\;& \mathsf{skip} \;\mid\; x := E \;\mid\; S_1;\; S_2 \\
&\mid\; \mathsf{if}\; B_1 \rightarrow S_1 \;[]\; B_2 \rightarrow S_2 \;[]\; \cdots \;[]\; B_n \rightarrow S_n \;\mathsf{fi} \\
&\mid\; \mathsf{do}\; B_1 \rightarrow S_1 \;[]\; B_2 \rightarrow S_2 \;[]\; \cdots \;[]\; B_n \rightarrow S_n \;\mathsf{od}
\end{aligned}
$$

where the B_i are Boolean expressions. The B_i are called *guards* because they guard the corresponding statements S_i. The symbol $[]$ is the *nondeterministic choice operator* and is not to be confused with $|$. In if and do statements, a clause $B_i \rightarrow S_i$ is said to be *enabled* if its guard B_i is true.

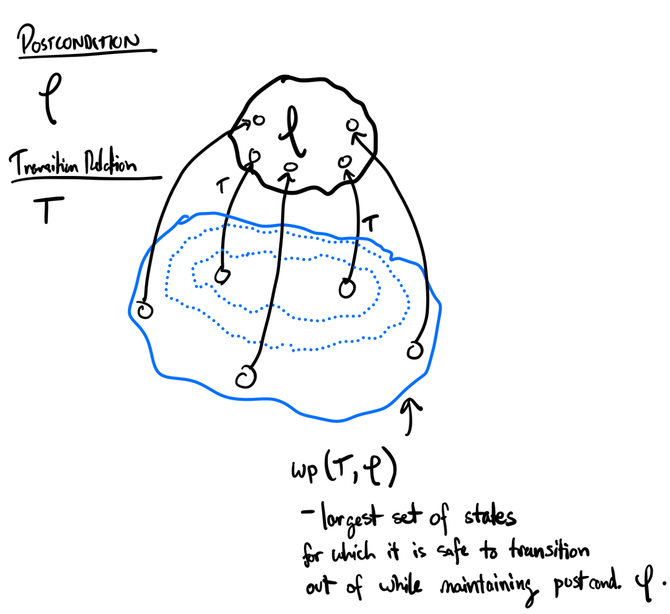
Given a program S and a postcondition φ, we define the associated *weakest precondition*, denoted $wp(S, \varphi)$, as the weakest property of the input state that guarantees that S will terminate with the postcondition φ. In the context of GCL, we can provide basic rules for how to compute the weakest precondition for various program statements. For example, for an assignment statement $x := E$, we have

$$wp(x := E, \varphi) \equiv \varphi\{E/x\}$$

where $\varphi\{E/x\}$ represents the property φ with appearances of x in φ replaced with E. For example,

$$
\begin{aligned}
wp(x := x + 1, x = 3) &\equiv (x = 3)\{x + 1/x\} \\
&\equiv (x + 1) = 3 \\
&\equiv x = 2
\end{aligned}
$$

We can also think about weakest preconditions from a more semantic perspective. If we have a symbolic transition relation T and a postcondition φ (i.e. a state predicate), the weakest precondition of T with respect to φ is the weakest predicate P (in other words, the largest set of states) such that a transition out of any state in P will uphold the property φ.



So, we can also consider weakest precondition computation as a kind of *backwards symbolic execution*. That is, we start from a given postcondition predicate, and a given transition relation, and execute the transition relation backwards to compute the states contained in the weakest precondition.

References

[1] Edsger W. Dijkstra. Guarded commands, nondeterminacy and formal derivation of programs. *Commun. ACM*, 18(8):453–457, aug 1975.