4.- Verification Conditions Generation

Both

1. the different types of assertions (requires, ensures, assert, invariant, ...) written by the user in a Dafny file, and
2. the verification conditions (or proof obligations) generated by Dafny to be sent to Z3

are first-order formulas.

The latter (VC) are always implications inferred from the former and the program.

Syntax and semantics of first-order formulas

- A signature Σ consist of
  - Σ_F = Set of function symbols
  - Σ_P = Set of predicate symbols

  along with a function arity : $\Sigma \to Nat$ that associates to each symbol its number of parameters.
- Constants are function symbols with arity 0.
- Propositions are predicate symbols with arity 0.
- $\mathcal{V}$ is an infinite numerable set of variable symbols.
- Examples: $+$ is a function of arity 2, but odd is a predicate of arity 1.
- In the sequel $s|n \in \Sigma$ denotes $s \in \Sigma$ and $arity(s) = n$.

FOL Syntax: Terms

- Terms represent individuals of the universe (of discourse).
- The set $\mathcal{T}(\Sigma, \mathcal{V})$ of all well-formed terms over Σ and $\mathcal{V}$ is the least set such that:
  - $\mathcal{V} \subseteq \mathcal{T}(\Sigma, \mathcal{V})$
  - $f(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \mathcal{V})$ if $f|n \in \Sigma_F$ and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{V})$.

- Exercise: Let $\Sigma_F = \{c|0, f|1, g|2, h|1\}$ which of the following are well-formed terms:
  1. $c(f)$
  2. $f(c)$
  3. $f(g(a))$
  4. $g(h(c), f(c))$
  5. $h(g(c, h(c)))$

FOL Syntax: Formulas

- The set $\mathcal{F}(\Sigma, \mathcal{V})$ of all well-formed formulas over Σ and $\mathcal{V}$ is the least set such that:
  - $\text{True}, \text{False} \in \mathcal{F}(\Sigma, \mathcal{V})$
  - $p(t_1, \ldots, t_n) \in \mathcal{F}(\Sigma, \mathcal{V})$ if $p|n \in \Sigma_P$ and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{V})$
  - If $\varphi \in \mathcal{F}(\Sigma, \mathcal{V})$ then $\neg\varphi \in \mathcal{F}(\Sigma, \mathcal{V})$
  - If $\varphi_1, \varphi_2 \in \mathcal{F}(\Sigma, \mathcal{V})$ then $\varphi_1 \wedge \varphi_2,\ \varphi_1 \vee \varphi_2,\ \varphi_1 \rightarrow \varphi_2,\ \varphi_1 \leftrightarrow \varphi_2 \in \mathcal{F}(\Sigma, \mathcal{V})$
  - If $\varphi \in \mathcal{F}(\Sigma, \mathcal{V})$ and $x \in \mathcal{V}$ then $\forall x\varphi,\ \exists x\varphi \in \mathcal{F}(\Sigma, \mathcal{V})$

Examples

Signature and variables

$$\Sigma = \{a/0, b/0, f/1, g/2, h/1, r/0, P/2, Q/1, S/2\}$$
$$\mathcal{V} = \{x, y, z, u, v, w, dots\}$$

Atoms:

r, $P(a, b)$, $Q(x)$, $S(h(a), w)$, $S(h(f(a)), g(f(y), b))$,
$(a = x)$, $(f(w) = g(h(a), f(b)))$, ...

Compound formulas:

$r \rightarrow \neg Q(f(x))$,
$\forall x(P(x, x) \wedge (w = v))$,
$\forall x(Q(h(f(b))) \vee \exists y(P(x, y)))$,
...

FOL Syntax: Free Variables.

- In $\forall x(\varphi)$ and $\exists x(\varphi)$, φ is in the scope of $\forall x$ and $\exists x$.
- An occurrence of x is *bound* if it is in the scope of some $\forall x$ or $\exists x$, otherwise it is a *free* occurrence.
- A variable is free in a formula φ if it has at least one free occurrence in φ.

- The set $FV(\varphi)$ of all free variables of a formula φ can be defined as follows:
  - $FV(\text{True}) = FV(\text{False}) = \emptyset$.
  - $FV(p(t_1, \ldots, t_n)) = var(t_1) \cup \cdots \cup var(t_n)$ where $var(t)$ is the set of all variables that occur in t.
  - $FV(\neg\varphi) = FV(\varphi)$
  - $FV(\varphi \wedge \psi) = FV(\varphi \vee \psi) = FV(\varphi \rightarrow \psi) = FV(\varphi \leftrightarrow \psi) = FV(\varphi) \cup FV(\psi)$
  - $FV(\forall x(\varphi)) = FV(\exists x(\varphi)) = FV(\varphi) \setminus \{x\}$ where $\setminus$ is set difference.

Example: $\varphi = (\forall x \exists y R(x, f(y)) \land (\forall z \neg (h(z,z) = f(y)))$

$\varphi = (\forall x \exists y R(x, f(y)) \land (\forall z \neg (h(z,z) = f(y)))$

- $\Sigma = \{f/1, h/2, R/2\}$
- Scope of $\forall x$: $\varphi_1 = \exists y R(x, f(y))$
- Scope of $\exists y$: $\varphi_2 = R(x, f(y))$
- Scope of $\forall z$: $\varphi_3 = \neg (h(z,z) = f(y))$
- $FV(\forall x \varphi_1) = \varnothing$
- $FV(\exists y \varphi_2) = \{x\}$
- $FV(\forall z \varphi_2) = \{y\}$
- $FV(\varphi) = \{y\}$
- Terms in φ: x, y, z, $f(y)$ and $h(z,z)$.
- φ has exactly one free occurrence of a variable.

- If $\varphi \in \mathcal{F}(\Sigma, \mathcal{V})$, $x_1, \ldots, x_n \in \mathcal{V}$ and $t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{V})$, then

  $$\varphi[t_1, \ldots, t_n / x_1, \ldots, x_n]$$

  denotes the formula obtained by simultaneously substituting in φ every free occurrence of x_i by t_i.

- We denote by $\varphi^{\forall}$ the sentence $\forall x_1 \ldots \forall x_n (\varphi)$ where $x_1, \ldots, x_n = FV(\varphi)$.
- A sentence is a formula without free variables.

Dijkstra Weakest Precondition

Given a code fragment P and postcondition ψ, find the unique formula $\mathsf{wp}(P, \psi)$ which is the *weakest* precondition for P and ψ.

- is a precondition: $\{\mathsf{wp}(P, \psi)\}P\{\psi\}$ is true.
- is the weakest one: $(\varphi \rightarrow \mathsf{wp}(P, \psi))^{\forall}$ is valid for any φ such that $\{\varphi\}P\{\psi\}$ is true.

Calculating the WP

- $\mathsf{wp}(\overline{\mathtt{x}} := \overline{\mathtt{t}},\ \psi) = \psi[\overline{\mathtt{t}}/\overline{\mathtt{x}}]$

For example:

$\mathsf{wp}(\mathtt{x}, \mathtt{y} := \mathtt{y} + \mathtt{1}, \mathtt{x} - \mathtt{1},\ x * y = 0)$
$= (y + 1) * (x - 1) = 0$
$= (y = -1) \vee (x = 1)$ is weaker than $x = 1$
is weaker than $(y = -1) \wedge (x = 1)$

- $\mathsf{wp}(\overline{\mathtt{x}} := \overline{\mathtt{t}},\ \psi) = \psi[\overline{\mathtt{t}}/\overline{\mathtt{x}}]$
- $\mathsf{wp}(\mathtt{P}_1;\mathtt{P}_2,\ \psi) = \mathsf{wp}(\mathtt{P}_1,\ \mathsf{wp}(\mathtt{P}_2,\ \psi))$

For example:

$$
\begin{aligned}
&\mathsf{wp}(\mathtt{x:=y+1;\ \ y:=x-1},\ x * y = 0)\\
=\ &\mathsf{wp}(\mathtt{x:=y+1},\ \mathsf{wp}(\mathtt{y:=x-1},\ x * y = 0))\\
=\ &\mathsf{wp}(\mathtt{x:=y+1},\ x * (x - 1) = 0)\\
=\ &(y + 1) * y = 0\\
=\ &(y = -1) \lor (y = 0)
\end{aligned}
$$

- $\mathsf{wp}(\overline{\mathtt{x}} := \overline{\mathtt{t}},\ \psi) = \psi[\overline{\mathtt{t}}/\overline{\mathtt{x}}]$
- $\mathsf{wp}(\mathtt{P}_1 ; \mathtt{P}_2,\ \psi) = \mathsf{wp}(\mathtt{P}_1,\ \mathsf{wp}(\mathtt{P}_2,\ \psi))$
- $\mathsf{wp}(\mathtt{if}\ b\ \mathtt{then}\ \mathtt{P}_1\ \mathtt{else}\ \mathtt{P}_2,\ \psi) = (b \to \mathsf{wp}(\mathtt{P}_1,\ \psi)) \wedge (\neg b \to \mathsf{wp}(\mathtt{P}_2,\ \psi))$

For example:

$$
\begin{aligned}
&\mathsf{wp}(\mathtt{if}\ \mathtt{x} \geq \mathtt{y}\ \mathtt{then}\ \mathtt{z:=x}\ \mathtt{else}\ \mathtt{z:=y},\ z = max(x,y)) \\
&= (x \geq y \to \mathsf{wp}(\mathtt{z:=x},\ z = max(x,y))) \wedge \\
&\quad (\neg(x \geq y) \to \mathsf{wp}(\mathtt{z:=y},\ z = max(x,y))) \\
&= (x \geq y \to x = max(x,y)) \wedge (\neg(x \geq y) \to y = max(x,y))
\end{aligned}
$$

- $\mathtt{wp}(\overline{\mathtt{x}} := \overline{\mathtt{t}},\ \psi) = \psi[\overline{\mathtt{t}}/\overline{\mathtt{x}}]$
- $\mathtt{wp}(\mathtt{P}_1; \mathtt{P}_2,\ \psi) = \mathtt{wp}(\mathtt{P}_1,\ \mathtt{wp}(\mathtt{P}_2,\ \psi))$
- $\mathtt{wp}(\mathtt{if}\ b\ \mathtt{then}\ \mathtt{P}_1\ \mathtt{else}\ \mathtt{P}_2,\ \psi) = (b \to \mathtt{wp}(\mathtt{P}_1,\ \psi)) \wedge (\neg b \to \mathtt{wp}(\mathtt{P}_2,\ \psi))$
- $\mathtt{wp}(\mathtt{skip},\ \psi) = \psi$
- $\mathtt{wp}(\mathtt{while}\ b\ \mathtt{do}\ \mathtt{P}\ ,\ \psi) = \alpha$ provided that
  - $(\alpha \wedge b) \to \mathtt{wp}(\mathtt{P}, \alpha)$
  - $(\alpha \wedge \neg b) \to \psi$

$\mathtt{wp}(\mathtt{while}\ (r+1)*(r+1) \leq x$ do $r := r+1,$
$r * r \leq x < (r+1)*(r+1)) = r*r \leq x$ provided that

- $(r*r \leq x \ \wedge (r+1)*(r+1) \leq x) \to \mathrm{wp}(r{:=}r+1,\ r*r \leq x)$
- $(\ r*r \leq x \wedge \neg(\ (r+1)*(r+1) \leq x)) \to r * r \leq x < (r+1)*(r+1)$

- $\mathtt{wp}(\overline{\mathtt{x}} := \overline{\mathtt{t}},\ \psi) = \psi[\overline{\mathtt{t}}/\overline{\mathtt{x}}]$
- $\mathtt{wp}(\mathtt{P}_1;\mathtt{P}_2,\ \psi) = \mathtt{wp}(\mathtt{P}_1,\ \mathtt{wp}(\mathtt{P}_2,\ \psi))$
- $\mathtt{wp}(\mathtt{if}\ b\ \mathtt{then}\ \mathtt{P}_1\ \mathtt{else}\ \mathtt{P}_2,\ \psi) = (b \rightarrow \mathtt{wp}(\mathtt{P}_1,\ \psi)) \wedge (\neg b \rightarrow \mathtt{wp}(\mathtt{P}_2,\ \psi))$
- $\mathtt{wp}(\mathtt{skip},\ \psi) = \psi$
- $\mathtt{wp}(\mathtt{while}\ b\ \mathtt{do}\ \mathtt{P}\ ,\ \psi) = \alpha$ provided that
  - $(\alpha \wedge b) \rightarrow \mathtt{wp}(\mathtt{P},\alpha)$
  - $(\alpha \wedge \neg b) \rightarrow \psi$

$\mathtt{wp}(\mathtt{while}\ (\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x}\ \mathsf{do}\ \mathsf{r} := \mathsf{r}+1,$
$\mathsf{r} * \mathsf{r} \leq \mathsf{x} < (\mathsf{r}+1)*(\mathsf{r}+1)) = \mathsf{r}*\mathsf{r} \leq \mathsf{x}$ provided that

- $(\mathsf{r}*\mathsf{r} \leq \mathsf{x}\ \wedge (\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x}) \rightarrow (\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x}$
- $(\ \mathsf{r}*\mathsf{r} \leq \mathsf{x} \wedge \neg(\ (\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x})) \rightarrow \mathsf{r} * \mathsf{r} \leq \mathsf{x} < (\mathsf{r}+1)*(\mathsf{r}+1)$

- $\mathsf{wp}(\overline{\mathtt{x}} := \overline{\mathtt{t}},\ \psi) = \psi[\overline{\mathtt{t}}/\overline{\mathtt{x}}]$
- $\mathsf{wp}(\mathtt{P}_1; \mathtt{P}_2,\ \psi) = \mathsf{wp}(\mathtt{P}_1,\ \mathsf{wp}(\mathtt{P}_2,\ \psi))$
- $\mathsf{wp}(\mathtt{if}\ b\ \mathtt{then}\ \mathtt{P}_1\ \mathtt{else}\ \mathtt{P}_2,\ \psi) = (b \to \mathsf{wp}(\mathtt{P}_1,\ \psi)) \land (\neg b \to \mathsf{wp}(\mathtt{P}_2,\ \psi))$
- $\mathsf{wp}(\mathtt{skip},\ \psi) = \psi$
- $\mathsf{wp}(\mathtt{while}\ b\ \mathtt{do}\ \mathtt{P}\ ,\ \psi) = \alpha$ provided that
  - $(\alpha \land b) \to \mathsf{wp}(\mathtt{P}, \alpha)$
  - $(\alpha \land \neg b) \to \psi$

$\{\varphi\}P\{\psi\}$ iff

- $\varphi \to \mathsf{wp}(P, \psi)$ and
- all the provisos for calculating $\mathsf{wp}(P, \psi)$

are all them valid sentences (after $(\_)^{\forall}$).

Verification Condition Generation

$$\text{VCG}(\{\varphi\}P\{\psi\}) = \{\ \varphi \rightarrow \text{wp}(P,\psi)\ \} \cup \mathbf{vc}(P,\psi)$$

where

- $\mathbf{vc}(\overline{\mathtt{x}} := \overline{\mathtt{t}},\ \psi) = \mathbf{vc}(\mathtt{skip},\ \psi) = \emptyset$
- $\mathbf{vc}(\mathtt{P}_1;\mathtt{P}_2,\ \psi) = \mathbf{vc}(\mathtt{P}_1,\ \text{wp}(\mathtt{P}_2,\ \psi)) \cup \mathbf{vc}(\mathtt{P}_2, \psi)$
- $\mathbf{vc}(\mathtt{if}\ b\ \mathtt{then}\ \mathtt{P}_1\ \mathtt{else}\ \mathtt{P}_2,\ \psi) = \mathbf{vc}(\mathtt{P}_1,\ \psi) \cup \mathbf{vc}(\mathtt{P}_2,\ \psi)$
- $\mathbf{vc}(\mathtt{while}\ b\ \mathtt{do}\ \mathtt{P},\ \psi) =$
$$\{\ (Inv \wedge b) \rightarrow \text{wp}(\mathtt{P},Inv),\ (Inv \wedge \neg b) \rightarrow \psi\ \} \cup \mathbf{vc}(P,Inv)$$

Inv is the (inferred/user-defined) invariant of the iteration `while` b `do P`

Example

$$Q \equiv \mathsf{r} := 0;\ \texttt{while}\ (\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x}\ \mathsf{do}\ \mathsf{r} := \mathsf{r}+1;$$

$\mathsf{VCG}(\{\ x \geq 0\ \}\ Q\ \{\ r * r \leq x < (r+1)*(r+1)\ \})$

$=$ [since $\mathsf{VCG}(\{\varphi\}P\{\psi\}) = \{\ \varphi \to \mathsf{wp}(P,\psi)\ \} \cup \mathbf{vc}(P,\psi)$]

$\{\ x \geq 0 \to \mathsf{wp}(Q, r * r \leq x < (r+1)*(r+1)\ )\}$
$\cup\ \mathbf{vc}(Q, r * r \leq x < (r+1)*(r+1))$

$=$ [since $\mathsf{wp}(P_1; P_2,\ \psi) = \mathsf{wp}(P_1,\ \mathsf{wp}(P_2,\ \psi))$ and
$\mathsf{wp}(\ \textbf{while}\ b\ \ \textbf{do}\ P\ ,\ \psi) = \alpha$]

$\{x \geq 0 \to \mathsf{wp}(r := 0;, r * r \leq x\ )\} \cup \mathbf{vc}(Q, r * r \leq x < (r+1)*(r+1))$

$=$ [since $\mathsf{wp}(\overline{x} := \bar{t},\ \psi) = \psi[\bar{t}/\overline{x}]$]

$\{x \geq 0 \to 0 * 0 \leq x\} \cup \mathbf{vc}(Q,\ r * r \leq x < (r+1)*(r+1))$

$= \mathbf{vc}(Q,\ r * r \leq x < (r+1)*(r+1))$

$$\mathsf{Q} \equiv \mathsf{r} := 0;\ \texttt{while}\ (\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x}\ \texttt{do}\ \mathsf{r} := \mathsf{r}+1$$

$\mathbf{vc}(\mathsf{Q}, \mathsf{r} * \mathsf{r} \leq \mathsf{x} < (\mathsf{r}+1)*(\mathsf{r}+1))$

$= [$ since $\mathbf{vc}(P_1; P_2,\ \psi) = \mathbf{vc}(P_1,\ \mathsf{wp}(P_2,\ \psi)) \cup \mathbf{vc}(P_2, \psi)\ ]\ ]$

$\mathbf{vc}(\mathsf{r}{:=}0, \mathsf{r} * \mathsf{r} \leq \mathsf{x})$
$\cup\ \mathbf{vc}(\texttt{while}\ (\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x}\ \texttt{do}\ \mathsf{r} := \mathsf{r}+1\ ,$
$\mathsf{r} * \mathsf{r} \leq \mathsf{x} < (\mathsf{r}+1)*(\mathsf{r}+1))$

$= [$ since $\mathbf{vc}(\mathsf{x}{:=}\mathsf{t},\ \psi) = \emptyset$ and $\mathbf{vc}(\textbf{while}\ b\ \ \textbf{do}\ P,\ \psi)$
$= \{(Inv \wedge b) \to \mathsf{wp}(P, Inv),\ (Inv \wedge \neg b) \to \psi\} \cup \mathbf{vc}(P, Inv)\ ]$

$\{\ (\mathsf{r}*\mathsf{r} \leq \mathsf{x} \wedge (\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x}) \to (\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x},$
$(\mathsf{r}*\mathsf{r} \leq \mathsf{x} \wedge \neg((\mathsf{r}+1)*(\mathsf{r}+1) \leq \mathsf{x})) \to \mathsf{r} * \mathsf{r} \leq \mathsf{x} < (\mathsf{r}+1)*(\mathsf{r}+1)\ \}$