MINI-COURSE on compositional methods of program verification: TABLES

Table 1: EL syntax

prg::= begin c end

c::= x:=a | c_1 ; c_2| if b then c_1 else c_2 |while b do c | begin c end | skip

a::= n | x | $a_1 + a_2$ | $a_1 - a_2$ | $a_1 * a_2$ | (a)

b::=a_1= a_2 |a_1>a_2 | $b_1 \vee b_2$ |$\neg b$ | (b),

where:

- n ranges over natural numbers Nat={0, 1, 2, …},
- x ranges over variables (names) V={M, N, …},
- a ranges over arithmetic expressions $Aexpr$,
- b ranges over Boolean expressions $Bexpr$,
- c ranges over commands (statements) (programs) Cmd,
- prg ranges over programs Prg.

Table 2: EL data operations

Let V be a set of names (variables). Then the set of partial functions $State = V \xrightarrow{p} Nat$ is a set of states.

Operation of overwriting (updating) $st \nabla st' = st' \cup [\, v \to n \in st \mid \neg \exists k\, (v \to k \in st')\,]$.

E.g.: $[M \mapsto 8, N \mapsto 16] \nabla [M \mapsto 3, X \mapsto 4, Y \mapsto 2] = [M \mapsto 3, N \mapsto 16, X \mapsto 4, Y \mapsto 2]$.

Operations over data sorts:

- n-ary operations: +, – , *, =, >, $\vee$, $\neg$.
- quasiary operations
  - *naming* $\Rightarrow x(n)=[x \mapsto n]$;
  - *denomination* $x \Rightarrow (st)=st(x)$ (or $\acute{x}(st)=st(x)$);
  - *constant* function $\overline{n} : State \to Nat$, $\overline{n}\,(st)=n$ ($n \in Nat$);
  - *identity* $id(st)=st$.

Table 3: Data algebra

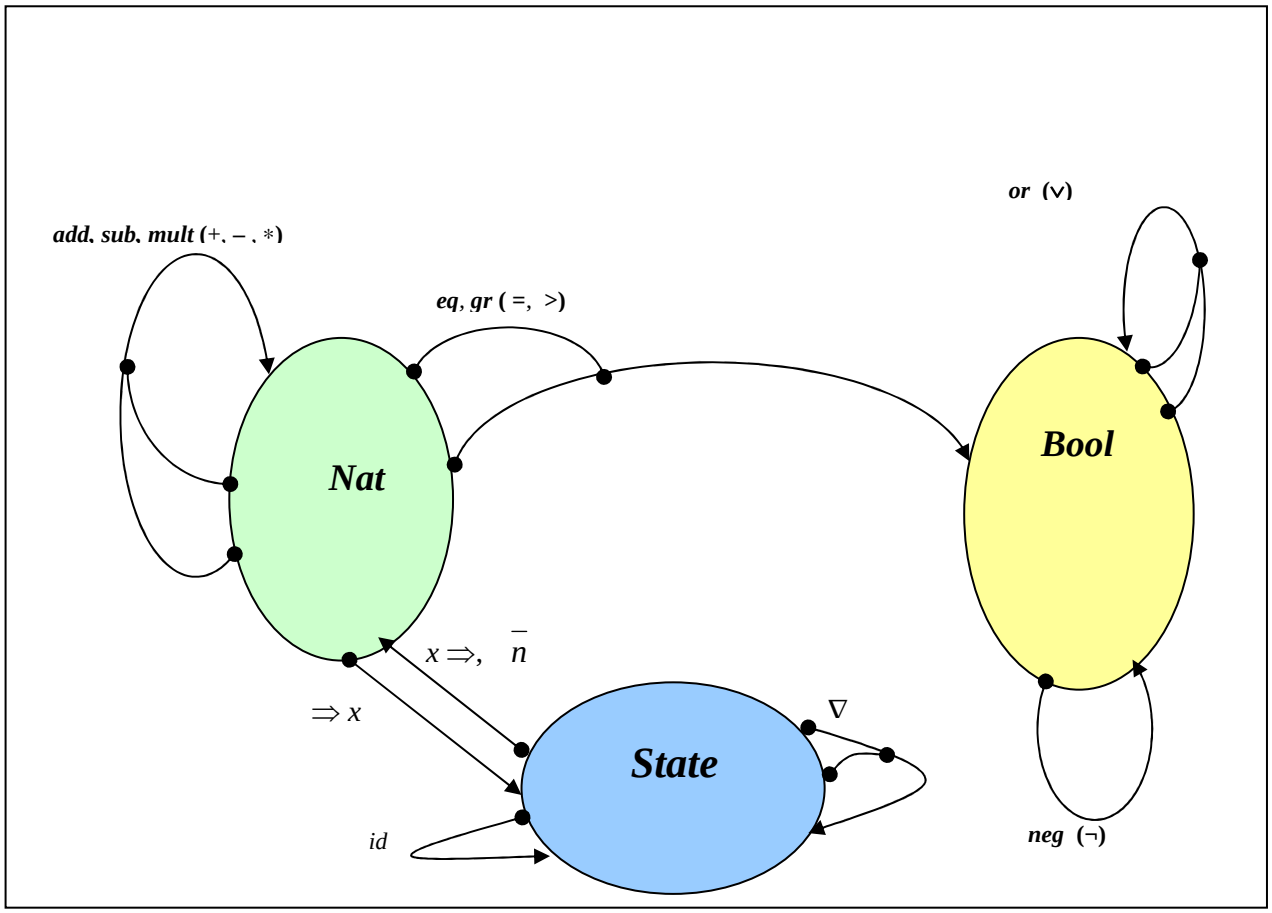
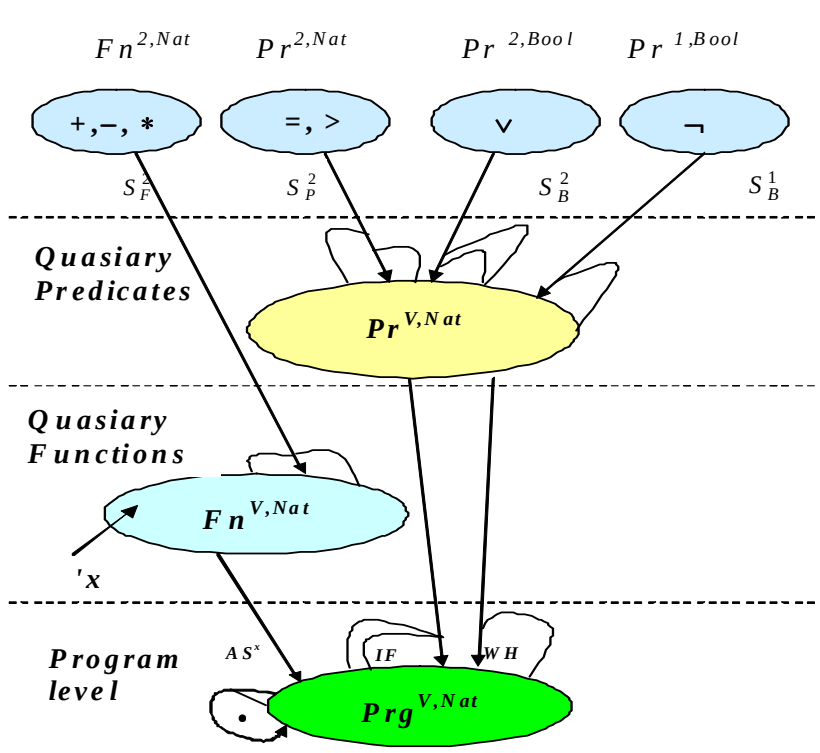


Table4: Program algebra

$$APn(V) = < Fn^{2,Nat}, Pr^{2,Bool}, Pr^{1,Bool}, Pr^{2,Nat}, Fn^{V,Nat}, Pr^{V,Nat}, Prg^{V,Nat};$$
$$+, -, *, \vee, \neg, =, >, S_F^2, S_P^2, S_B^2, S_B^1, 'x, AS^x, \bullet, IF, WH> .$$



$Fn^{n,Nat} = Nat^n \xrightarrow{p} Nat$

$Pr^{n,Nat} = Nat^n \xrightarrow{p} Bool$

$Pr^{n,Bool} = Bool^n \xrightarrow{p} Bool.$

$Pr^{V,Nat} = State \xrightarrow{p} Bool =$
$= {}^{V}Nat \xrightarrow{p} Bool$

$Fn^{V,Nat} = State \xrightarrow{p} Nat = {}^{V}Nat \xrightarrow{p} Nat$

$Prg^{V,Nat} = {}^{V}Nat \xrightarrow{p} {}^{V}Nat$

biquasiary functions (program functions)

Table 5: EL Program Compositions

Composition	Formula
Superposition	$(S_T^n(f, g_1,\ldots, g_n))(st) = f(g_1(st),\ldots, g_n(st)),\ T \in \{B,P,F\}$
Assignment	$AS^x(fa)(st) = st\,\nabla[x \mapsto fa(st)]$
Sequential execution	$fs_1 \bullet fs_2(st) = fs_2(fs_1(st))$
Conditional operator	$IF(fb, fs1, fs2)(st) = \begin{cases} fs1(st), & \text{if } fb(st) = true, \\ fs2(st), & \text{if } fb(st) = false. \end{cases}$
Loop	$WH(fb,fs)(st) = st_n$, where $st_0 = st$, $st_1 = fs(st_0)$, $st_2 = fs(st_1)$, …, $st_n = fs(st_{n-1})$, such that $fb(st_0) = true$, $fb(st_1) = true$,…, $fb(st_{n-1}) = true$, $fb(st_n) = false$.
Denomination function	$x\Rightarrow(st) = st(x)$ (or $'x\,(st) = st(x)$)
Identity function	$id(st) = st$

Table 6: Transformation rules from EL to Semantic terms (Denotational Semantics)

$[\![\text{begin } c \text{ end}]\!] = [\![c]\!]$
$[\![x := a]\!] = AS^x([\![a]\!])$
$[\![c_1 ; c_2]\!] = [\![c_1]\!] \bullet [\![c_2]\!]$
$[\![\text{ if } b \text{ then } c_1 \text{ else } c_2]\!] =$
$= IF([\![b]\!], [\![c_1]\!], [\![c_2]\!])$
$[\![\text{while } b \text{ do } c]\!] = WH([\![b]\!], [\![c]\!])$
$[\![\text{begin } c \text{ end}]\!] = [\![c]\!]$
$[\![skip]\!] = id$
$[\![n]\!] = \bar{n}$
$[\![x]\!] = 'x$ (or $x\Rightarrow$)

$[\![a_1 + a_2]\!] = S_F^2(+, [\![a_1]\!], [\![a_2]\!])$
$[\![a_1 - a_2]\!] = S_F^2(-, [\![a_1]\!], [\![a_2]\!])$
$[\![a_1 * a_2]\!] = S_F^2(*, [\![a_1]\!], [\![a_2]\!])$
$[\![(a)]\!] = [\![a]\!]$
$[\![b_1 = b_2]\!] = S_P^2(=, [\![b_1]\!], [\![b_2]\!])$
$[\![b_1 > b_2]\!] = S_P^2(>, [\![b_1]\!], [\![b_2]\!])$
$[\![b_1 \vee b_2]\!] = S_B^2(\vee, [\![b_1]\!], [\![b_2]\!])$
$[\![\neg b]\!] = S_B^1(\neg, [\![b]\!])$

Table 7: Floyd-Hoare rules for terms of program algebra with total predicates

Let $S_P^{[x]}(P, fa)(st)=P(st\nabla[x\mapsto fa(st)])$

Inference rule	Rule #
$\{S^{[x]}(P, fa)\}\ AS^x(fa)\ \{P\}$	*Ax_AS*
$\{P\}id\{P\}$	*Ax_id*
$\dfrac{\{P\}f\{Q\},\{Q\}g\{R\}}{\{P\}\ f\bullet g\ \{R\}}$	*Ax_SEQ*
$\dfrac{\{fb\wedge P\}f\{Q\},\{\neg fb\wedge P\}g\{Q\}}{\{P\}IF(fb, f, g)\{Q\}}$	*Ax_IF*
$\dfrac{\{fb\wedge P\}fs\{P\}}{\{P\}WH(fb, fs)\{\neg fb\wedge P\}}$	*Ax_WH*
$\dfrac{\{P'\}fs\{Q'\}}{\{P\}fs\{Q\}}$, if $P\Rightarrow P'$, $Q'\Rightarrow Q$	*Ax_CONS*

Table 8: Floyd-Hoare rules for EL programs with total predicates

Let $P[x\mapsto a](st)=P(st\nabla[x\mapsto [\![a]\!](st)])$ – semantic understanding

or $P[x\mapsto a]$ is substitution of a into P instead of x – syntactical understanding

Inference rule	Rule #
$\{P[x\mapsto a]\}x:=a\{P\}$	*AS*
$\{P\}skip\{P\}$	*skip*
$\dfrac{\{P\}S1\{Q\},\{Q\}S2\{R\}}{\{P\}S1; S2\{R\}}$	*SEQ*
$\dfrac{\{b\wedge P\}S1\{Q\},\{\neg b\wedge P\}S2\{Q\}}{\{P\}\text{if b then } S1\text{ else } S2\{Q\}}$	*IF*
$\dfrac{\{b\wedge P\}S\{P\}}{\{P\}\text{while b do } S\ \{\neg b\wedge P\}}$	*WH*
$\dfrac{\{P'\}S\{Q'\}}{\{P\}S\{Q\}}$, if $P\Rightarrow P'$, $Q'\Rightarrow Q$	*CONS*

Example:

1. EL program $GCD^{M,N}$=begin while $\neg(M=N)$ do if $M>N$ then $M:=M-N$ else $N:=N-M$ end

2. Semantic term: $STR^{M,N}=WH(S_B^1(\neg, S_P^2(=,'M,'N)), IF(S_P^2(>,'M,'N), AS^M(S_F^2(-,'M,'N)), AS^N(S_F^2(-,'N,'M))))$.

3. "Testing" of the term on the state $[M\mapsto 8, N\mapsto 16]$:

$$WH(S_B^1(\neg, S_P^2(=,'M,'N)), IF(S_P^2(>,'M,'N), AS^M(S_F^2(-,'M,'N)), AS^N(S_F^2(-,'N,'M))))([M\mapsto 8, N\mapsto 16]).$$

First we evaluate $S_B^1(\neg, S_P^2(=,'M,'N))([M\mapsto 8, N\mapsto 16])=\neg(S_P^2(=,'M,'N)([M\mapsto 8, N\mapsto 16]))=$

$$=\neg(=('M([M\mapsto 8, N\mapsto 16]), 'N([M\mapsto 8, N\mapsto 16]))=\neg(=(8, 16))=\neg(F)=T$$

Then we evaluate the body of the loop (if-statement) in two steps:

$S_P^2(>,'M,'N)([M\mapsto 8, N\mapsto 16])=\ >('M([M\mapsto 8, N\mapsto 16]),'N([M\mapsto 8, N\mapsto 16]))=\ >(8, 16)=F$;

$AS^N(S_F^2(-,'N,'M))([M\mapsto 8, N\mapsto 16])=[M\mapsto 8, N\mapsto 16]\nabla[N\mapsto S_F^2(-,'N,'M)([M\mapsto 8, N\mapsto 16])]=$

$=[M\mapsto 8, N\mapsto 16]\nabla[N\mapsto -('N([M\mapsto 8, N\mapsto 16]),'M([M\mapsto 8, N\mapsto 16]))=[M\mapsto 8, N\mapsto 16]\nabla[N\mapsto -(16, 8)]=$

$=[M\mapsto 8, N\mapsto 16]\nabla[N\mapsto 8]=[M\mapsto 8, N\mapsto 8]$.

At last, we evaluate the loop condition on the state obtained:

$S_B^1(\neg, S_P^2(=,'M,'N))([M\mapsto 8, N\mapsto 8])=\ \neg(S_P^2(=,'M,'N)([M\mapsto 8, N\mapsto 8]))=$

$$=\ \neg(=('M([M\mapsto 8, N\mapsto 8]),'N([M\mapsto 8, N\mapsto 8]))=\ \neg(=(8, 8))=\ \neg(T)=F.$$

Loop condition was evaluated to F (false), so, the state $[M\mapsto 8, N\mapsto 8]$ is final. In this state both variables are equal to the *gcd* of the initial values of these variables. The semantic term is semantically correct for the state $[M\mapsto 8, N\mapsto 8]$.

4. Partial and total correctness of the semantic term.

Theorem 1 (partial correctness of $STR^{M,N}$). Let a state st be such that $'M(st)=m$, $'N(st)=n$ (natural numbers $m, n>0$).

If $STR^{M,N}(st)\downarrow=str$, then $'M(str)='N(str)=gcd(m, n)$.

Proof (general schema without details). Induction on k : the number of loop iterations (denoted $\downarrow^k$).

Induction statement: $STR^{M,N}(st)\downarrow^k=str\Rightarrow 'M(str)='N(str)=gcd(m, n)$.

Base of induction (k=0). In this case $S_B^1(\neg, S_P^2(=,'M,'N))(st)=F$. From this follows that $m=n=gcd(m, n)$.

Step of induction (*for* k+1). Let $STR^{M,N}(st)\downarrow^{k+1}=str$. From this follows that $S_B^1(\neg, S_P^2(=,'M,'N))(st)=T$. Hence, $m\neq n$.

Let $IF(S_P^2(>,'M,'N), AS^M(S_F^2(-,'M,'N)), AS^N(S_F^2(-,'N,'M)))(st)=st'$.

Evaluating both cases of conditional operator we obtain:

$$AS^M(S_F^2(-,'M,'N))(st)=st\nabla[M\mapsto m-n]\ (m>n) \text{ and } AS^N(S_F^2(-,'N,'M))(st)=st\nabla[N\mapsto n-m]\ (n>m).$$

Since $gcd(m-n\ n)=gcd(m, n)$ for $m>n$ and $gcd(m-n, n)=gcd(m, n-m)$ for $n>m$ (recall the theory of natural numbers) we have that $gcd('M(st'),'N(st'))=gcd(m, n)$.

By inductive hypothesis $'M(str)='N(str)=gcd('M(st'),'N(st'))=gcd(m, n)$.

Termination of the program is based on the fact that $n+m$ is decreasing for each loop execution until these numbers become equal (details of this proof are omitted). So, the following statement is valid:

Theorem 2. $STR^{M,N}$ is totally correct.

5. Correctness proof of $STR^{M,N}$ with the help of Floyd-Hoare logic

First, two "logical" variables X and Y are introduced. Then we make "sugaring" of the term:

$M–N =_{def} S_P^2(–,'M,'N)$,

$N–M =_{def} S_P^2(–,'N,'M)$,

$M>N =_{def} S_P^2(>,'M,'N)$,

$\neg(M=N) =_{def} S_B^1(\neg, S_P^2(=,'M,'N))$.

Thus, instead of $WH(S_B^1(\neg, S_P^2(=,'M,'N)), IF(S_P^2(>,'M,'N), AS^M(S_F^2(–,'M,'N)), AS^N(S_F^2(–,'N,'M))))$ we get

$$WH(\neg(M=N), IF(M>N, AS^M(M–N), AS^N(N–M)))$$

Let

- $P =_{def} gcd(X,Y)=gcd(M,N)$,
- $P1 =_{def} S^{[M]}(P, M–N)$,
- $P2 =_{def} S^{[N]}(P, N–M)$.

Then by *Ax_AS* we have that $\vdash \{S^{[M]}(P, M–N)\}\ AS^M(M–N)\{P\}$.
By *Ax_CONS* we have $\vdash \{(M>N) \wedge S^{[M]}(P, M–N)\}\ AS^M(M–N)\{P\}$.
Here we need to prove $(M>N) \wedge S^{[M]}(P, M–N) \Rightarrow S^{[M]}(P, M–N)$. We use semantic reasoning.
Then again by *Ax_CONS* using $(M>N)\wedge P \Rightarrow (M>N)\wedge\{S^{[M]}(P, M–N)\}$ we obtain $\vdash \{(M>N) \wedge P\}\ AS^M(M–N)\{P\}$.
Similarly, we obtain $\vdash\{\neg(M>N) \wedge P\}\ AS^N(N–M)\{P\}$.

By *Ax_IF* we get $\vdash \{P\}\ IF(M>N, AS^M(M–N), AS^N(N–M))\ \{P\}$.

By *Ax_CONS* we get $\vdash \{(\neg(M=N)) \wedge P\}\ IF(M>N, AS^M(M–N), AS^N(N–M))\ \{P\}$.

By *Ax_WH* we get $\{P\}\ WH(\neg(M=N), IF(M>N, AS^M(M–N), AS^N(N–M)))\ \{P\wedge\neg(\neg(M=N))\}$.

Proving $(X=M\wedge Y=N) \Rightarrow P$ and $P\wedge\neg(\neg(M=N)) \Rightarrow (M=N) \wedge gcd(X,Y)=M$ and using *Ax_CONS* we get
$\vdash\{(X=M\wedge Y=N)\}\ WH(\neg(M=N), IF(M>N, AS^M(M–N), AS^N(N–M)))\ \{(M=N) \wedge gcd(X,Y)=M\}$.

The derivation is presented in the following table.

Table 9: Derivation tree for $STR^{M,N}$

$$\frac{\{S^{[M]}(P, M–N)\}\ AS^M(M–N)\{P\} \qquad \{S^{[N]}(P, N–M)\}\ AS^N(N–M)\{P\}}{\{(M>N) \wedge S^{[M]}(P, M–N)\}\ AS^M(M–N)\{P\} \qquad \{\neg(M>N) \wedge S^{[N]}(P, N–M)\}\ AS^N(N–M)\{P\}}\ Ax\_AS$$

$$\frac{\{(M>N) \wedge S^{[M]}(P, M–N)\}\ AS^M(M–N)\{P\} \qquad \{\neg(M>N) \wedge S^{[N]}(P, N–M)\}\ AS^N(N–M)\{P\}}{\{(M>N) \wedge P\}\ AS^M(M–N)\{P\} \qquad \{\neg(M>N) \wedge P\}\ AS^M(N–M)\{P\}}\ Ax\_CONS$$

$$\frac{\{(M>N) \wedge P\}\ AS^M(M–N)\{P\} \qquad \{\neg(M>N) \wedge P\}\ AS^M(N–M)\{P\}}{\{P\}\ IF(M>N, AS^M(M–N), AS^N(N–M))\ \{P\}}\ Ax\_IF$$

$$\frac{\{P\}\ IF(M>N, AS^M(M–N), AS^N(N–M))\ \{P\}}{\{(\neg(M=N)) \wedge P\}\ IF(M>N, AS^M(M–N), AS^N(N–M))\{P\}}\ Ax\_CONS$$

$$\frac{\{(\neg(M=N)) \wedge P\}\ IF(M>N, AS^M(M–N), AS^N(N–M))\{P\}}{\{P\}\ WH(\neg(M=N), IF(M>N, AS^M(M–N), AS^N(N–M)))\ \{P\wedge\neg(\neg(M=N))\}}\ Ax\_WH$$

$$\frac{\{P\}\ WH(\neg(M=N), IF(M>N, AS^M(M–N), AS^N(N–M)))\ \{P\wedge\neg(\neg(M=N))\}}{\{(X=M\wedge Y=N)\}\ WH(\neg(M=N), IF(M>N, AS^M(M–N),AS^N(N–M)))\ \{(M=N)\wedge gcd(X,Y)=M\}}\ Ax\_CONS$$

One more example is presented in the slides (part 3).