VERIFICA FORMALE DI PROGRAMMI (SEMANTICA ASSIOMATICA)

--> dimostrare che i programmi sono corretti! Nel senso matematico, diverso dal testing

WHILE PROGRAMS: sottoinsieme di programmi, sono semplici.
Assegnamento: *y:=t* //diverso dall'uguale di confronto...
Composizione: *s1, s2* // con s1 e s2 istruzioni
"if then else": IF e THEN s1 ELSE s2 FI (e è il predicato)
WHILE e DO s OD

Solo VARIABILI INTERE. Non ci sono oggetti, puntatori, stringhe, metodi...

{P} S {Q} --> se è vero P ed eseguo S, allora vale Q

S: while program, P: precondizione, Q: postcondizione

P, Q asserzioni

and logico, or, not, --> (implica ad esempio x=2 --> x>0) x=2 è + forte, x>0 è + debole

Esempio:
P: {TRUE}
S: m:=0
if x>y THEN m:=x ELSE m=y
Q: {m=max(x,y)}

Idea: spezzare la dimostrazione in sotto-dimostrazione

{P} S {Q} piccoli passi, ad ogni passo applico una regola (assiomi)

ASSEGNAMENTO: {P[y-->t]}y:=t {P} (da ricordare!!!!)
- all'indietro:
$\{0>=0_{(true)}\}\ x_{(y)}:=0_{(t)}\ \{x>=0_{(P)}\}$
$\{u+1>3_{(u>2)}\}\ x_{(y)}:=u+1_{(t)}\ \{x>3_{(P)}\}$
$\{x>=5_{(2x>=10)}\}\ y_{(y)}:=2*x_{(t)}\ \{y>=10_{(P)}\}$
$\{y=5\}\ y_{(y)}:=y+5_{(t)}\ \{y=10_{(P)}\}$
$\{6>10_{(false)}\}\ x_{(y)}:=6_{(t)}\ \{x>10_{(P)}\}$

- in avanti:
{P} y:=t {?} $P_{[y-->y']}$ and $t_{[y-->y']} = y$
Esempi:
{y>5} y:=2*(y+5) {y'>5 and y=2*(y'+5)} --> {y>20}

COMPOSIZIONE: {P} S1 {R} (A1) , {R} S2 {Q} (A2)
{P} S1 ; S2 {Q} (A)

se vuoi provare A, --> dimostra A1 e A2
se vale A1 e A2 allora vale A

Esempi:
1) Voglio dimostrare che:
{x=y+1} x:=x+1; y:=y+2 {x=y}

A2: {x=y+2} y:=y+2 {x=y} applico regola dell'assegnamento
A1 {x+1=y+2} x=x+1 {x=y+2} applico regola dell'assegnamento
{x=y+1}
A2 e A1 implica A per via della regola di composizione

2)Voglio dimostrare che:
x>2 {x>2} P
y:= 3*x+1, x:=y+3 {x>10}
{y>7} x:=y+3 {x>10} regola assegnamento A2
{3*x+1>7} y:= 3*x+1 {y>7} regola assegnamento A1
x>10 {x>10} Q

3)
{x>=0} x:=x+1 {x>=0}
⇧ +forte
{x>=-1} x:=x+1 {x>=0} --> Regola Assegnamento
R-->P {x>=0} --> {x>=-1} conseguenza

CONSEGUENZA: R-->P {P} S {Q}
{R} S {Q}
detto anche “rafforzamento pre-condizione”

{P} S {Q} Q-->R
{P} S {R}
detto anche “indebolimento post-condizione”

FALSE --> TRUE {TRUE} S {Q}
{FALSE} S {Q} --> questo è sempre vero!

IF THEN ELSE: {P and e} S1 {Q}, {P and not e} S2 {Q}
{P} IF e THEN S1 ELSE S2 FI {Q}

Esempi:
{y>1}(P) if(x>0)(e) then y:=y-1;(S1) else y:=y+1(S2) {y>0(Q)}

- {P and e} S1 {Q}
  1. {y>1} y:=y-1 {y>0} ASSEGNAMENTO
  2. {y>1 and x>0} --> {y>1} LOGICA
  3. {y>1 and x>0} y:=y-1 {y>0} RAFFORZAMENTO PRE 1 e 2
  4. {y>-1} y:=y+1 {y>0} ASSEGNAMENTO
  5. {y>1 and x<=0} --> {y>-1} LOGICA
  6. {y>1and x<=0} y:=y+1 {y>0} RAFFORZAMENTO PRE 4 e 5
  7. 3, 6 con la regola IF THEN ELSE

{TRUE} if x>0 m:=x else m:=-x fi {m>0}

1) {x>=0} m:=x {m>=0} assegnamento
2) {x>0} --> {x>=0} logica
3) {TRUE and x>0} --> {x>=0} logica
4) {TRUE and x>0} m:=x {m>=0} rafforzamento precondizione 1,3
5) {x<0} m:=-x {m>=0} assegnamento
6) {TRUE and not (x>0)} --> {x<=0} logica
7) {TRUE and not (x>0)} m:=x {m>=0} rafforzamento precondizione 5,6
8) {TRUE} if.... {m>=0} regola if 4,7