Trabajo práctico 1:

Especificación y WP

6 de mayo de 2024 Algoritmos y Estructuras de Datos

Grupo parenLosAlgoritmos

Integrante	LU	Correo electrónico
Ballerio, Francisco	986/23	`francisco.ballerio@hotmail.com`
Lopez, Gabriel	615/23	`gabriellopezdu@gmail.com`
Suárez, Francisco	104/23	`plottier2002@gmail.com`
Valesk, Benjamín	004/01	`email4@dominio.com`

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

1. Especificación

1.1. trayectoriaDeLosFrutosIndividualesALargoPlazo

`proc` `trayectoriaDeLosFrutosIndividualesALargoPlazo` (inout trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in cooperan:$seq\langle Bool\rangle$, in apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in eventos: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$)

> `requiere` $\{(trayectorias = trayectorias_0) \wedge \ |trayectorias| = |cooperan| = |apuestas| = |pagos| = |eventos| \wedge (\forall i : \mathbb{Z})\ (0 \leq i < |pagos| \longrightarrow_L (\forall k : \mathbb{Z})\ (0 \leq k < |eventos_{[i]}| \longrightarrow_L eventos_{[i][k]} > 0) \ \wedge$ $(\forall j : \mathbb{Z})\ (0 \leq j < |pagos_{[i]}| \longrightarrow_L |pagos_{[i]}| = |apuestas[i]| \wedge pagos_{[i][j]} > 0 \wedge apuestas_{[i][j]} > 0 \wedge trayectorias_{[i][0]} > 0))\wedge$ $sumatoriaApuestas(apuestas)\}$
>
> `asegura` $\{|tractorias| = |trayectorias_{[0]}| \ \wedge \ longFinal(trayectorias, eventos) \ \wedge$ $elPrimeroSeMantiene \ (trayectorias, trayectorias_0) \ \wedge$ $esTrayectoriaMod \ (trayectorias, apuestas, pagos, eventos, cooperan)\}$

`pred` `longFinal` (trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias| \longrightarrow_L trayectorias[i] = |eventos| + 1)$
}

`pred` `elPrimeroSeMantiene` (trayectorias:$seq\langle seq\langle\mathbb{R}\rangle\rangle$, trayectorias$_0$: $seq\langle\mathbb{R}\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |trayectorias| \longrightarrow_L (trayectoria[i][0] = trayectorias_0[i][0])$
}

`pred` `esTrayectoriaMod` (trayectorias: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, eventos $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, cooperan: $seq\langle Bool\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |pagos| \longrightarrow_L (\exists k : \mathbb{Z})\ (0 \leq k < |pagos[i]| \wedge_L$
$(\forall j : \mathbb{Z})\ (1 \leq j < |trayectorias[i]| \longrightarrow_L$
$trayectorias[i][j] = decideGanancia(cooperan[i], fondoComúnDividido(cooperan, loGanado(trayectorias[i][j-1],$
$tasa(apuestas[i][k], pagos[i][k])), loGanado(trayectorias[i][j-1], tasa(apuestas[i][k], pagos[i][k])))))$
}

`aux` `decideGanancia` (in cooperan: Bool,$\in fondoComúnDiv : \mathbb{R}, loGanado : \mathbb{R})\ : \mathbb{R}\ =$
if $coopera = true$ then $fondoComúnDiv$ else $(loGanado + fondCúnDiv)$ fi ;

`aux` `fondoComúnDiv` (in cooperan: $seq\langle Bool\rangle$, in contribución : $\mathbb{R}$) : $\mathbb{R}\ =$
$\sum_{j=0}^{|cooperan|-1}$(if $cooperan[j] = true$ then $contribución$ else 0 fi) / $|cooperan|$;

`aux` `tasa` (in apuesta: $\mathbb{R}$,pago: $\mathbb{R}$) : $\mathbb{R}\ = apuesta * pago$;

`aux` `loGanado` (in recurso: $\mathbb{R}$, n: $\mathbb{R}$) : $\mathbb{R}\ = recurso * n$;

`pred` `sumatoriaApuestas` (apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |apuestas| \longrightarrow_L (\sum_{k=0}^{|apuestas[i]|-1} apuestas[i][k]) = 1)$
}

1.2. individuoActualizaApuesta

`proc` `individuoActualizaApuesta` (in individuo : $\mathbb{Z}$,in recursos $seq\langle\mathbb{R}\rangle$,in cooperan:$seq\langle Bool\rangle$, inout apuestas: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in pagos: $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in eventos: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$)

> `requiere` $\{sumatoriaApuestas(apuestas) \wedge (apuestas = apuestas_0) \wedge \ |recursos| = |cooperan| = |apuestas| = |pagos| = |eventos| \wedge (\forall i : \mathbb{Z})\ (0 \leq i < |pagos| \longrightarrow_L (\forall k : \mathbb{Z})\ (0 \leq k < |eventos_{[i]}| \longrightarrow_L eventos_{[i][k]} > 0) \ \wedge$ $(\forall j : \mathbb{Z})\ (0 \leq j < |pagos_{[i]}| \longrightarrow_L |pagos_{[i]}| = |apuestas[i]| \wedge pagos_{[i][j]} > 0 \wedge apuestas_{[i][j]} > 0 \wedge recursos_{[i]} > 0))\}$
>
> `asegura` $\{|apuestas| = |apuestas_{[0]}| \ \wedge \ soloCambiaIndividuo(individuo, apuestas, apuestas_0) \ \wedge$
> $(\forall trayCom : seq\langle seq\langle\mathbb{R}\rangle\rangle)\ (esTrayectoriaMod(trayCom, apuestas, pagos, eventos, cooperan) \ \wedge$
> $recursoInicial(trayCom, recursos) \ \wedge \ longFinal(trayCom, eventos) \longrightarrow$
> $(\exists trayMax, apuestasMax : seq\langle seq\langle\mathbb{R}\rangle\rangle)\ (sumatoriaApuestas(apuestasMax) \wedge$
> $esTrayectoriaMod(trayMax, apuestasMax, pagos, eventos, cooperan) \ \wedge$
> $recursoInicial(trayMax, recursos) \ \wedge longFinal(trayMax, eventos) \wedge_L$
> $(trayMax[individuo][|trayMax|-1] \geq trayCom[individuo][|trayCom|-1]) \longrightarrow$
> $apuestas[indivduo] = apuestasMax[individuo]))\}$

`pred` `recursoInicial` (in trayectoria : $seq\langle seq\langle\mathbb{R}\rangle\rangle$, in recursos : $seq\langle\mathbb{R}\rangle$) {

$(\forall i : \mathbb{Z})\ (0 \leq i < |trayectoria| \longrightarrow_L (trayectoria[i][0] = recursos[i]))$
}
`pred soloCambiaIndiviuo` (in apuestas: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, in $apuestas_0$: $seq\langle seq\langle \mathbb{R}\rangle\rangle$, in inividuo: $\mathbb{Z}$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |apuestas| \longrightarrow ((i \neq inividuo \ \wedge \ apuestas[i] = apuestas_0 i)$
}

2. Demostraciones de correctitud

En este punto del trabajo vamos a probar que la especificación de la función frutoDelTrabajoPuramenteIndividual es correcta respecto de su implementación.
Probamos la correctitud del programa de la siguiente manera:

S1 $\equiv res = recurso$
$S2 \equiv i = 0$
$S3 \equiv$ while ($i < |eventos|$) do S4,S5
endwhile
$S4 \equiv$ (if $eventos[i]$ then $S6$ else $S7$ fi)
$S5 \equiv$ i = i + 1
$S6 \equiv res = (res * apuesta.c) * pago.c$
$S7 \equiv res = (res * apuesta.s) * pago.s$
$Q \equiv res = recurso * (apuesta.c * pago.c)^{apariciones(eventos,T)} * (apuesta.s * pago.s)^{apariciones(eventos,t)}$

wp(S1, S2, S3, Q) $\equiv_{axioma3} wp(S1, wp(S2, wp(S3, Q)))$
$wp(S3, Q) \equiv_{axioma5}$ Por este axioma sabemos que no se puede hacer wp de un ciclo, pues quedamos encerrados en un bucle infinito.
Por eso usamos el teorema de la invariante para probar la correctitud del ciclo y que este termina.
Entonces decimos que si existe un predicado I que cumple con:

1 Pc $\longrightarrow$ I (Precondición del ciclo implica a la invariante)

2 $I \wedge B\{S\}I$ (Durante cualquier momento del ciclo la invariante sigue valiendo)

3 $I \wedge \neg B \longrightarrow Q_c$ (Se cumple la postcondición al salir del ciclo

4 $(I \wedge B \wedge V_0 = f_v)\{S\}(f_v < V_0)$ (fv es estrictamente decreciente)

5 $(I \wedge fv \leq 0) \longrightarrow \neg B$ (Si fv alcanza la cota inferior, la guarda (B) no se cumple)

Los puntos 1,2 y 3 demuestran la correctitud del ciclo. Mientras que los puntos 4 y 5 demuestran, mediante una funcion variante, que el ciclo termina.
Ahora definimos:
$Pc \equiv (res = recurso \wedge i = 0)$
$Qc \equiv Q \equiv res = recurso((apuesta_c * pago_c)^{\#(eventos),t)} * (apuestas_s * pago_s)^{\#(eventos,f)})$
$B \equiv (i <$ |eventos|$)$
$C \equiv eventos_{[i]}$
$I \equiv (0 \leq i \leq |eventos| \wedge$
$res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)})$
$fv \equiv |eventos| - i$

1 Pc $\longrightarrow$ I

$res = recurso \wedge i = 0 \wedge apuesta_c + apuesta_s = 1 \wedge paco_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0)$
Todo esto es PC. Voy a asignarlo a **PC** para facilitar la lectura
PC$\longrightarrow res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)})$
Por $i = 0$
PC$\longrightarrow res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,0),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,0),f)})$
Como $subseq(lista, 0, 0) = subseq(\{\})$
PC$\longrightarrow res = recurso((apuesta_c * pago_c)^{\#(subseq(\{\}),t)} * (apuestas_s * pago_s)^{\#(subseq(\{\}),f)})$
PC$\longrightarrow res = recurso((apuesta_c * pago_c)^0 * (apuestas_s * pago_s)^0)$
PC$\longrightarrow res = recurso(((apuesta_c)^0 * (pago_c)^0) * ((apuestas_s)^0 * (pago_s)^0))$
Desarmo **PC** para que se vea claramente

$res = recurso \wedge i = 0 \wedge apuesta_c + apuesta_s = 1 \wedge paco_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0) \longrightarrow res = recurso((1)(1) * (1)(1))$

$res = recurso \wedge i = 0 \wedge apuesta_c + apuesta_s = 1 \wedge paco_c > 0 \wedge pago_s > 0 \wedge apuesta_c > 0 \wedge apuesta_s > 0 \wedge recurso > 0) \longrightarrow res = recurso$

Luego, es cierto que $Pc \longrightarrow$ I

2 $I \wedge B \ \{S\} \ I$

$Calculamos\ wp(S3, I)\ para\ probar\ (I \wedge B) \longrightarrow wp(S3, I)$

$wp(S3, I) \equiv^{(por\ axioma3)} wp(s5, wp(s4, I))$

Vamos por partes, primero calculamos $wp(s4, I) \equiv^{\text{por axioma 4}} def(C) \wedge_L ((C \wedge wp(S6, I)) \vee ((\neg C \wedge wp(S7, I))$

$WP(S6, I) \equiv def(res * apuestas_c * pago_c) \wedge_L I^{res}_{res * apuestas_c * pago_c}$

$WP(S6, I) \equiv (0 \leq i \leq |eventos| \wedge res * apuestas_c * pago_c = recursos * ((apuestas_c * pago_c)^{\#(subseq(eventos,o,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,o,i),F)}))$

$(C \wedge WP(S6, I) \equiv (True \wedge (0 \leq i \leq |eventos| \wedge res*apuestas_c*pago_c = recursos*((apuestas_c*pago_c)^{\#(subseq(eventos,o,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,o,i),F)}))$

$WP(S7, I) \equiv def(res * apuestas_s * pago_s) \wedge_L I^{res}_{res * apuestas_s * pago_s}$

$WP(S7, I) \equiv (0 \leq i \leq |eventos| \wedge res * apuestas_s * pago_s = recursos * ((apuestas_c * pago_c)^{\#(subseq(eventos,o,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,o,i),f)}))$

$(\neg C \wedge WP(S7, I) \equiv (False \wedge (0 \leq i \leq |eventos| \wedge res*apuestas_s*pago_s = recursos*((apuestas_c*pago_c)^{\#(subseq(eventos,o,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,o,i),F)})) \equiv False$

$Luego,\ no\ seguiremos\ esta\ rama\ ya\ que\ False\ es\ la\ precondicion\ mas\ restrictiva\ y\ no\ nos\ servira\ para\ calcular\ la\ wp\ genera$

$Para\ simplificar\ la\ escritura\ llamaremos\ E_1\ a\ (C \wedge wp(S6, I))$

$wp(S5, E1) \equiv^{por\ axioma\ 1} def(i + 1) \wedge_L E1^{i}_{i+1}$

$wp(S5, E1) \equiv (True \wedge (0 \leq i{+}1 \leq |eventos| \wedge res*apuestas_c*pago_c = recursos*((apuestas_c*pago_c)^{\#(subseq(eventos,o,i+1),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,o,i+1),F)}))$

$Finalmente\ ,como\ tenemos\ (I \wedge B)\ sabemos\ que\ i < |eventos| \wedge (0 \leq i \leq |eventos|) \longrightarrow (0 \leq i < |eventos|\ separamos$

$las\ implicaciones:$

$(0 \leq i < |eventos|) \longrightarrow (0 \leq i + 1 \leq |eventos|)$ Luego , esto es verdadero

$res = recursos * ((apuestas_c * pago_c)^{\#(subseq(eventos,o,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,o,i),F)}) \longrightarrow$

$res*apuestas_c*pago_c = recursos*((apuestas_c*pago_c)^{\#(subseq(eventos,o,i+1),t)} * (apuestas_s*pago_s)^{\#(subseq(eventos,o,i+1),F)})$
Luego, buscamos que la igualdad sea cierta en el consecuente para que evalue True y quede una tautologia.

Notamos que para que esto pase, #(subseq(eventos,0,i+1),t) debe evaluar a 1 y #(subseq(eventos,0,i+1),f) debe evaluar a 0 de este modo, (apuestas$_c * pago_c)^{\#(subseq(eventos,o,i+1),t)} = (apuestas_c * pago_c) y (apuestas_s * pago_s)^{\#(subseq(eventos,o,i+1),f)} = 1$.

$Luego, paraque \#(subseq(eventos, 0, i{+}1), f) = 0 y \#(subseq(eventos, 0, i{+}1), t) = 1, eventostendraquesertalque : eventos = [True]$.

Finalmete WP(S3,I) $\equiv$ $(0 \leq i + 1 \leq |eventos| \wedge eventos = [True])$ y por lo antes explicado esto es una tautologia y demuestra la correctitud de este paso.

3 $I \wedge \neg B \longrightarrow Q_c$

$\neg B \longrightarrow \neg(i < |eventos|) \longrightarrow (i \geq |eventos|)$; entonces, usando que $\wedge$ es conmutativa:

$(i \geq |eventos|) \wedge 0 \wedge \leq i \leq |eventos| \wedge$
$res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \longrightarrow Q_c$

$\equiv i = |eventos| \wedge$

$res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)} \longrightarrow Q_c$

$\equiv res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,|eventos|),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,|eventos|),f)}) \ \wedge$
$(i \geq |eventos|) \longrightarrow Q_c$

Pero la subsecuencia de eventos que va desde el 0 hasta la longitud de eventos $((subseq(eventos, 0, |eventos|)$ es, en realidad, la secuencia eventos original, entonces queda:

$\equiv res = recurso((apuesta_c * pago_c)^{\#(eventos),t)} * (apuestas_s * pago_s)^{\#(eventos,f)})$
$\longrightarrow res = recurso((apuesta_c * pago_c)^{\#(eventos),t)} * (apuestas_s * pago_s)^{\#(eventos,f)})$

Así, queda probado que I $\wedge \neg B \longrightarrow Q_c$

4 $((I \wedge B) \wedge (V_0 = f_v))\{S\}(f_v < V_0)\ fv \equiv |eventos| - i$

$S \equiv$ if $elementos[i]$ then
$res = (res * apuestas_c) * pago_c$
else
$res = (res * apuestas_s) * pago_s$
endif
$i = i + 1$

Para probar este punto, hago la wp entre {S} y $(f_v < V_0)$.

$WP(S, F_v < V_0)$
$\equiv WP($if $eventos[i]$ then $res = (res*apuestas_s)*pago_s$ else $res = (res*apuestas_c)*Pago_c$ fi; $\ i = i+1, (|eventos|-i) < V_0$
Por axioma 3;
$\equiv WP($if $eventos[i]$ then $res = (res*apuestas_s)*pago_s$ else $res = (res*apuestas_c)*Pago_c$ fi$, WP(i = i+1, |eventos|-i < V_0)$

Por un lado, hago $WP(i = i + 1, |eventos| - i < V_0)$

$WP(i = i + 1, |eventos| - i < V_0)$ por axioma 1;
$\equiv def(i + 1) \wedge_L (|eventos| - (i + 1) < V_0)$
$\equiv |eventos| - i - 1 < V_0$
$\equiv |eventos| - i \leq V_0$
Ahora vuelvo a la WP original.

$WP($if $eventos[i]$ then $res = ((res * apuestas_s) * pago_s$ else $res = (res * apuestas_c) * Pago_c$ fi, $\ |eventos - i \leq V_0)$
Por Axioma 4;
$\equiv def(eventos[i]) \wedge_L (eventos[i] \ \wedge \ WP((res = (rs * apuestas_c) * pago_c), |eventos| - i > V_0)$
$\vee (\neg(eventos[i] \ \wedge \ WP((res = (res * apuestas_s) * pagos_s), |eventos| - i > V_0)$

Como $WP((res = (rs*apuestas_c)*pago_c), |eventos|-i > V_0)$ no tiene nada en común entre {S} y Q, entonces la ejecución del programa (en este caso, if $eventos[i]$ then $res = (res * apuestas_s) * pago_s$ else $res = (res * apuestas_c) * Pago_c$ fi) no se relaciona con la postcondición. Es decir, se podría interpretar a {S} como skip. Lo mismo ocurre con $WP((res = (res * apuestas_s) * pagos_s), |eventos| - i > V_0)$ Así;

$\equiv 0 \leq i < |eventos| \wedge_L (eventos[i] \ \wedge \ WP(skip, |eventos| - i \leq V_0) \ \vee \ (\neg(eventos[i] \wedge WP(skip, |eventos| - i \leq V_0)$
$\equiv 0 \leq i < |eventos| \wedge_L (eventos[i] \ \wedge \ |eventos| - i \leq V_0) \ \vee \ (\neg(eventos[i] \wedge |eventos| - i \leq V_0)$
$\equiv 0 \leq i < |eventos| \wedge_L |eventos| - i \leq V_0$

Ahora, tomamos $(I \wedge B) \ \wedge \ (V_0 = f_v)$:

$(0 \leq i \leq |eventos| \wedge$
$res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \ \wedge \ i < |eventos| \ \wedge \ V_0 =$
$|eventos| - i$

Y se puede ver que la implica, por lo que la wp entre {S} y $(f_v < V_0)$ demuestra que fv es estrictamente decreciente en el cuerpo del ciclo.

5 $(I \wedge fv \leq 0) \longrightarrow \neg B$

$(0 \leq i \leq |eventos| \wedge res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \ \wedge$
$|eventos| - i \leq 0) \longrightarrow (\neg(i < |eventos|)$

$\equiv (0 \leq i \leq |eventos| \wedge res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \ \wedge$

$(|eventos| \leq i)) \longrightarrow (i \leq |eventos|)$

$\equiv (res = recurso((apuesta_c * pago_c)^{\#(subseq(eventos,0,i),t)} * (apuestas_s * pago_s)^{\#(subseq(eventos,0,i),f)}) \ \wedge$
$(|eventos| = i)) \longrightarrow (i \leq |eventos|)$

Se puede ver en la última implicación es verdadera, demostrando así que al llegar fv a la cota inferior, la guarda deja de cumplirse.

Queda así demostrada la correctitud y la finitud del ciclo. Como el programa termina junto con el ciclo, queda también demostrada la correctitud la especificación del programa respecto a su implementación.