Verificación de software

Trabajo Práctico Final - Ingeniería de Software II

Autor:

Juan Ignacio Farizano

Departamento de Ciencias de la Computación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
Rosario, Santa Fe, Argentina

18 de febrero de 2025

1. Enunciado del problema

Queremos especificar el funcionamiento del sistema *Mis Estudiantes* que utilizan directivos de escuelas primarias y secundarias para inscribir y reinscribir alumnos (por ej. promoción de grado, repitente, etc) en la base de datos del Ministerio de Educación de la provincia de Buenos Aires, lo descrito es similar al sistema existente pero simplificado para mantener una dificultad razonable para este trabajo práctico.

Un directivo desea inscribir o reinscribir a un alumno en su escuela. Cada alumno puede encontrarse en una de las siguientes situaciones:

- **Inscripto**: el alumno fue inscripto y es habilitado a cursar el grado registrado
- **Promovido**: el alumno cumplió los requisitos para promocionar el grado al cual se encuentra inscripto y es habilitado a ser inscripto en el grado siguiente o a graduarse si se encontraba en el 12avo grado
- **Repitente**: el alumno no cumplió los requisitos para promocionar de grado y debe ser inscripto en el mismo grado

Se deben especificar las siguientes operaciones:

1. **Inscribir alumno**: se inscribe por primera vez a un alumno a primer grado
2. **Reinscribir alumno**: se inscribe un alumno previamente registrado al grado que corresponda según su último estado de inscripción
3. **Cerrar inscripción**: la inscripción actual es cerrada y se registra una promoción o repitencia
4. **Consultar graduado**: se desea consultar si un alumno se graduó

Para simplificar no diferenciamos entre primaria y secundaria, registramos desde 1er grado hasta 12avo, el último del secundario. Los requisitos para promocionar de grado se encuentran por fuera del sistema y no deben ser tenidos en cuenta.

2. Designaciones

alumno es un alumno $\approx$ *alumno* $\in ALUMNO$

grado es un grado correspondiente a un año escolar $\approx$ *grado* $\in GRADO$

estado es un estado de inscripción $\approx$ $estado \in ESTADO$

rep es un mensaje de respuesta $\approx$ $rep \in REP$

Estado de inscripción actual correspondiente al alumno *alumno* $\approx$ $inscripciones(alumno)$

3. Especificación en Z

$[ALUMNO]$

$GRADO == \mathbb{N}$

$ESTADO ::= inscripto \mid promueve \mid repite$

$REP ::= alumnoEsGraduado \mid alumnoNoEsGraduado \mid alumnoNoEncontrado$

MisEstudiantes

$registrados : \mathbb{P}\, ALUMNO$

$inscripciones : ALUMNO \pfun (GRADO \times ESTADO)$

InscripcionesInv

MisEstudiantes

$\mathrm{dom}\, inscripciones = registrados$

MaximoGradoInv

MisEstudiantes

$\forall\, alumno : registrados \bullet (inscripciones\ alumno).1 \leq 12$

MisEstudiantesInicial

MisEstudiantes

$registrados = \varnothing$

$inscripciones = \varnothing$

InscribirAlumnoOk

$\Delta MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \notin registrados$

$inscripciones' = inscripciones \cup \{alumno? \mapsto (1, inscripto)\}$

$registrados' = registrados \cup \{alumno?\}$

InscribirAlumnoRegistradoE

$\Xi MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \in registrados$

$InscribirAlumno == InscribirAlumnoOk \lor InscribirAlumnoRegistradoE$

ReinscribirAlumnoPromovidoOk

$\Delta MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \in registrados$

$(inscripciones\ alumno?).1 < 12$

$(inscripciones\ alumno?).2 = promueve$

$inscripciones' = inscripciones \oplus \{alumno? \mapsto ((inscripciones\ alumno?).1 + 1, inscripto)\}$

$registrados' = registrados$

ReinscribirAlumnoRepitenteOk

$\Delta MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \in registrados$

$(inscripciones\ alumno?).1 \leq 12$

$(inscripciones\ alumno?).2 = repite$

$inscripciones' = inscripciones \oplus \{alumno? \mapsto ((inscripciones\ alumno?).1, inscripto)\}$

$registrados' = registrados$

ReinscribirAlumnoNoEncontradoE

$\Xi MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \notin registrados$

ReinscribirAlumnoGraduadoE

$\Xi MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \in registrados$

$((inscripciones\ alumno?)).1 = 12$

$((inscripciones\ alumno?)).2 = promueve$

ReinscribirAlumnoDobleInscripE

$\Xi MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \in registrados$

$(inscripciones\ alumno?).2 = inscripto$

$ReinscribirAlumnoE == ReinscribirAlumnoGraduadoE \lor ReinscribirAlumnoDobleInscripE \lor ReinscribirAlumnoNoEncontradoE$

$ReinscribirAlumnoOk == ReinscribirAlumnoPromovidoOk \lor ReinscribirAlumnoRepitenteOk$

$ReinscribirAlumno == ReinscribirAlumnoOk \lor ReinscribirAlumnoE$

CerrarInscripcionOk

$\Delta MisEstudiantes$

$alumno? : ALUMNO$

$estado? : ESTADO$

$alumno? \in registrados$

$estado? \neq inscripto$

$inscripciones' = inscripciones \oplus \{alumno? \mapsto ((inscripciones\ alumno?).1, estado?)\}$

$registrados' = registrados$

CerrarInscripcionEstadoInvalidoE

$\Xi MisEstudiantes$

$estado? : ESTADO$

$estado? = inscripto$

CerrarInscripcionAlumnoNoEncontradoE

$\Xi MisEstudiantes$

$alumno? : ALUMNO$

$alumno? \notin registrados$

$CerrarInscripcionE == CerrarInscripcionEstadoInvalidoE \lor CerrarInscripcionAlumnoNoEncontradoE$

$CerrarInscripcion == CerrarInscripcionOk \lor CerrarInscripcionE$

AlumnoEsGraduadoSiOk

$\Xi MisEstudiantes$

$alumno? : ALUMNO$

$rep! : REP$

$alumno? \in registrados$

$(inscripciones\ alumno?).1 = 12$

$(inscripciones\ alumno?).2 = promueve$

$rep! = alumnoEsGraduado$

AlumnoEsGraduadoNoOk

$\Xi MisEstudiantes$

$alumno? : ALUMNO$

$rep! : REP$

$alumno? \in registrados$

$(inscripciones\ alumno?).1 \neq 12 \lor (inscripciones\ alumno?).2 \neq promueve$

$rep! = alumnoNoEsGraduado$

AlumnoEsGraduadoNoEncontradoE

$\Xi MisEstudiantes$

$alumno? : ALUMNO$

$rep! : REP$

$alumno? \notin registrados$

$rep! = alumnoNoEncontrado$

$AlumnoEsGraduadoOk == AlumnoEsGraduadoSiOk \lor AlumnoEsGraduadoNoOk$

$AlumnoEsGraduado == AlumnoEsGraduadoOk \lor AlumnoEsGraduadoNoEncontradoE$

4. {log}

La traducción de la especificación Z a un programa *{log}* se puede encontrar en el archivo **misestudiantes.pl**, para ejecutar este programa no es necesario consultar ninguna librería.

Simulaciones

Para realizar las siguientes simulaciones se cargó el programa con el type checker activado y una vez verificado que el programa tipa correctamente se desactivó para simplificar la ejecución de las simulaciones.

Primera simulación

```
misEstudiantesInicial(R0, I0) &
inscribirAlumno(R0, I0, pepe, R1, I1) &
cerrarInscripcion(R1, I1, pepe, promueve, R2, I2) &
reinscribirAlumno(R2, I2, pepe, R3, I3) &
cerrarInscripcion(R3, I3, pepe, repite, R4, I4) &
cerrarInscripcion(R4, I4, pepe, promueve, R5, I5) &
inscribirAlumno(R5, I5, pepe, R6, I6) &
alumnoEsGraduado(R6, I6, pepe, Rep1, R6, I6) &
alumnoEsGraduado(R6, I6, pepito, Rep2, R6, I6).
```

El objetivo de esta simulación es probar los usos habituales del programa, comenzando desde el estado inicial se inscribe un alumno *Pepe* se da una serie de inscripciones cerradas y reinscripciones, incluso una ocasión donde se cerró una inscripción como repitente y luego se modificó a promoción (por ejemplo, el alumno puede haber recuperado materias luego del cierre o se corrigió una equivocación al cerrar por primera vez). Además se consulta si *Pepe* está graduado después de promover segundo grado, lo cual no sería posible, y por último se consulta por un alumno que no se encuentra registrado.

```
R0 = {},
I0 = {},
R1 = {pepe},
I1 = {[pepe,[1,inscripto]]},
R2 = {pepe},
I2 = {[pepe,[1,promueve]]},
R3 = {pepe},
```

```
I3 = {[pepe,[2,inscripto]]},
R4 = {pepe},
I4 = {[pepe,[2,repite]]},
R5 = {pepe},
I5 = {[pepe,[2,promueve]]},
R6 = {pepe},
I6 = {[pepe,[2,promueve]]},
Rep1 = alumnoNoEsGraduado,
Rep2 = alumnoNoEncontrado
```

Los resultados cumplen lo esperado.

Segunda simulación

```
misEstudiantesInicial(R0, I0) &
inscribirAlumno(R0, I0, pepe, R1, I1) &
cerrarInscripcion(R1, I1, pepe, promueve, R2, I2) &
reinscribirAlumno(R2, I2, pepe, R3, I3) &
cerrarInscripcion(R3, I3, pepe, promueve, R4, I4) &
reinscribirAlumno(R4, I4, pepe, R5, I5) &
cerrarInscripcion(R5, I5, pepe, promueve, R6, I6) &
reinscribirAlumno(R6, I6, pepe, R7, I7) &
cerrarInscripcion(R7, I7, pepe, promueve, R8, I8) &
reinscribirAlumno(R8, I8, pepe, R9, I9) &
cerrarInscripcion(R9, I9, pepe, promueve, R10, I10) &
reinscribirAlumno(R10, I10, pepe, R11, I11) &
cerrarInscripcion(R11, I11, pepe, promueve, R12, I12) &
reinscribirAlumno(R12, I12, pepe, R13, I13) &
cerrarInscripcion(R13, I13, pepe, promueve, R14, I14) &
reinscribirAlumno(R14, I14, pepe, R15, I15) &
cerrarInscripcion(R15, I15, pepe, promueve, R16, I16) &
reinscribirAlumno(R16, I16, pepe, R17, I17) &
cerrarInscripcion(R17, I17, pepe, promueve, R18, I18) &
reinscribirAlumno(R18, I18, pepe, R19, I19) &
cerrarInscripcion(R19, I19, pepe, promueve, R20, I20) &
reinscribirAlumno(R20, I20, pepe, R21, I21) &
```

```
cerrarInscripcion(R21, I21, pepe, promueve, R22, I22) &
reinscribirAlumno(R22, I22, pepe, R23, I23) &
alumnoEsGraduado(R23, I23, pepe, Rep1, R23, I23) &
cerrarInscripcion(R23, I23, pepe, promueve, R24, I24) &
alumnoEsGraduado(R24, I24, pepe, Rep2, R24, I24) &
reinscribirAlumno(R24, I24, pepe, R25, I25) &
alumnoEsGraduado(R25, I25, pepe, Rep3, R25, I25) &
cerrarInscripcion(R25, I25, pepe, repite, R26, I26) &
alumnoEsGraduado(R26, I26, pepe, Rep4, R26, I26).
```

Esta segunda simulación es más extensa pero consiste en inscribir al alumno *Pepe* partiendo del estado inicial y realizar el ciclo cerrar inscripción/reinscribir hasta que se encuentre cursando el último grado, una vez aquí se consulta si está graduado a lo que se debería obtener una respuesta negativa, se cierra la inscripción para graduarlo, se consulta de nuevo y el resultado esta vez debería ser positivo. Luego, se intenta reinscribir al alumno graduado pero esto no es permitido, por lo que consultando de nuevo se debería ver que mantiene su estado de graduado, el cual solo es perdido cuando se decide modificar el estado del último cierre de inscripción por una repitencia, y consultado por última vez, el resultado sería negativo.

```
R0 = {},
I0 = {},
R1 = {pepe},
I1 = {[pepe,[1,inscripto]]},
R2 = {pepe},
I2 = {[pepe,[1,promueve]]},
.
.
.
I22 = {[pepe,[11,promueve]]},
R23 = {pepe},
I23 = {[pepe,[12,inscripto]]},
Rep1 = alumnoNoEsGraduado,
R24 = {pepe},
I24 = {[pepe,[12,promueve]]},
Rep2 = alumnoEsGraduado,
R25 = {pepe},
```

```
I25 = {[pepe,[12,promueve]]},
Rep3 = alumnoEsGraduado,
R26 = {pepe},
I26 = {[pepe,[12,repite]]},
Rep4 = alumnoNoEsGraduado
```

Los resultados fueron acortados ya que contaban con muchos estados intermedios similares. Estos resultados cumplen con lo esperados.

Demostración automática de propiedades

Invocando el generador de condiciones de verificación (VCG) con el comando *vcg('misestudiantes.pl').* es generado el archivo **misestudiantes-vc.pl**, consultando este archivo y ejecutando el comando *check_vcs_misestudiantes.* es posible descargar las condiciones de verificación.

En un principio todas las condiciones son verificadas correctamente excepto por la condición **inscribirAlumno_pi_inscripcionesPfunInv**, por lo cual fue necesario agregar la hipótesis **inscripcionesInv(Registrados, Inscripciones)** y luego esta condición fue descargada exitosamente. No fue necesario utilizar el comando *findh.*

5. Demostración de preservación de invariantes en Z/Eves

Utilizando la interfaz gráfica de *Z/Eves* se definió y demostró el teorema **InscribirAlumnoPI** el cual verifica que la operación **InscribirAlumno** preserva el invariante de estado **InscripcionesInv**.

theorem InscribirAlumnoPI

$InscripcionesInv \land InscribirAlumno \Rightarrow InscripcionesInv'$

proof[*InscribirAlumnoPI*]

invoke InscribirAlumno;

split InscribirAlumnoOk;

simplify;

cases;

invoke InscribirAlumnoOk;

invoke InscripcionesInv;

equality substitute;

reduce;

next;

invoke InscribirAlumnoRegistradoE;

invoke Ξ *MisEstudiantes*;

reduce;

next;

■

6. Generación de casos de prueba con Fastest

Utilizando Fastest se generarán casos de prueba para la operación *ReinscribirAlummno*, los comandos ejecutados para generar los casos de prueba fueron los siguientes:

```
loadspec misestudiantes.tex
selop ReinscribirAlumno
genalltt
addtactic ReinscribirAlumno_DNF_1 SP \oplus inscripciones \oplus
  \{ alumno? \mapsto ( ( inscripciones(alumno?) ).1 + 1 , inscripto ) \}
addtactic ReinscribirAlumno_DNF_2 SP \oplus inscripciones \oplus
  \{ alumno? \mapsto ( ( inscripciones(alumno?) ).1 , inscripto ) \}
addtactic ReinscribirAlumno_DNF_3 SP \in alumno? \in registrados
addtactic ReinscribirAlumno_DNF_4 SP \in alumno? \in registrados
addtactic ReinscribirAlumno_DNF_5 SP \notin alumno? \notin registrados
genalltt
genalltca
```

```
ReinscribirAlumno_VIS
  !______ReinscribirAlumno_DNF_1
  |  !______ReinscribirAlumno_SP_15
  |  !______ReinscribirAlumno_SP_16
  |  !______ReinscribirAlumno_SP_17
  |  !______ReinscribirAlumno_SP_18
  |  !______ReinscribirAlumno_SP_19
  |  !______ReinscribirAlumno_SP_20
  |  !______ReinscribirAlumno_SP_21
  |  !______ReinscribirAlumno_SP_22
  |
  !______ReinscribirAlumno_DNF_2
  |  !______ReinscribirAlumno_SP_7
  |  !______ReinscribirAlumno_SP_8
  |  !______ReinscribirAlumno_SP_9
  |  !______ReinscribirAlumno_SP_10
  |  !______ReinscribirAlumno_SP_11
  |  !______ReinscribirAlumno_SP_12
  |  !______ReinscribirAlumno_SP_13
  |  !______ReinscribirAlumno_SP_14
  |
  !______ReinscribirAlumno_DNF_3
  |  !______ReinscribirAlumno_SP_5
  |  !______ReinscribirAlumno_SP_6
  |
  !______ReinscribirAlumno_DNF_4
  |  !______ReinscribirAlumno_SP_3
  |  !______ReinscribirAlumno_SP_4
  |
  !______ReinscribirAlumno_DNF_5
    !______ReinscribirAlumno_SP_1
    !______ReinscribirAlumno_SP_2
```

asdasdadasdsad

```
ReinscribirAlumno_VIS
```

```
!______ReinscribirAlumno_DNF_1
|  !______ReinscribirAlumno_SP_18
|  |  !______ReinscribirAlumno_SP_18_TCASE
|  |
|  !______ReinscribirAlumno_SP_19
|    !______ReinscribirAlumno_SP_19_TCASE
|
|
!______ReinscribirAlumno_DNF_2
|  !______ReinscribirAlumno_SP_10
|  |  !______ReinscribirAlumno_SP_10_TCASE
|  |
|  !______ReinscribirAlumno_SP_11
|    !______ReinscribirAlumno_SP_11_TCASE
|
|
!______ReinscribirAlumno_DNF_3
|  !______ReinscribirAlumno_SP_5
|  |  !______ReinscribirAlumno_SP_5_TCASE
|  |
|  !______ReinscribirAlumno_SP_6
|    !______ReinscribirAlumno_SP_6_TCASE
|
|
!______ReinscribirAlumno_DNF_4
|  !______ReinscribirAlumno_SP_3
|  |  !______ReinscribirAlumno_SP_3_TCASE
|  |
|  !______ReinscribirAlumno_SP_4
|    !______ReinscribirAlumno_SP_4_TCASE
|
|
!______ReinscribirAlumno_DNF_5
  !______ReinscribirAlumno_SP_1
  |  !______ReinscribirAlumno_SP_1_TCASE
  |
```

```
!______ReinscribirAlumno_SP_2
  !______ReinscribirAlumno_SP_2_TCASE
```

Acá explicar por qué no generó para algunas hojas del árbol de prueba

Esquemas de casos de pruebas generados

ReinscribirAlumno_SP_18_TCASE

ReinscribirAlumno_SP_18

ReinscribirAlumno_SP_19_TCASE

ReinscribirAlumno_SP_19

ReinscribirAlumno_SP_10_TCASE

ReinscribirAlumno_SP_10

ReinscribirAlumno_SP_11_TCASE

ReinscribirAlumno_SP_11

ReinscribirAlumno_SP_5_TCASE

ReinscribirAlumno_SP_5

ReinscribirAlumno_SP_6_TCASE

ReinscribirAlumno_SP_6

ReinscribirAlumno_SP_3_TCASE

ReinscribirAlumno_SP_3

ReinscribirAlumno_SP_4_TCASE

ReinscribirAlumno_SP_4

ReinscribirAlumno_SP_1_TCASE

ReinscribirAlumno_SP_1

$registrados = \varnothing$

$inscripciones = \varnothing$

$alumno? = aLUMNO1$

ReinscribirAlumno_SP_2_TCASE

ReinscribirAlumno_SP_2

$registrados = \{aLUMNO1\}$

$inscripciones = \varnothing$

$alumno? = aLUMNO1$