

Algoritmos y Estructura de Datos 2

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Exorcismo Extremo

TP1

Integrante	LU	Correo electrónico
Rosinov, Gaston Einan	37/18	grosinov@gmail.com
Schuster, Martin Ariel	208/18	m.a.schuster98@gmail.com
Panichelli, Manuel	72/18	panicmanu@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. TAD JUEGO	3
2. TAD HABITACION	6
3. TAD ACCION	7
4. TAD DIRECCION	9
5. EXTENSIONES Y RENOMBRES	10

1. TAD JUEGO

TAD JUEGO

géneros : juego

exporta : TODO

usa : HABITACION

igualdad observacional

$$(\forall j, j' : \text{juego}) \left(j =_{\text{obs}} j' \iff \left((n = 0? =_{\text{obs}} m = 0?) \wedge_L (\neg(n = 0?) \Rightarrow_L (\text{pred}(n) =_{\text{obs}} \text{pred}(m))) \right) \right)$$

igualdad observacional

$$(\forall j, j' : \text{juego}) \left(j =_{\text{obs}} j' \iff \left(\begin{array}{l} (\text{accionesPJs}(j) =_{\text{obs}} \text{accionesPJs}(j')) \wedge \\ (\text{accionesFan}(j) =_{\text{obs}} \text{accionesFan}(j')) \wedge \\ (\text{localizarJugadores}(j) =_{\text{obs}} \text{localizarJugadores}(j')) \wedge \\ (\text{hab}(j) =_{\text{obs}} \text{hab}(j')) \wedge \\ ((\forall p : \text{pj}) (\text{vivePJ?}(j, p) =_{\text{obs}} \text{vivePJ?}(j', p))) \wedge \\ ((\forall f : \text{fantasma}) ((\text{viveFan?}(j, p) =_{\text{obs}} \text{viveFan?}(j', p)) \wedge \\ (\text{ubicacionInicialFan}(j, f) =_{\text{obs}} \text{ubicacionInicialFan}(j', f)))) \end{array} \right) \right)$$

observadores básicos

accionesPJs	: juego	→	dicc(pj, secu(accion))
accionesFan	: juego	→	dicc(pj, secu(accion))
hab	: juego	→	hab
vivePJ?	: juego $j \times \text{pj } p$	→	bool {p ∈ jugadores(j)}
viveFan?	: juego $j \times \text{fantasma } f$	→	bool {f ∈ fantasmas(j)}
ubicacionInicialFan	: juego $j \times \text{fantasma } f$	→	ubicacion {f ∈ fantasmas(f)}
localizarJugadores	: juego	→	dicc(pj, ubicacion)

generadores

iniciar	: conj(pj) $pjs \times \text{secu}(accion) as \times \text{ubicacion } u$ × hab h	→	juego {esConexa?(h) ∧ ¬ ∅?(as) ∧ ¬ ∅?(pjs) ∧ esValida?(h, pos(u))}
proxPaso	: juego $j \times \text{pj } p \times \text{accion } a$	→	juego {p ∈ jugadores(j) ∧ vivePJ?(j, p) ∧ ¬ termino?(j) ∧ ¬ esMirar(a)}

otras operaciones

jugadores	: juego	→	conj(pj)
fantasmas	: juego	→	conj(fantasma)
puntaje	: juego	→	nat
ronda	: juego	→	nat
paso	: juego	→	nat
cantAcciones	: juego × conj(pj)	→	nat
terminaRonda	: juego $j \times \text{pj } p \times \text{accion}$	→	bool {p ∈ jugadores(j)}
fantasmaEspecial	: juego	→	fantasma
termino?	: juego	→	bool

estanVivos	: juego \times conj(pj) pjs	\longrightarrow bool	$\{pjs \subseteq \text{jugadores}(j)\}$
ubicacionInicialPJ	: juego $j \times$ pj p	\longrightarrow ubicacion	$\{p \in \text{jugadores}(j)\}$
ubicacionPJ	: juego $j \times$ pj p	\longrightarrow ubicacion	$\{p \in \text{jugadores}(j)\}$
ubicacionFan	: juego $j \times$ fantasma f	\longrightarrow ubicacion	$\{f \in \text{fantasmas}(j)\}$
deducirUbicacion	: juego $j \times$ ubicacion $u \times$ acciones	\longrightarrow ubicacion	$\{esValida?(hab(j), pos(u))\}$
moriraFantasma	: juego $j \times$ pj $p \times$ accion \times fantasma f	\longrightarrow bool	$\{p \in \text{jugadores}(j) \wedge$ $f \in \text{fantasmas}(j) \wedge_L$ $viveFan?(j, f) \wedge$ $vivePJ?(j, p)\}$
moriraPJ	: juego $j \times$ conj(fantasma) $fs \times$ pj $p \times$ accion	\longrightarrow bool	$\{p \in \text{jugadores}(j) \wedge$ $fs \subseteq \text{fantasmas}(j) \wedge_L$ $vivePJ?(j, p)\}$
moriraPJPorFan	: juego $j \times$ fantasma $f \times$ pj $p \times$ accion	\longrightarrow bool	$\{p \in \text{jugadores}(j) \wedge$ $f \in \text{fantasmas}(j) \wedge_L$ $vivePJ?(j, p) \wedge$ $viveFan?(j, f)\}$
accionFan	: juego $j \times$ fantasma f	\longrightarrow accion	$\{f \in \text{fantasmas}(j) \wedge_L$ $viveFan?(j, f)\}$
inicializarAcciones	: conj(pj)	\longrightarrow dicc(pj, secu(accion))	
agregarFantasma	: juego $j \times$ ubicacion $u \times$ dicc(fantasma \times secu(accion)) \times fantasma \times secu(accion)	\longrightarrow dicc(fantasma, secu(accion))	$\{esValida?(hab(j), pos(u))\}$
generarAccionesFan	: juego $j \times$ ubicacion $u \times$ secu(accion)	\longrightarrow secu(accion)	$\{esValida?(hab(j), pos(u))\}$
nombreSiguienteFan	: juego	\longrightarrow fantasma	
axiomas	$\forall p: pj,$ $\forall pjs: \text{conj}(pj),$ $\forall f: \text{fantasma},$ $\forall fs: \text{conj}(\text{fantasma}),$ $\forall j: \text{juego},$ $\forall h: \text{hab},$ $\forall u, uInicialPJ: \text{ubicacion},$ $\forall a: \text{accion},$ $\forall as: \text{secu}(\text{accion})$		
accionesPJs(iniciar(pjs, as, u, h))	\equiv inicializarAcciones(pjs)		
accionesPJs(proxPaso(j, p, a))	\equiv if \neg terminaRonda(j, p, a) then definir(p, obtener(p, accionesPJs(j)) \circ a, accionesPJs(j)) else inicializarAcciones(jugadores(j)) fi		
accionesFan(iniciar(pjs, as, u, h))	\equiv definir(nombreSiguienteFan(j), as, vacio)		
accionesFan(proxPaso(j, p, a))	\equiv if \neg terminaRonda(j, p, a) then accionesFan(j) else agregarFantasma(j, ubicacionInicialPJ(j, p), accionesFan(j), nombreSiguienteFan(j), obtener(p, accionesPJs(j)) \circ a) fi		
hab(iniciar(pjs, as, u, h))	\equiv h		
hab(proxPaso(j, p, a))	\equiv hab(j)		
vivePJ?(iniciar(pjs, as, u, h), p')	\equiv true		

vivePJ?(proxPaso(j, p, a), p')	≡ terminaRonda?(j, p, a) ∨ if p = p' then ¬ moriraPJ(j, fantasmas(j), p, a) else vivePJ?(j, p') ∧ _L ¬ moriraPJ(j, fantasmas(j), p, a) fi
viveFan?(iniciar(pjs, as, u, h), f)	≡ true
viveFan?(proxPaso(j, p, a), f)	≡ terminaRonda?(j, p, a) ∨ (viveFan?(j, f) ∧ _L ¬ moriraFantasma(j, p, a, f))
ubicacionInicialFan(iniciar(pjs, as, u, h))	≡ u
ubicacionInicialFan(proxPaso(j, p, a))	≡ if f ∈ fantasmas(j) then ubicacionInicialFan(j, f) else ubicacionInicialPJ(j, p) fi
jugadores(j)	≡ claves(accionesPJs(j))
fantasmas(j)	≡ claves(accionesFan(j))
puntaje(j)	≡ ronda(j) - 1
ronda(j)	≡ #(fantasmas(j))
paso(j)	≡ cantAcciones(j, jugadores(j))
cantAcciones(j, pjs)	≡ if ∅?(pjs) then 0 else long(obtener(dameUno(pjs), accionesPJs(j))) + cantAcciones(j, sinUno(pjs)) fi
termino?(j)	≡ ¬ estanVivos(j, jugadores(j))
estanVivos(j, pjs)	≡ if ∅?(pjs) then true else vivePJ?(j, dameUno(pjs)) ∧ estanVivos(j, sinUno(pjs)) fi
fantasmaEspecial(j)	≡ #(claves(accionesFan(j)))
ubicacionInicialPJ(j, p)	≡ obtener(p, localizarJugadores(j))
ubicacionPJ(j, p)	≡ deducirUbicacion(j, ubicacionInicialPJ(j, p), obtener(p, accionesPJs(j)))
ubicacionFan(j, f)	≡ deducirUbicacion(j, ubicacionInicialFan(j, f), obtener(f, accionesFan(j)))
deducirUbicacion(j, u, as)	≡ if vacia?(as) then u else deducirUbicacion(j, ubicacionLuegoDe(prim(as), hab(j), u), fin(as)) fi
agregarFantasma(j, uInicialPJ, accionesFantasmas, f, as)	≡ definir(f, generarAccionesFantasma(j, uInicialPJ, as), accionesFantasmas)
generarAccionesFan(j, uInicialPJ, as)	≡ as & (nada • nada • nada • nada • nada) & invertir(hab(j), uInicialPJ, as)
inicializarAcciones(pjs)	≡ if ∅?(pjs) then vacio else definir(dameUno(pjs), <>, inicializarAcciones(sinUno(pjs))) fi
terminaRonda(j, p, a)	≡ moriraFantasma(j, p, a, fantasmaEspecial(j))

$\text{moriraFantasma}(j, p, a, f)$	$\equiv \text{pos}(\text{ubicacionFan}(j, f)) \in \text{posicionesAfectadasPor}(a, \text{hab}(j), \text{ubicacionPJ}(j, p))$
$\text{moriraPJ}(j, fs, p, a)$	$\equiv \text{if } \emptyset?(fs) \text{ then false else } (\text{viveFan?}(j, \text{dameUno}(fs)) \wedge_L \text{moriraPJPorFan}(j, \text{dameUno}(fs), p, a)) \vee_L \text{moriraPJ}(j, \text{sinUno}(fs), p, a) \text{ fi}$
$\text{moriraPJPorFan}(j, f, p, a)$	$\equiv \neg \text{moriraFantasma}(j, p, a, f) \wedge (\text{pos}(\text{ubicacionLuegoDe}(a, \text{hab}(j), \text{ubicacionPJ}(j, p))) \in \text{posicionesAfectadasPor}(\text{accionFan}(j, f), \text{hab}(j), \text{ubicacionFan}(j, f)))$
$\text{accionFan}(j, f)$	$\equiv \text{obtener}(\text{accionesFan}(j), f)[\text{paso}(j) \% \text{obtener}(\text{accionesFan}(j), f)]$
$\text{nombreSiguienteFan}$	$\equiv \#(\text{claves}(\text{accionesFan}(j))) + 1$

Fin TAD

2. TAD HABITACION

TAD HABITACION

géneros hab

exporta hab, observadores, generadores, esConexa?

usa POSICION, BOOL, NAT

igualdad observacional

$$(\forall h, h' : \text{hab}) \left(h =_{\text{obs}} h' \iff \left((\forall p : \text{posicion}) (\text{esValida?}(p, h) =_{\text{obs}} \text{esValida?}(p, h') \wedge_L \left(\text{esValida?}(p, h) \Rightarrow_L \text{estaOcupada?}(p, h) =_{\text{obs}} \text{estaOcupada?}(p, h') \right) \right) \right)$$

observadores básicos

esValida?	$: \text{hab} \times \text{posicion}$	$\longrightarrow \text{bool}$	
estaOcupada?	$: \text{hab } h \times \text{posicion } p$	$\longrightarrow \text{bool}$	$\{\text{esValida?}(h, p)\}$

generadores

nueva	$: \text{nat } n$	$\longrightarrow \text{hab}$	$\{n > 1\}$
ocupar	$: \text{hab } h \times \text{posicion } p$	$\longrightarrow \text{hab}$	$\{\text{esValida?}(h, p) \wedge_L \neg \text{estaOcupada?}(h, p)\}$

otras operaciones

esConexa?	$: \text{hab}$	$\longrightarrow \text{bool}$	
tamano	$: \text{hab}$	$\longrightarrow \text{nat}$	
posiciones	$: \text{hab}$	$\longrightarrow \text{conj}(\text{posicion})$	
posicionesLibres	$: \text{hab } h \times \text{conj}(\text{posicion}) \text{ ps}$	$\longrightarrow \text{conj}(\text{posicion})$	$\{\text{ps} \subseteq \text{posiciones}(h)\}$
verificarAlcance	$: \text{hab } h \times \text{conj}(\text{posicion}) \text{ ps}$	$\longrightarrow \text{bool}$	$\{\text{ps} \subseteq \text{posiciones}(h)\}$
$\text{verificarAlcancePos}$	$: \text{hab } h \times \text{conj}(\text{posicion}) \text{ ps} \times \text{posicion } p$	$\longrightarrow \text{bool}$	$\{\text{ps} \subseteq \text{posiciones}(h) \wedge p \in \text{posiciones}(h)\}$

axiomas $\forall h : \text{hab} \forall \text{ps} : \text{conj}(\text{posicion}) \forall p : \text{posicion} \forall n, k, \text{tam} : \text{nat}$

$$\text{esValida?}(\text{nueva}(n), p) \equiv 0 \leq \Pi_1(p) < n \wedge 0 \leq \Pi_2(p) < n$$

```

esValida?(ocupar(h,p'),p)      ≡ p = p' ∨L esValida?(h, p)
estaOcupada?(nueva(n),p)      ≡ false
estaOcupada?(ocupar(h,p'),p)  ≡ p = p' ∨ estaOcupada?(h, p)
tamano(nueva(n))              ≡ n
tamano(ocupar(h, p))          ≡ tamano(h)
esConexa?(h)                  ≡ verificarAlcance(h, posicionesLibres(posiciones(h)))
posicionesLibres(h, ps)       ≡ if  $\emptyset?(ps)$ 
                               then  $\emptyset$ 
                               else
                               ((if estaOcupada?(h, dameUno(ps)) then  $\emptyset$  else {dameUno(ps)} fi)
                               ∪ posicionesLibres(h, sinUno(ps))
                               fi
verificarAlcance(h, ps)       ≡ if  $\emptyset?(ps)$ 
                               then true
                               else
                               verificarAlcancePos(h, ps, dameUno(ps)) ∧ verificarAlcance(h, p)
                               fi
verificarAlcancePos(h, ps, p) ≡ if  $\emptyset?(ps)$ 
                               then true
                               else
                               esAlcanzable(h, p, dameUno(ps)) ∧ verificarAlcancePos(h, p, sinUno(ps))
                               fi
posiciones(h)                 ≡ darPosiciones(h, tamano(h) - 1, tamano(h) - 1, tamano(h) - 1)
darPosiciones(h, n, k, tam)   ≡ if  $n = 0? \wedge k = 0?$ 
                               then  $\emptyset$ 
                               else if  $k = 0?$ 
                               then Ag((n,k), darPosiciones(h, n - 1, tam, tam))
                               else Ag((n,k), darPosiciones(h, n, k - 1, tam))
                               fi
                               fi

```

Fin TAD

3. TAD ACCION

TAD ACCION

géneros accion

exporta observadores, generadores, genero, otras operaciones

igualdad observacional

$$(\forall a, a' : \text{accion}) \left(a =_{\text{obs}} a' \iff \left(\begin{array}{l} \text{esNada}(a) =_{\text{obs}} \text{esNada}(a') \wedge \\ \text{esDisparar}(a) =_{\text{obs}} \text{esDisparar}(a') \wedge \\ \text{esMover}(a) =_{\text{obs}} \text{esMover}(a') \wedge \\ \text{esMirar}(a) =_{\text{obs}} \text{esMirar}(a') \wedge \\ ((\text{esMover}(a) \vee \text{esMirar}(a)) \Rightarrow_L \text{direccion}(a) =_{\text{obs}} \text{direccion}(a')) \end{array} \right) \right)$$

secu(accion)

observadores básicos

esMover	: accion	→ bool
esMirar	: accion	→ bool

esDisparar	: accion	→ bool	
esNada	: accion	→ bool	
direccion	: accion a	→ direccion	{esMirar(a) \vee esMover(a)}

generadores

mover	: direccion	→ accion
mirar	: direccion	→ accion
disparar	:	→ accion
nada	:	→ accion

otras operaciones

ubicacionLuegoDe	: accion $a \times$ hab $h \times$ ubicacion u	→ conj(pos)	{esValida?(h , pos(u))}
posicionesAfectadasPor	: accion $a \times$ hab $h \times$ ubicacion u	→ conj(pos)	{esValida?(h , pos(u))}
$\neg \bullet$: accion	→ accion	
invertir	: hab $h \times$ ubicacion $u \times$ secu(accion)	→ secu(accion)	{esValida?(h , pos(u))}

axiomas $\forall n, m: \text{nat}, \forall u: \text{ubicacion}, \forall a: \text{habitacion}$

posicionesAfectadasPor(mover(d), h , u)	$\equiv \emptyset$
posicionesAfectadasPor(mirar(d), h , u)	$\equiv \emptyset$
posicionesAfectadasPor(nada, h , u)	$\equiv \emptyset$
posicionesAfectadasPor(disparar, h , u)	\equiv if esValida?(h , proxPosEnDir(dir(u), pos(u)) \wedge_L \neg estaOcupada?(h , proxPosEnDir(dir(u), pos(u))) then Ag(proxPosEnDir(dir(u), pos(u)),posicionesAfectadasPor(disparar, h , (proxPosEnDir(dir(u), pos(u)), dir(u)))) else \emptyset fi
invertir(h , u , as)	\equiv if vacia?(as) then $\langle \rangle$ else invertir(h , ubicacionLuegoDe(prim(as), h , u), fin(as)) \vee \neg (prim(as), h , u) fi
\neg (mover(d), h , u)	\equiv if pos(ubicacionLuegoDe(mover(d), h , u)) = pos(u) then mirar(opuesta(d)) else mover(opuesta(d)) fi
\neg (mirar(d), h , u)	\equiv mirar(opuesta(d))
\neg (disparar, h , u)	\equiv disparar
\neg (nada, h , u)	\equiv nada
ubicacionLuegoDe(nada, h , u)	$\equiv u$
ubicacionLuegoDe(disparar, h , u)	$\equiv u$
ubicacionLuegoDe(mirar(d), h , u)	$\equiv \langle \text{pos}(u), d \rangle$
ubicacionLuegoDe(mover(d), h , u)	$\equiv \langle (\text{if esValida?(h, proxPosEnDir(d, pos(u))) \wedge_L \negestaOcupada?(h, proxPosEnDir(d, pos(u)))then proxPosEnDir(d, pos(u))else pos(u)fi}), d \rangle$
esMirar(mirar(d))	\equiv true
esMirar(mover(d))	\equiv false

esMirar(disparar)	\equiv false
esMirar(nada)	\equiv false
esMover(mirar(d))	\equiv false
esMover(mover(d))	\equiv true
esMover(disparar)	\equiv false
esMover(nada)	\equiv false
esDisparar(mirar(d))	\equiv false
esDisparar(mover(d))	\equiv false
esDisparar(disparar)	\equiv true
esDisparar(nada)	\equiv false
esNada(mirar(d))	\equiv false
esNada(mover(d))	\equiv false
esNada(disparar)	\equiv false
esNada(nada)	\equiv true
direccion(mirar(d))	\equiv d
direccion(mover(d))	\equiv d

Fin TAD

4. TAD DIRECCION

TAD DIRECCION

géneros direccion

exporta observadores, generadores, otras operaciones

igualdad observacional

$$(\forall d, d' : \text{direccion}) \left(d =_{\text{obs}} d' \iff \left(\begin{array}{l} \text{esArriba}(d) =_{\text{obs}} \text{esArriba}(d') \wedge \\ \text{esAbajo}(d) =_{\text{obs}} \text{esAbajo}(d') \wedge \\ \text{esIzquierda}(d) =_{\text{obs}} \text{esIzquierda}(d') \wedge \\ \text{esDerecha}(d) =_{\text{obs}} \text{esDerecha}(d') \end{array} \right) \right)$$

observadores básicos

esArriba	: direccion	\longrightarrow bool
esAbajo	: direccion	\longrightarrow bool
esIzquierda	: direccion	\longrightarrow bool
esDerecha	: direccion	\longrightarrow bool

generadores

arriba	:	\longrightarrow direccion
abajo	:	\longrightarrow direccion
izquierda	:	\longrightarrow direccion
derecha	:	\longrightarrow direccion

otras operaciones

opuesta	: direccion	\longrightarrow direccion
---------	-------------	-----------------------------

$\text{proxPosEnDir} : \text{direccion} \times \text{posicion} \longrightarrow \text{posicion}$

axiomas

$\text{opuesta}(\text{arriba})$	$\equiv \text{abajo}$
$\text{opuesta}(\text{abajo})$	$\equiv \text{arriba}$
$\text{opuesta}(\text{izquierda})$	$\equiv \text{derecha}$
$\text{opuesta}(\text{derecha})$	$\equiv \text{izquierda}$
$\text{proxPosEnDir}(\text{arriba}, p)$	$\equiv \langle \Pi_1(p), \Pi_2(p) + 1 \rangle$
$\text{proxPosEnDir}(\text{abajo}, p)$	$\equiv \langle \Pi_1(p), \Pi_2(p) - 1 \rangle$
$\text{proxPosEnDir}(\text{izquierda}, p)$	$\equiv \langle \Pi_1(p) - 1, \Pi_2(p) \rangle$
$\text{proxPosEnDir}(\text{derecha}, p)$	$\equiv \langle \Pi_1(p) + 1, \Pi_2(p) \rangle$
$\text{esArriba}(\text{arriba})$	$\equiv \text{true}$
$\text{esArriba}(\text{abajo})$	$\equiv \text{false}$
$\text{esArriba}(\text{izquierda})$	$\equiv \text{false}$
$\text{esArriba}(\text{derecha})$	$\equiv \text{false}$
$\text{esAbajo}(\text{arriba})$	$\equiv \text{false}$
$\text{esAbajo}(\text{abajo})$	$\equiv \text{true}$
$\text{esAbajo}(\text{izquierda})$	$\equiv \text{false}$
$\text{esAbajo}(\text{derecha})$	$\equiv \text{false}$
$\text{esIzquierda}(\text{arriba})$	$\equiv \text{false}$
$\text{esIzquierda}(\text{abajo})$	$\equiv \text{false}$
$\text{esIzquierda}(\text{izquierda})$	$\equiv \text{true}$
$\text{esIzquierda}(\text{derecha})$	$\equiv \text{false}$
$\text{esDerecha}(\text{arriba})$	$\equiv \text{false}$
$\text{esDerecha}(\text{abajo})$	$\equiv \text{false}$
$\text{esDerecha}(\text{izquierda})$	$\equiv \text{false}$
$\text{esDerecha}(\text{derecha})$	$\equiv \text{true}$

Fin TAD

5. EXTENSIONES Y RENOMBRES

TAD FANTASMA ES NAT

TAD PJ ES NAT

TAD POSICION ES TUPLA(NAT, NAT)

TAD NAT extiende NAT

otras operaciones

$\bullet \% \bullet : \text{nat} \times \text{nat} \longrightarrow \text{nat}$

axiomas $\forall n, m: \text{nat}$

$n \% m \equiv \text{if } n < m \text{ then } n \text{ else } (n - m) \% m \text{ fi}$

Fin TAD

TAD UBICACION extiende TUPLA(POSICION, DIRECCION)

otras operaciones

$\text{pos} : \text{ubicacion} \longrightarrow \text{posicion}$

$\text{dir} : \text{ubicacion} \longrightarrow \text{direccion}$

axiomas $\forall u: \text{ubicacion}$

$\text{pos}(u) \equiv \Pi_1(u)$

$\text{dir}(u) \equiv \Pi_2(u)$

Fin TAD

TAD SECUENCIA extiende SECUENCIA

otras operaciones

$\bullet[\bullet] : \text{secu}(\alpha) \times \text{nat } i \longrightarrow \alpha \quad \{i < \text{long}(s)\}$

axiomas $\forall s: \text{secu}(\alpha), \forall i: \text{nat}$

$s[i] \equiv \text{if } i = 0? \text{ then } \text{prim}(s) \text{ else } \text{fin}(s)[i - 1] \text{ fi}$

Fin TAD